Вакал Є.С., Личман В.В., Обвінцев О.В., Бублик В.В., Попов В.В.

Задачі до курсу «Інформатика та програмування»

Вступ

Цю збірку задач створено для підтримки курсу «Інформатика і програмування» для студентів механіко-математичного факультету Київського національного університету імені Тараса Шевченка. Нумерація тем співпадає з відповідною нумерацією курсу лекцій. У більшості задач розв'язком повинен бути алгоритм або (та) програми у мовах програмування Паскаль та Сі. До найбільш важливих з методичної точки зору завдань, а також до найбільш важких з них надано розв'язки, відповіді або вказівки. Розв'язки задач наводяться у алгоритмічній мові.

При підборі завдань використано, зокрема, матеріали курсів інформатики та програмування, які викладаються на механіко-математичному факультеті Московського Державного Університету ім. М.В. Ломоносова, та на факультеті кібернетики Київського національного університету імені Тараса Шевченка.

1. Лінійні програми

1.1. Обчислити силу притягання F між двома тілами, що мають маси m_1, m_2, на відстані r.
<u>*Розв`язок*</u>. Шукана сила визначається за формулою $F=\gamma * m_1 * m_2 / r^2$, де $\gamma = 6.673 * 10^{-11}\ Н*м^2/кг^2$.
Отримаємо алгоритм

<u>*Алгоритм*</u> *Сила* <u>*це*</u>
    <u>*змін*</u> *m1,m2,R,Gam,F:дійсн;*
<u>*поч*</u>
    взяти(*m1,m2,R*);
    γ←6.673E-11;
    *F←Gam*m1*m2 /(R*R);*
    показати(*F*)
<u>*ка.*</u>

1.2. Знайти ланцюги простих присвоєнь, рівносильні наступним узагальненим присвоєнням

а) $$x \leftarrow a \times a - c + \cfrac{a \times a - c}{b \times c + \cfrac{c}{d + \cfrac{l}{f} + b \times c}};$$

б) $$y \leftarrow \frac{\frac{1}{a} + \frac{1}{b}}{\frac{1}{a} - \frac{1}{c}};$$ в) $$z \leftarrow \frac{a+b}{c+d} - \frac{a \times b}{a+b}.$$

1.3. Скласти алгоритми та програми для обчислення значення многочленів:

a) $y=x^4-2x^3+x^2+1$, $x=3$;
b) $y=x^6+3x^4-5x^2+x+1$, $x=2$;
c) $y=4x^5+2x^4+6x^3+7x^2+x+3$, $x=-1$;
d) $y=x^8+5x^4-2x^2+x$, $x=2$;
e) $y=x^9+2x^6+3x^3-5$, $x=2$.

1.4. Скласти алгоритми та програми для обчислення значення многочленів від двох змінних та виконати їх при заданих значеннях аргументів.

a) $z=x^6y^3+x^4y^2+x^2$, $x=-1$, $y=2$;
b) $z=x^2y^2+x^3+y^3+3x^2y+3xy^2+x^2+2xy+y^2$, $x=2$, $y=-1$;

1.5. Скласти алгоритми та програми для обчислення значень виразів та виконати їх при заданих значенях аргументів:

а) $y=x^2+x+1/x+1/x^2$ $\qquad x=3;$

b) $y=x^{16}+x^4,$ $\qquad x=2.$

1.6. Скласти алгоритм та програми для виконання взаємного обміну значень змінних *x* та *y*.

1.7. Скласти алгоритм та програми, що переводять значення змінних *a, b, c, d* у *b, c, d, a* у вказаному порядку.

1.8. Яку задачу вирішує наступний ланцюг присвоєнь

$x\leftarrow x+y;\ y\leftarrow x-y;\ x\leftarrow x-y$?

2. Розгалужені програми

2.1 Умови

2.1. Довести властивості бульових операцій

а) $p \vee q \equiv q \vee p$;

б) $(p \ \&\ q)\ \&\ r \equiv p\ \&\ (q\ \&\ r)$;

в) $p \vee q\ \&\ r \equiv (p \vee q)\ \&\ (p \vee r)$;

г) $\neg(p \vee q) \equiv \neg p\ \&\ \neg q$.

2.2. Довести властивості імплікації

а) $p \supset (p \vee q) \equiv Іст$;

б) $\neg p \supset \neg q \equiv q \supset p$;

в) $(p\ \&\ q) \supset r \equiv p \supset (q \supset r)$;

г) $p \supset (q \vee r) \equiv (p \supset q) \vee r$.

2.3. Спростити висловлювання

а) $(r \supset p\ \&\ q) \supset (r \supset p)\ \&\ (r \supset q)$;

б) $(p \supset (p \supset (p \supset (p \supset q))))$;

в) $(p \supset q) \supset p\ \&\ q$.

2.4. Довести властивості альтернативи

а) $if\ (Іст, q, r) \equiv q$;

б) $if\ (Хиб, q, r) \equiv r$;

в) $if\ (p, q, q) \equiv q$.

2.5. Вважаючи доведеними властивості

$if\ (\neg p, q, r) \equiv if\ (p, r, q)$,

$if\ (p_1\ \&\ p_2, q, r) \equiv if\ (p_1, if\ (p_2, q, r), r)$,

довести властивості альтернативи

а) $if\ (p_1 \vee p_2, q, r) \equiv if\ (p_1, q, if\ (p_2, q, r))$;

б) $if\ (p_1 \supset p_2, q, r) \equiv if\ (p_1, if\ (p_2, q, r), q)$

2.6. Довести властивості умов

а) $x<=y \equiv \neg(x>y) \equiv x<y \vee x=y$;

б) $x>=y \equiv \neg(x<y) \equiv x>y \vee x=y$.

2.7. Спростити бульові вирази

а) $(x>0 \vee x=0)$; б).$(x>=0\ \&\ x=0)$; в) $\neg\ (x>0\ \&\ y>0)$; г) $\neg\ (x=0)$;

д) $(x>0 \vee x>=0)$; є).$(x>0 \vee x<=0)$; ж).$(x>0\ \&\ x<0)$;

з) $x=0 \vee (x=0\ \&\ y>0)$; і).$(x>0 \vee x<=0)\ \&\ y>0$;

<u>Розв'язок</u>. є). Користуючись властивостями бульових операцій та відношень, отримаємо

$(x>0 \vee x<=0) = (x>0) \vee (x<0) \vee (x=0) = (x<>0) \vee (x=0) = Іст.$

2.8. Записати умову впорядкованості значень змінних *a,b* та *c.*

2.9. Довести властивості умовного виразу

а) $IF\ (Іст, a, b) \equiv a$;

б) $IF\ (Іст, a, b) \equiv a$;

в) $IF\ (Хиб, a, b) \equiv b$;

г) $IF\ (F, a, b) \equiv IF\ (\neg F, b, a)$;

д) $IF\ (F, a, a) \subset a$;

е) $IF\ (F_1\ \&\ F_2, a, b) \subset IF\ (F_1, IF\ (F_2, a, b), b)$;

2.10. Вважаючи доведеними властивості г) та е) попередньої задачі, довести властивості умовного виразу

а) $IF\ (F_1 \vee F_2, a, b) \subset IF\ (F_1, a, IF\ (F_2, a, b))$;

б) IF $(F_1 \supset F_2, a, b) \subset IF\ (F_1, IF\ (F_2, a, b), a)$.

2.2 Бульове присвоєння

2.11. Виявити приналежність точки (*x,y*):

а) першому; б) другому; в) третьому; г) четвертому координатному квадранту.

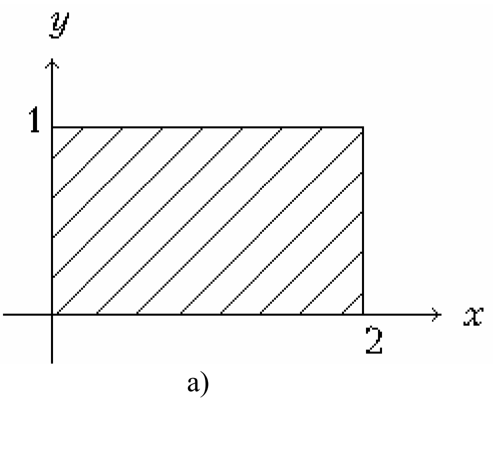
Вказівка. б).Розглянути булів вираз ($x<=0\ \&\ y>=0$).

2.12. Скласти алгоритм перевірки можливості існування трикутника з заданими сторонами *a,b,c.*

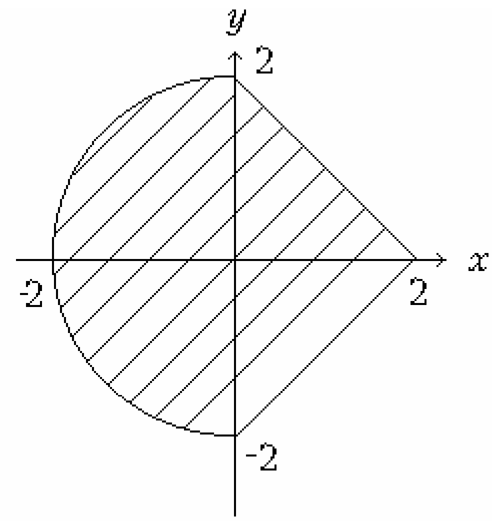
Розв'язок. Результатом виконання цього алгоритму треба вважати деякий булів вираз. Причому значення *Іст* інтерпретується як відповідь «так, трикутник з такими сторонами існує», а значення *Хиб* — як відповідь «ні, трикутник з такими сторонами не існує». Якщо пам'ятати, що в будь-якому трикутнику кожна сторона менше ніж сума двох інших сторін, можливість існування трикутника відповідає істинності виразу $(a+b>c)\&(a+c>b)\&(b+c>a)$. Отримаємо алгоритм

```
Алгоритм Трикутник це
        змін A,B,C:дійсн;
                L:бул;
поч
        взяти(A,B,C);
        L←(A+B>C)&(A+C>B)&(B+C>A);
        показати(L)
ка.
```

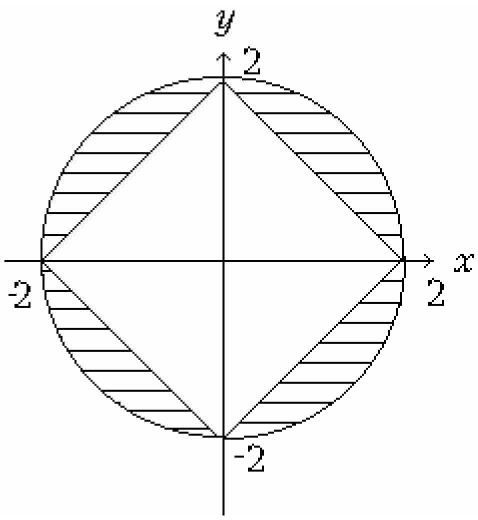
2.6. Скласти алгоритм перевірки приналежності точки (*x,y*) зафарбованій області (див мал. 2.1.)



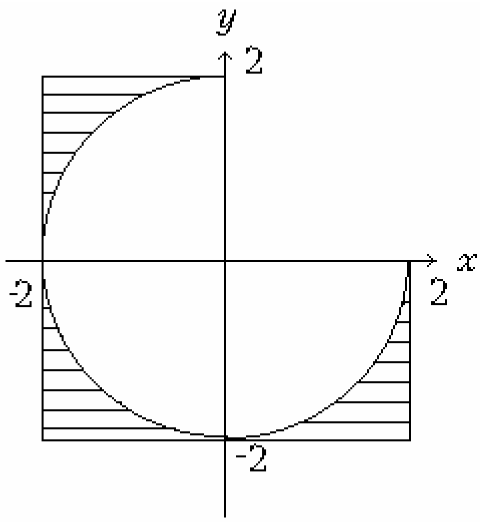
а)



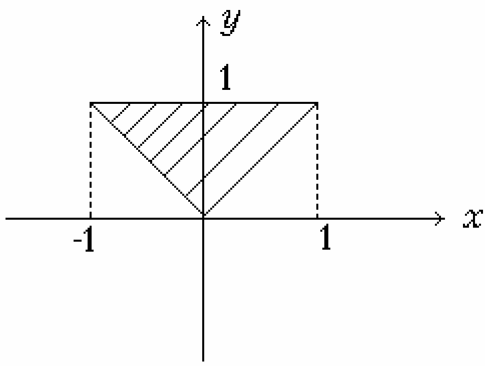
б)



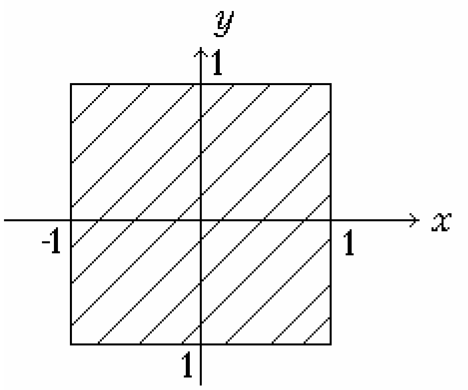
в)



г)



д)



е)

Мал. 2.1.

Вказівка д). Розглянути булів вираз ($y>=x$)&($y+x>=0$)&($y<=1$)

2.13. Точка площини задана своїми координатами (x,y). Перевірити її приналежність

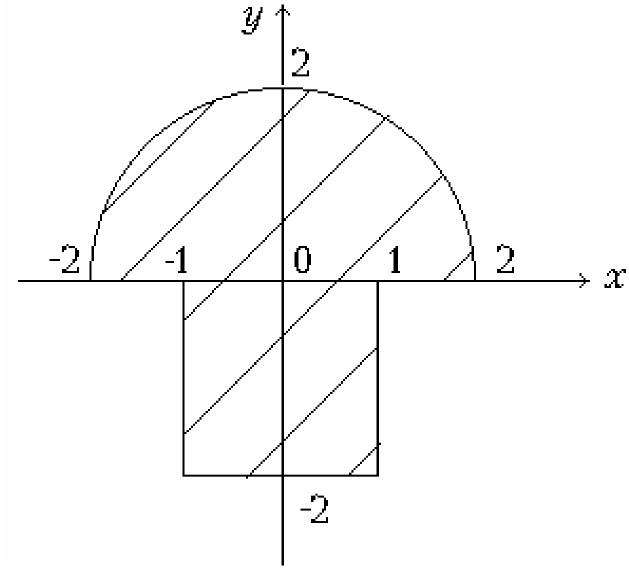
а) кільцю з центром у точці (1,2),внутрішнім радіусом 3 та зовнішнім радіусом 4;

б) квадрату з центром у точці (–2,1),сторони якого паралельні до координатних осей, довжина сторони дорівнює 4.

Вказівка а):розглянути булів вираз:

$$((x-1)^2+(y-2)^2>=9)\&((x-1)^2+(y-2)^2<=16)$$

2.14. Точка площини задана своїми координатами (x,y). Скласти алгоритм який перевіряє приналежність точки грибу, зображеному на мал. 2.2.



Мал. 2.2. Гриб.

2.3 Розгалуження

2.15. Скласти алгоритм вибору максимального з двох значень змінних *a* та *b*.

Розв`язок: За означенням
$$\max(a,b) = \begin{cases} a, & a > b, \\ b, & b > a \end{cases}$$
Отримаємо алгоритм.

Алгоритм Max_ab це
    змін a,b,max:дійсн;
поч
    взяти(*a,b*);
    якщо a>b то max←a
    інакше max←b
    кр;
    {max=max(a,b)}
    показати(*max*)
ка.

2.16. Впорядкувати значення змінних *a* та *b* таким чином, щоб виконувалося співвідношення *a<=b*.

2.17. Скласти алгоритм обчислення *min*(*a,b,c*).

Розв`язок. Легко бачити, що *min*(*a,b,c*)=*min*(*min*(*a,b*),*c*). Тому спочатку обрахуємо *min_ab*=*min*(*a,b*), а потім *min_abc*=*min*(*min_ab,c*). Оскільки запам`ятовування значення *min*(*a,b*) умовою задачі не вимагається, використаємо змінну *min* для позначення як *min_ab* ,так і *min_ abc*. Отримаємо алгоритм:

Алгоритм Min_abc це
    змін min,a,b,c:дійсн;
поч
    взяти (*a,b,c*);
    якщо a<=b то min←a
    інакше min←b
    кр;
    якщо min >c то min←c кр;

показати (min)

ка.

2.18. Скласти алгоритм для обчислення величини *a* і виконати його для вказаних значень аргументів:

а) $a=max(x,y,z)$; $x=1, y=2, z=3; x=2, y=1, z=0$;

б) $a=max(2*x, x^2, 1-x)$; $x=0, x=1, x=-2$.

2.19. Скласти алгоритм визначення кількості максимальних чисел серед *a,b,c*.

Вказівка: позначимо через *число_max*(*a,b*) і *max*(*a,b*) відповідно кількість максимальних серед чисел *a* і *b* та їх максимум. Кількість максимумів визначимо як

$$\text{число\_max}(a,b) = \begin{cases} 2, \text{ якщо } a = b, \\ 1, \text{ якщо } a \neq b. \end{cases}$$

Тоді *число_max*(*a,b,c*) можна визначити як

$$\text{число\_max}(a,b,c) = \begin{cases} 1, & \text{якщо } c > max(a,b), / \\ \text{число\_max}(a,b)+1, & \text{якщо } c = max(a,b), \\ \text{число\_max}(a,b), & \text{якщо } c < max(a,b). \end{cases}$$

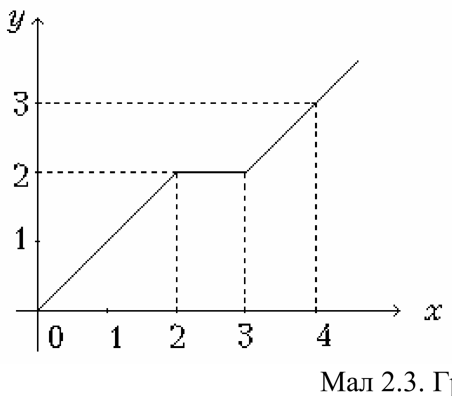
2.20.Скласти алгоритм знаходження кількості різних чисел серед *a,b,c*.

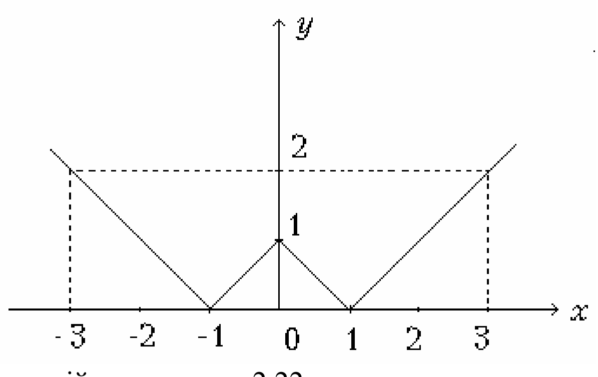
2.21. Обчислити значення функцій:

а) $y = |x|$; б) $y = sign(x)$; в) $y = \begin{cases} x^2, & \text{при } -3 \le x < 2, \\ 9, & \text{у інших випадках,} \end{cases}$

г) $y = \begin{cases} 0, \text{ якщо } x \le 0, \\ x^2, \text{ якщо } 0 < x \le 1, \\ x^4, \text{ у інших випадках.} \end{cases}$

2.22. Обчислити значення функцій, графіки яких зображені на мал. 2.3.





Мал 2.3. Графіки функцій до завдання 2.22.

2.23. Обчислити значення виразу:

$$z = \begin{cases} \max(x, y+5), & x > y, \\ \min(x+1, y, 3), & x <= y. \end{cases}$$

2.24. Обчислити значення $x=f(y)-6.3$, де $y=z+2$ та

$$f(y)=\begin{cases} y^2-0.3, & y<0,\\ 0, & 0<=y<=1,\\ y^2+y, & y>1.\end{cases}$$

2.25. Скласти розгалуження для обчислення величин:

а) $$Arctg(x,y)=\begin{cases} arctg\left(\frac{x}{y}\right), & \text{якщо } y>0,\\ \frac{\pi}{2}, & \text{якщо } x>0 \text{ і } y=0,\\ \pi+arctg\left(\frac{x}{y}\right), & \text{якщо } x>=0 \text{ і } y<0,\\ -\frac{\pi}{2}, & \text{якщо } x<0 \text{ і } y=0,\\ -\pi+arctg\left(\frac{x}{y}\right), & \text{якщо } x<0 \text{ і } y<0.\end{cases}$$

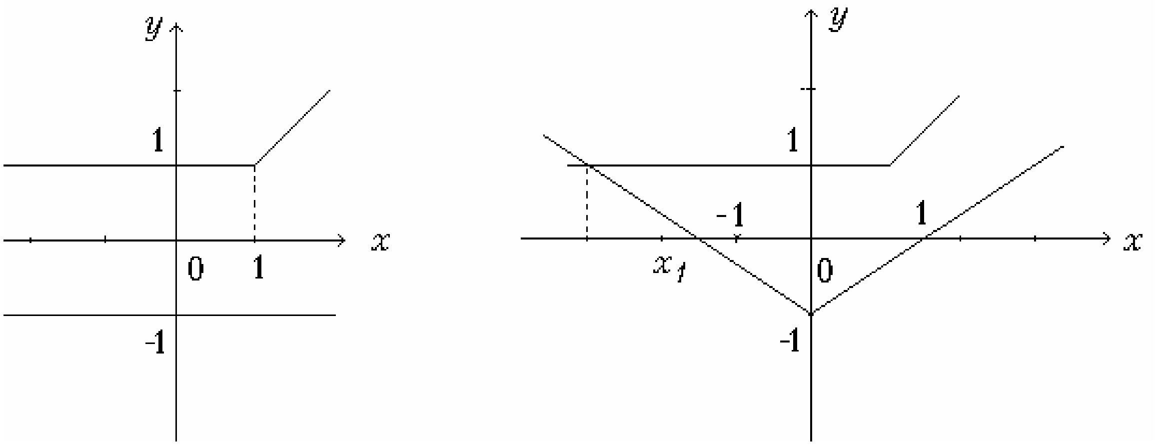
б) $$a=\begin{cases} x^{\left(y^z\right)}, & \text{якщо } y>0,\ z>0,\\ x^{(-y)^{(-z)}} & \text{якщо } y<0,\ z<0,\\ 0, & \text{у інших випадках.}\end{cases}$$

2.26. Скласти алгоритм для розв'язання системи рівнянь

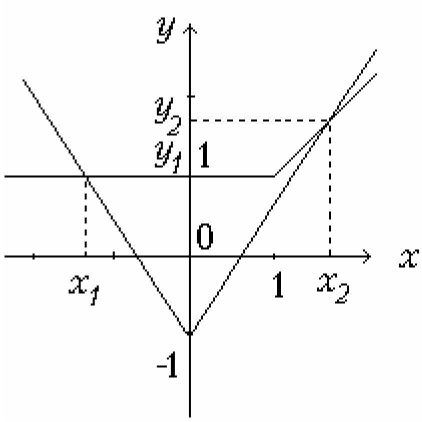
$$\begin{cases} y=\max(1,x)\\ y=|a\cdot x|-1.\end{cases}$$

Виконати його при a=-0.5; a=1.5; a=3.

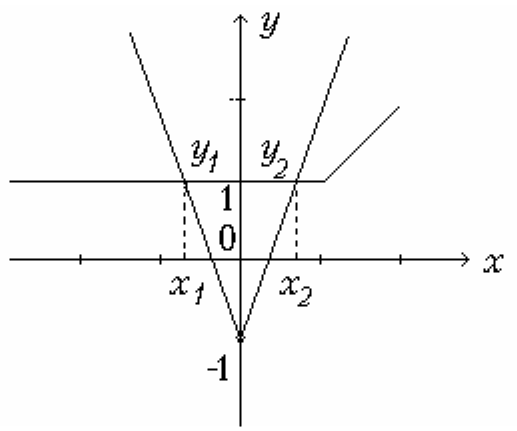
Розв`язок: взаємне розміщення графіків при різних значеннях параметра a зображено на мал. 2.4.



а) $a=0$ б) $0<|a|<=1$



в) *1<|a|<=2*



г) *|a|>0*

Мал. 2.4. Залежність графіка системи від|a|.

Розгляд перерахованих випадків приводить до наступного алгоритма:

Алгоритм Система це
    змін a,x1,y1,x2,y2:дійсн;
        k:нат;
поч
    взяти (*a*);
    якщо a=0 то {випадок а)}
        показати ('розв`зків немає')
    інакше {система сумісна}
        якщо a<0 то a← — a кр;
        *x1← -2/a ; y1←*1 *; k←*1 *; {k-кількість розв`язків}*
        якщо a>2 то {випадок г)}
            x2← — x1 ; y2←y1 ; k←2
        інакше {0<a<=2}
            *якщо a>*1 *то { випадок в)}*
                x2←1/(a-1) *; y2←x2 ; k←*2
            кр
        кр;
        показати (x1,y1);
        якщо k=2 *то показати (x2,y2) кр*
    кр
ка.

Виконання алгоритму при *a* =1.5 показане у табл. 2.1.

Таблиця 2.1. Трасувальна таблиця до завдання 2.18 при *1<|a|<=2*

	a	*x1*	*y1*	*x2*	*y2*	*к*
взяти (*a*)	1.5					
$a=0^-$						
$a<0^-$						
x1← -2/a		-4/3				
*y1←*1			*1*			
*k←*1						1
$a>2^-$						
$a>1^+$						
*x2←*1/(*a*-1)				2		
y2←x2					*2*	

k←2						*2*
показати (*x1,y1*)		-4/3	1			
к=2^+						
показати (*x2,y2*)				2	2	

2.27. Скласти алгоритми для розв'язання систем рівнянь:

а) $\begin{cases} y = x + 1 \\ y = \left|a(x-1)\right| \end{cases}$; б) $\begin{cases} y = ax + b \\ |x| + |y| = 1 \end{cases}$; в) $\begin{cases} (x^2-1)(y^2-1) = 0 \\ (x-a)^2 = (y-b)^2 \end{cases}$.

2.28. Скласти алгоритми дослідження

а) рівняння $ax^2 + bx + c = 0$;

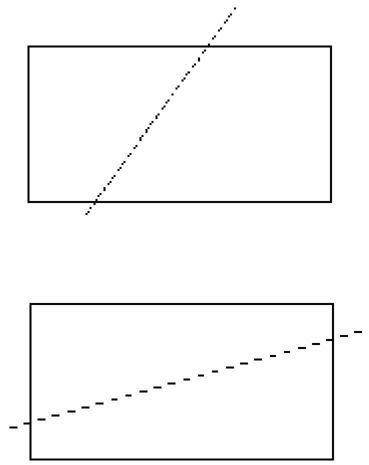
б) рівняння $ax^4 + bx^2 + c = 0$;

в) системи $\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$

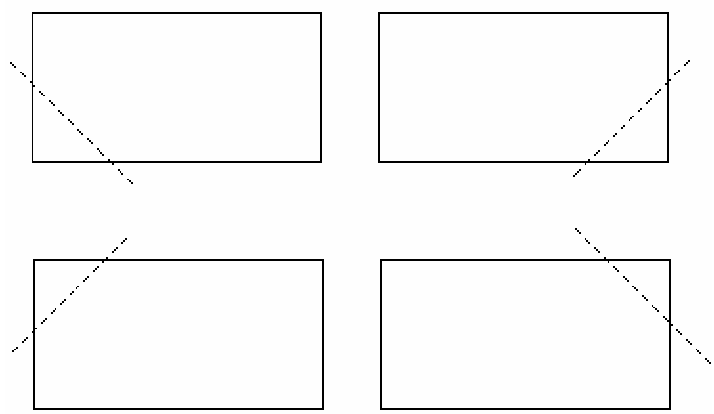
2.29. Скласти алгоритм обчислення інтервалу від'ємності функції $\varphi(x) = \max(a - x,\ b - 2x,\ x + c)$ при довільних *a,b,c.*

2.30. Скласти алгоритм для обчислення різниці площ фігур на які пряма *y=ax+b* ділить прямокутник *P*={(*x,y*): a_1<=*x*<=a_2, b_1<=*y*<=b_2}.
Вказівка: розглянути випадок *a* = 0 та табл. 2.2.

Таблиця 2.2. Розгляд випадків до завдання 2.22

<table>
<tr><th colspan="2">Дві трапеції</th></tr>
<tr><td>$a_1 \le \dfrac{b_2 - b}{a} \le a_2$ з
$a_1 \le \dfrac{b_1 - b}{a} \le a_2$
$\overline{b_1 \le aa_1 + b \le b_2}$ з
$b_1 \le aa_2 + b \le b_2$</td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Трикутник і п`ятикутник</th></tr>
<tr><td></td><td>$b_1 \le aa_1 + b \le b_2$ | $b_1 \le aa_2 + b \le b_2$</td></tr>
</table>

$a_1 \le \frac{b_1 - b}{a} \le a_2$ $a_1 \le \frac{b_2 - b}{a} \le a_2$	

2.31. Скласти алгоритм, який перевіряє приналежність точки $P(x,y,z)$ поверхні кулі з радіусом R і центром в точці $O(a,b,c)$.

2.32. Скласти алгоритм, який перевіряє приналежність початку координат трикутнику з вершинами (x_1,y_1), (x_2,y_2), (x_3,y_3).

2.33. Скласти алгоритм, який по колу $((x - u)^2 + (y - v)^2 = r^2$ та прямій $ax+by+c=0$ встановлює, який випадок має місце:

а) дві точки перетину;
б) одна точка дотику;
в) жодної спільної точки.

2.34. Знайти число точок перетину кола $x^2 + y^2 = r^2$ з відрізком $\{x=a,\ b<=y<=b+d^2\}$.
2.35. З'ясувати, чи перетинаються два відрізки на площині.
2.36. З'ясувати, чи перетинаються два кола на площині.
2.37. Скласти алгоритм обчислення площі та периметра а)об'єднання; б) перетину двох прямокутників:

$P_1=\{(x,y):a_1<=x<=a_2,\ b_1<=y<=b_2\}$, $P_2=\{(x,y):\ c_1<=x<=c_2,\ d_1<=y<=d_2\}$.

2.38. Задано два квадрати, сторони яких паралельні координатним вісям. З'ясувати, чи перетинаються вони, якщо так,то зобразити їх різниці у вигляді об'єднання прямокутників.

3. Циклічні програми

3.1 Арифметичний цикл

3.1. Скласти алгоритм обчислення степенів:

а) $y = a^n$, n — натуральне число;
б) $y = \frac{1}{a^n}$, n — натуральне число;
в) $y = a^n$, n — ціле число.

3.2. Скласти алгоритм обчислення
а) $y=sin(sin(...sin(x)...))$ (n раз); б) $y=sin^n x$.

3.3. Скласти алгоритм обчислення добутку
$a*b*a*b*...*a*b*a$ ($2n$ знаків множення).

3.4. Скласти алгоритми для обчислення значень многочленів і виконати їх при заданих значеннях аргументів:

а) $y = x^n+x^{n-1}+...+x^2+x+1$, n=3, x=2;

б) $y = x^{2^n} + x^{2^{n-1}} + ... + x^4 + x^2 + 1$, n=4,x=1;

в) $y = x^{3^n} + x^{3^{n-1}} + ... + x^9 + x^3 + 1$, n=3,x=1;

г) $y = x^{2^n}y^n + x^{2^{n-1}}y^{n-1} + ... + x^2y + 1$, n=4,x=1,y=2;

д) $y = x^{1^2} + x^{2^2} + ... + x^{n^2}$, n=5,x=-1.

Розв`язок а) Розглянемо два способи розв`язування.

Спосіб 1. Позначимо $z^k=x_k$. Тоді цикл

z←1;
повт n раз
*z←z*x*
кц

забезпечить послідовне обчислення z_0, z_1, ... , z_n. Передбачивши підсумовування, введення і виведення, отримаємо

```
алг Многочлен_1 це
    змін x,y,z:дійсн;
        n:нат;
поч
    взяти (n,x);
    z←1;y←1;
    повт n раз
        z←z*x ; y←y+z
    кц;
    показати (y)
ка.
```

Спосіб 2. Розставивши дужки наступним чином

$$= x^n+x^{n-1}+...+x^2+x+1 = (...(x+1)x+1)x+...+1)x+1,$$

отримаємо алгоритм

```
алг Многочлен_2 це
    змін x,y:дійсн;
        n:нат;
поч
    взяти (n,x);
    y←1;
    повт n раз
        y←y*x+1
    кц;
    показати (y)
ка.
```

Трасувальна таблиця алгоритму приведена в табл. 3.1.

Таблиця 3.1. Трасувальна таблиця до завдання 3.4 а)

	x	n	y	коментар
взяти(n,x)	2	3		
y←1			1	y+1
повт 3 *рази*				

1) $y \leftarrow y*x+1$			3	$y=x+1$
2) $y \leftarrow y*x+1$			7	$y=x^2+x+1$
3) $y \leftarrow y*x+1$			15	$y=x^3+x^2+x+1$
кінець_циклу				
показати(y)			15	

Вказівка б). Цикл *повт n раз* $x \leftarrow x*x$ *кц* забезпечує послідовне обчислення степенів x^{2^k}.

Вказівка д). $x^{i^2} = x^{(i-1)^2} * x^{2*i-1}$.

3.5. Скласти алгоритм обчислення добутку $p=m*n$, використовуючи операцію додавання та виконати його при $m=5$, $n=3$.

3.6. Скласти алгоритм обчислення факторіалу

$$p=n!$$

3.7. Скласти алгоритм обчислення

а) $\sqrt{2+\sqrt{2+\ldots+\sqrt{2}}}$ (n коренів),

б) $\sqrt{3+\sqrt{6+\ldots+\sqrt{3(n-1)+\sqrt{3n}}}}$.

3.8. Скласти алгоритми обчислення значень многочлена

а) $y = nx^{n-1}+(n-1)x^{n-2}+\ldots+2x+1$;

б) $y = x^n(1-x)^m$, $(n,m \geq 0)$;

в) $y = \dfrac{x^n}{n!} + \dfrac{x^{n-1}}{(n-1)!} + \ldots + \dfrac{x}{1} + 1$.

3.2. Програмування рекурентних співвідношень

3.9. Скласти алгоритми для обчислення елементів послідовностей

а) $x_k = \dfrac{x^k}{k}$ $(k \geq 1)$;

б) $x_k = \dfrac{(-1)^k x^k}{k}$ $(k \geq 1)$;

в) $x_k = \dfrac{x^k}{k!}$ $(k \geq 0)$;

г) $x_k = \dfrac{(-1)^k x^k}{k!}$ $(k \geq 0)$;

д) $x_k = \dfrac{x^{2k}}{(2k)!}$ $(k \geq 0)$;

е) $x_k = \dfrac{x^{2k+1}}{(2k+1)!}$ $(k \geq 0)$;

ж) $x_k = \dfrac{(-1)^k x^{2k}}{(2k)!}$ $(k \geq 0)$;

з) $x_k = \dfrac{(-1)^k x^{2k+1}}{(2k+1)!}$ $(k \geq 0)$.

Розв`язок г) Складемо рекурентне співвідношення для обчислення x_k

$x_0=x,\ x_k=x_{k-1}\dfrac{x}{k},\ k=1,2...$

Передбачивши введення і виведення, отримаємо

<u>*алг*</u> *Рекур* <u>*це*</u>
    <u>*змін*</u> *x,y:дійсн;*
        i,n:нат;
<u>*поч*</u>
    взяти (*x,n*);
    y←1;i←0;
    <u>*повт*</u> *n* <u>*раз*</u>
        i←i+1;
        *y←y*x/i*
    <u>*кц*</u>*;*
    показати (*y*)
<u>*ка*</u>.

3.10. Числами Фібоначчі називається числова послідовність $\{F_n\}$, задана рекурентним співвідношенням другого порядку

$F_0=0;\ F_1=1;\ F_k=F_{k-1}+F_k$, k=2,3,...

Скласти алгоритм для обчислення F_n

<u>Розв`язок</u>.Якщо змінна *f* буде пробігати послідовність Фібоначчі, то двох додаткових змінних *sf* і *t* буде достатньо для позначення наступних двох чисел

<u>*Алг*</u> *Фібоначчі* <u>*це*</u>
    <u>*змін*</u> *n,f,sf,t:нат;*
<u>*поч*</u>
    взяти (*n*);
    f←0;sf←1;
    $\{f=F_0,\ sf=F_1\}$
    <u>*повт*</u> *n* <u>*раз*</u>
        t←f+sf;
        f←sf;sf←t
    <u>*кц*</u>*;*
    $\{f=F_n,\ sf=F_{n+1}\}$
    показати (*f*)
<u>*ка*</u>.

3.11. Скласти алгоритм обчислення довільного члена послідовностей, які задані рекурентними співвідношеннями:

а) $x_n=x_{n-1}+x_{n-3}$, $x_0=x_1=1,\ x_2=2$, $n=3,4,...$;
б) $x_n=2x_{n-1}+3x_{n-2}$, $x_0=0,\ x_1=9$, $n=2,3,...$;
в) $x_n=x_{n-1}+x_{n-2}+x_{n-3}$, $x_0=x_1=1,\ x_2=6$, $n=3,4,...$;
г) $x_n=x_{n-1}+4x_{n-3}$, $x_0=x_1=x_2=2$, $n=3,4,...$;
д) $x_n=x_{n-1}*(x_{n-2}+1)$, $x_0=1,\ x_1=1$, $n=2,3,...$;
е) $x_n=\dfrac{x_{n-1}}{2}+\dfrac{3}{4}x_{n-2}$, $x_0=0,\ x_1=\dfrac{5}{8}$, $n=2,3,...$;

3.12. Скласти алгоритми для обчислення сум:

а) $S_n=1+\dfrac{1}{2}+\dfrac{1}{3}+\ldots+\dfrac{1}{n}$;

б) $S_n = 1 - \frac{1}{2} + \frac{1}{3} - \ldots + (-1)^{n-1}\frac{1}{n}$;

в) $S_n = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \ldots + \frac{1}{(n-1) \cdot n}$;

г) $S_n = -\frac{1}{3 \cdot 1} + \frac{1}{5 \cdot 3} - \frac{1}{7 \cdot 5} + \ldots + \frac{(-1)^n}{(2n+1) \cdot (2n-1)}$;

д) $S_n = 1-2-3+4-5-6+\ldots+(3n-2)-(3n-1)-3n$;

е) $S_n = 1+2+3+\ldots+n$; ж) $S_n = a+2a^2+3a^3+\ldots+na^n$;

з) $S_n = \sum_{i=1}^{n} 2^{n-i} i^2$; і) $S_n = \sum_{i=0}^{n} C_n^i$; к) $S_n = \sum_{i=0}^{n} iC_n^i$;

л) $S_n = \sum_{i=0}^{n} a^i b^{n-i}$; м) $S_n = \sum_{i=0}^{n} 2^i\, i!$; н) $S_n = \sum_{i=0}^{n} i!$;

о) $S_n = \sum_{i=0}^{n} \frac{1}{i!}$; п) $S_n = \frac{1!}{\frac{1}{2}} + \frac{2!}{\frac{1}{2}+\frac{1}{3}} + \ldots + \frac{n!}{\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n+1}}$.

<u>Вказівки</u>. Суму S_n обчислити за допомогою рекурентного співвідношення $S_0=0$, $S_k=S_{k-1}+a_k$, $k=1,2,\ldots n$, де a_k — k-тий доданок.

з) $S_i=2S_{i-1}+i^2$; і) $S_n=2^n$; л) $S_n = \frac{a^n - b^n}{a-b}$; м) $a_k=2ka_{k-1}$.

3.13. Скласти алгоритми для обчислення добутків:

а) $P_n = \prod_{i=1}^{n}\left(2+\frac{1}{i!}\right)$; б) $P_n = \prod_{i=1}^{n}\frac{i+1}{i+2}$;

в) $P_n = \prod_{i=1}^{n}\frac{1}{i+1!}$; г) $P_n = \prod_{i=1}^{n}\frac{1}{1+i^i}$.

<u>Вказівка</u>. Добуток P_n обчислити за допомогою рекурентного співвідношення $P_0=1$; $P_k=P_{k-1}*a_k$, $k=1,2,\ldots,n$, де a_k — k- тий множник.

3.14. Скласти алгоритми для обчислення ланцюгових дробів а) $$b_n = b + \cfrac{1}{b + \cfrac{1}{b + \ddots + \cfrac{1}{b}}};$$

$$б) \lambda_n = 2 + \cfrac{1}{6 + \cfrac{1}{10 + \ddots + \cfrac{1}{4n+2}}}.$$

$$в) x_{2n} = 1 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{2 + \ddots + \cfrac{1}{2}}}};$$

Вказівка. Використати рекурентні співвідношення

а) $b_0=b$, $b_k=b+\dfrac{1}{b_{k-1}}$, $k=1,2,...n;$

в) $b_0=4n+2$, $b_k=4(n-k)+2+\dfrac{1}{b_{k-1}}$, $k=1,2,...n.$

3.15. Скласти алгоритми для обчислення

а) многочлена Чебишова
$T_0(x)=1$, $T_1(x)=x,$
$T_n(x)=2xT_{n-1}(x) - T_{n-2}(x)$, $n=2,3,...;$

б) многочлена Ерміта
$H_0(x)=1$, $H_1(x)=2x,$
$H_n(x)=2xH_{n-1}(x) - 2(n-1)H_{n-2}(x)$, $n=2,3,...$
заданого степеню n в точці x.

3.16. Скласти алгоритм обчислення довільного елемента послідовностей, заданих рекурентними співвідношеннями

а) $v_0=1, v_1=0.3,$ $v_i=(i+2)v_{i-2}$, $i=2,3,...$

б) $v_0=v_1=v_2=1,$ $v_i=(i+4)(v_{i-1} - 1)+(i+5)v_{i-3}$, $i=3,4,...$

в) $v_0=v_1=0, v_2=\dfrac{3}{2}$, $v_i=\dfrac{i-2}{(i-3)^2+1}v_{i-1} - v_{i-2}v_{i-3}+1$, $i=3,4,...$

3.17. Скласти алгоритм обчислення довільного елемента послідовності v_n, визначеної системою співвідношень

$v_0=v_1=1,\qquad v_i=\dfrac{u_{i-1}-v_{i-1}}{|u_{i-2}+v_{i-1}|+2}\,,\qquad i=2,3,..;$

де $u_0=u_1=0\,,\qquad u_i=\dfrac{u_{i-1}-u_{i-2}v_{i-1}-v_{i-2}}{1+u_{i-1}^2+v_{i-1}^2},\quad i=2,3,\ldots\,.$

3.18. Скласти алгоритм для обчислення сум

а) $S_n=\sum\limits_{k=1}^{n}2^k a_k,$ де $a_1=0,\ a_2=1,\ a_k=a_{k-1}+k*a_{k-2},\quad k=3,4,\ldots;$

б) $S_n=\sum\limits_{k=1}^{n}\dfrac{3^k}{a_k},$ де $a_1=1,\ a_2=1,\ a_k=\dfrac{a_{k-1}}{k}+a_{k-2},\quad k=3,4,\ldots;$

в) $S_n=\sum\limits_{k=1}^{n}\dfrac{k!}{a_k},$ де $a_1=1,\ a_2=1,\ a_k=a_{k-1}+\dfrac{a_{k-2}}{2^k},\quad k=3,4,\ldots;$

г) $S_n=\sum\limits_{k=1}^{n}k!\,a_k$, де $a_1=0,\ a_2=1,\ a_k=a_{k-1}+\dfrac{a_{k-2}}{(k-1)!}\,,\quad k=3,4,\ldots;$

д) $S_n=\sum\limits_{k=1}^{n}\dfrac{a_k}{2^k},$ де $a_1=a_2=a_3=1,\ a_k=a_{k-1}+a_{k-3},\quad k=4,5,\ldots;$

е) $S_n=\sum\limits_{k=1}^{n}\dfrac{2^k}{k!}a_k,$ де $a_0=1,\ a_k=ka_{k-1}+\dfrac{1}{k},\quad k=1,2,\ldots$

Вказівка. Позначимо загальний член ряду через b_k. Послідовність a_k задається залежностями вигляду (R_1) для е), (R_2) для а)—г) та (R_3) для д); $S_k=g(S_{k-1},\ b_k)$. Значення a_k будуть обчислюватись за теоремами 1-2. Для обчислення послідовності S_k цикли доповнюються однією змінною.

Розв`язок д). Вводячи допоміжну змінну для обчислення 2^k і використовуючи рекурентні співвідношення $t_0=1,\ t_k=2t_{k-1},\ k=2,3,..$ для знаходження цієї системи отримаємо алгоритм

```
алг Сума це
    змін u , v, w, t, r, s: дійс;
        n: нат;
поч
    взяти (n);
    u←1;  v←1; w←1;  t←1;  s←0;
    { u=a1,  v=a2,  w=a3 }
    повт  n  раз
        t←t*2;
        s←s+u/t;
        r←u+w;  u←v;  v←w;  w←r
    кц;
    показати (s)
ка.
```

3.19. Скласти алгоритми для обчислення сум а) $S_n = \sum_{k=1}^{n} \frac{2^k}{a_k + b_k}$,

де $\begin{cases} a_1 = 0, a_2 = 1, \\ a_k = \frac{a_{k-1}}{k} + a_{k-2}b_k \end{cases}$, $\begin{cases} b_1 = 1, b_2 = 1, \\ b_k = b_{k-1} + a_{k-1} \end{cases}$; k=3,4,...;

б) $S_n = \sum_{k=1}^{n} \frac{a_k b_k}{(k+1)!}$,

де $\begin{cases} a_1 = u, \\ a_k = 2b_{k-1} + a_{k-1} \end{cases}$; $\begin{cases} b_1 = v, \\ b_k = 2a_{k-1}^2 + b_{k-1} \end{cases}$; k=2,3,...;

u,v — задані дійсні числа;

в) $S_n = \sum_{k=1}^{n} \frac{2^k}{(1 + a_k^2 + b_k^2)k!}$,

де $\begin{cases} a_1 = 1, \\ a_k = 3b_{k-1} + 2a_{k-1} \end{cases}$; $\begin{cases} b_1 = 1, \\ b_k = 2a_{k-1} + b_{k-1} \end{cases}$; k=2,3,...;

г) $S_n = \sum_{k=1}^{n} \left(\frac{a_k}{b_k}\right)^k$,

де $\begin{cases} a_0 = 1, a_1 = 2, \\ a_k = a_{k-2} + \frac{b_k}{2} \end{cases}$; $\begin{cases} b_0 = 5, b_1 = 5, \\ b^k = b_{k-2}^2 - a_{k-1} \end{cases}$; k=2,3,...;

д) $S_n = \sum_{k=1}^{n} \frac{a_k}{1 + b_k}$,

де $\begin{cases} a_0 = 1, \\ a_k = a_{k-1}b_{k-1} \end{cases}$; $\begin{cases} b_0 = 1, \\ b_k = a_{k-1} + b_{k-1} \end{cases}$; k=1,2,... .

Розв`язок д). Послідовності $\{a_n\}$ і $\{b_n\}$ задані рекурентним співвідношеннями першого порядку, проте залежність перехресна. Використовуємо по одній допоміжній змінній для кожної з послідовностей. Отримаємо алгоритм

алг Сума це
змін a, b, aa, bb, s:дійс;

n:нат;

<u>*поч*</u>

взяти (*n*);

a←*1; b*←*1; s*←*0,5;*

<u>*повт*</u> *n* <u>*раз*</u>

aa←*a*b; bb*←*a+b;*

a ←*aa; b*←*bb*; *s*←*s+a*/(1+*b*)

<u>*кц*</u>*;*

показати (*s*)

<u>*ка*</u>.

Неважко побачити, що змінну *bb* легко виключити.

3.20. Скласти алгоритми для обчислення добутків

а) $P_n = \prod_{k=0}^{n} \frac{a_k}{3^k}$, де $\begin{cases} a_0 = a_1 = 1, \quad a_2 = 3, \\ a_k = a_{k-3} + \dfrac{a_{k-2}}{2^{k-1}} \end{cases}$, k=3,4,...;

б) $P_n = \prod_{k=0}^{n} a_k b_k$,

де $\begin{cases} a_1 = 1, \\ a_k = \left(\sqrt{b_{k-1}} + a_{k-1}\right)/5, \end{cases}$ $\begin{cases} b_1 = 1, \\ b_k = 2b_{k-1} + 5a_{k-1}^2 \end{cases}$, k=2,3,... .

<u>*Розв`язок*</u> а) Послідовність $\{a_n\}$ задана рекурентним співвідношенням третього порядку. Для обчислення її довільного елемента a_k використаємо теорему 2. Тоді добуток P_n обчислюється за допомогою рекурентного співвідношення P_0=1, $P_k=P_{k-1}*a_k/z_k$, де z_k — *к*-тий степінь числа *3*, визначений із співвідношень z_0=1, $z_k=z_{k-1}*3$. Передбачивши також змінну *t* для обчислення степеня $2^{к-1}$, отримаємо алгоритм

<u>*алг*</u> *Добуток* <u>*це*</u>

<u>*змін*</u> *a0, a1, a2, w, p, z, t:дійсн;*

n:нат;

<u>*поч*</u>

взяти (*n*);

a0←*1; a1*←1; *a2*←3; *z*←1; *p*←1; *t*←2;

<u>*повт*</u> *n* <u>*раз*</u>

z←*z**3; *p*←*p*a1/z;*

t←*t**2;

w←*a0+a1/t;*

a0←*a1; a1*←*a2; a2*←*w*

<u>*кц*</u>*;*

показати (*p*)

<u>*ка*</u>.

3.3. Цикли за умовою

3.21. Для довільного цілого числа *m*>1 знайти найбільше ціле *k*, при якому $4^k<m$.

3.22. Для заданого натурального числа *n* одержати найменше число вигляду 2^r, яке перевищує *n*.

3.23. Визначити із скількох від`ємних чисел починається задана послідовність чисел.

3.24. Задана непорожня послідовність ненульових цілих чисел, за якою йде 0. Визначити кількість змін знаку в цій послідовності. Наприклад, у послідовності 1, –34, 8, 14, –5, 0 знак змінюється три рази.

```
Розв`язок .
алг Зміна_Знаку_2 це
        змін k:нат ;
                a,x:цiл;
поч
        взяти(x,a); k←0;
        {цикл за умовою повторення}
        поки a<>0  повт
                якщо a*x<0 то k←k+1 кр ;
                x←a; взяти (a)
        кц;
        показати(k)
ка.
```

Замінимо цикл з умовою повторення відповідно на цикл з умовою закінчення та цикл з виходом. Модифіковані алгоритми приймуть вигляд

```
алгоритм Зміна _Знаку_2 це
        змін k:нат;
                a,x:цiл;
поч
        взяти(x,a); k←0;
        {цикл з умовою закінчення}
        якщо a<>0 то
                повт
                        якщо a*x<0 то k←k+1 кр;
                        x←a; взяти(a)
                до a=0
                кц
        кр;
        показати(k)
ка.

алгоритм Зміна_Знаку_3 це
        змін k:нат;
                a,x:цiл;
поч
        взяти(x,a);k←0;
        {цикл з виходом}
        цикл
                якщо a=0 то вихід ;
                якщо a*x<0  то  k←k+1 кр;
                x←a; взяти(a)
        кц;
        показати(k)
ка.
```

3.25. Дана непорожня послідовність різних натуральних чисел, за якою слідує 0. Визначити порядковий номер найменшого з них.

3.26. Дана непорожня послідовність різних дійсних чисел, серед яких є хоча б одне від`ємне число, за якою йде 0. Визначити величину найбільшого серед від`ємних членів цієї послідовності.

3.27. Маємо дійсне число a. Скласти алгоритми обчислення:

а) серед чисел $1, 1+\frac{1}{2}, 1+\frac{1}{2}+\frac{1}{3}, \ldots$ першого, більшого за a;

б) такого найменшого n, що $1+\frac{1}{2}+\ldots+\frac{1}{n} > a.$

3.28. Скласти алгоритми обчислення:

а) номера найбільшого числа Фібоначчі, яке не перевищує задане число a;

б) номера найменшого числа Фібоначчі, яке більше заданого числа a;

в) суми всіх чисел Фібоначчі, які не перевищують 1000.

3.29. Дана непорожня послідовність з натуральних чисел, за якою йде 0. Обчислити суму тих з них, порядкові номери яких — числа Фібоначчі.

3.30. Скласти алгоритми для обчислення найменшого додатнього члена числових послідовностей, які задаються рекурентними співвідношеннями, та його номера

а) $x_n=x_{n-1}+x_{n-2}+100,$ $x_1=x_2=-99,$ $n=3,4,\ldots;$

б) $x_n=x_{n-1}+x_{n-2}+x_{n-3}+200,$ $x_1=x_2=x_3=-99,$ $n=4,5,\ldots;$

в) $x_n=x_{n-1}+x_{n-3}+100,$ $x_1=x_2=x_3=-99,$ $n=4,5,\ldots$

<u>Розв`язок</u> а) Оскільки послідовність x_n задана рекурентним співвідношенням другого порядку, то для обчислення довільного елементу послідовності потрібні три змінні.

Нехай змінна u пробігає послідовність x_k, k=1,2,....Тоді, в якості умови завершення циклу будемо розглядати умову u>0. Її заперечення — це умова u<=0 яка і розглядається як умова повторення циклу. Одержуємо алгоритм

```
алгоритм Мінім_елемент це
    змін u,v,w:ціл;
поч
    u←–99,v←–99; {u=x1, v=x2}
    поки u<=0 повт
        w←u+v+100;u←v;v←w
    кц;
    показати(u)
ка.
```

3.31. Скласти алгоритм, який з`ясовує, чи входить задана цифра до запису заданого натурального числа.

3.32. Скласти алгоритм «обернення» (запису в оберненому порядку цифр) заданого натурального числа.

<u>Вказівка.</u> Для побудови числа використати рекурентне співвідношення $y_0=0,\ y_i=y_{i-1}*10+a_i$, де a_i — наступна цифра числа n при розгляді цифр справа наліво.

3.33. Скласти алгоритм, який визначає потрібний спосіб розміну будь-якої суми грошей до 99 коп. за допомогою монет вартістю 1, 2, 3, 5, 10, 15, 20, 50 коп.

3.34. Скласти алгоритми наближеного обчислення суми всіх доданків, абсолютна величина яких не менше ε>0 :

а) $y = \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots;$

б) $$y = cos\ x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - ...;$$

в) $$y = shx = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \ldots;$$

г) $$y = chx = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots;$$

д) $$y = e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \ldots;$$

е) $$y = \ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots (|x| < 1);$$

ж) $$y = \frac{1}{1 + x} = 1 - x + x^2 - x^3 + \ldots (|x| < 1);$$

з) $$y = \ln\frac{1 + x}{1 - x} = 2 \cdot \left[\frac{x}{1} + \frac{x^3}{3} + \frac{x^5}{5} + \ldots\right] \ (|x| < 1);$$

і) $$y = \frac{1}{(1 + x)^2} = 1 - 2 \cdot x + 3 \cdot x^2 - \ldots (|x| < 1);$$

к) $$y = \frac{1}{(1 + x)^3} = 1 - \frac{2 \cdot 3}{2} \cdot x + \frac{3 \cdot 4}{2} \cdot x^2 - \frac{4 \cdot 5}{2} \cdot x^3 + \ldots (|x| < 1);$$

л) $$y = \frac{1}{1 + x^2} = 1 - x^2 + x^4 - x^6 + \ldots (|x| < 1);$$

м) $$y = \sqrt{1 + x} = 1 + \frac{1}{2} \cdot x - \frac{1}{2 \cdot 4} \cdot x^2 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6} \cdot x^3 - \ldots (|x| < 1);$$

н) $$y = \frac{1}{\sqrt{1 + x}} = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \ldots (|x| < 1);$$

о) $$y = \arcsin x = x + \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} + \ldots (|x| < 1).$$

Вказівка. Суму y обчислювати за допомогою рекурентного співвідношення $S_0=0$, $S_k=S_{k-1} + a_k$, k=1,2,... , де a_k — k–тий доданок, для обчислення якого також складається рекурентне співвідношення. В якості умови повторення циклу розглядається умова $|a_k| \geq \varepsilon$.

Розв`язок в) Рекурентне співвідношення для знаходження a_k має вигляд $a_1=x$, $a_{k+1}=a_k*x^2/(2*k*(2*k+1))$, $k=1,2,\ldots$ Передбачивши захищене введення для завдання ε, одержимо наступний алгоритм

```
алгоритм Сінус_гіпербол це
        змін k:цiл;
                x,y,z,a,eps:дійсн;
поч
        повт
                взяти(x,eps)
        до eps>0
        кц;
        k←0;a←x;z←x*x;y←0;
        поки abs(a)>=eps повт
                k←k+1;y←y+a;a←a*z/(2*k*(2*k+1))
        кц;
        показати(y)
ка.
```

3.35. Маємо дійсні числа x,ε ($x\neq 0$,$\varepsilon>0$). Обчислити з точністю ε нескінченну суму і вказати кількість врахованих доданків.

а) $\sum_{k=0}^{\infty}\frac{x^{2k}}{2k!}$; б) $\sum_{k=0}^{\infty}\frac{(-1)^k x^k}{(k+1)^2}$;

в) $\sum_{k=0}^{\infty}\frac{x^{2k}}{2^k k!}$; г) $\sum_{k=0}^{\infty}\frac{(-1)^k x^{2k+1}}{k!(2k+1)}$.

3.36. Маємо ціле $n>2$. Скласти алгоритм для обчислення всіх простих чисел з діапазону $[2,n]$.

Розв`язок. Нехай змінна m приймає послідовно значення цілих чисел з діапазону $[2,n]$. Тоді m буде простим числом , якщо воно не має дільників в діапазоні $[2,\sqrt{m}]$. Враховуючи цю обставину, складаємо алгоритм

```
алгоритм Прості_числа це
        змін k,n,m:нат;
                p:бул;
поч
        повт
                взяти(n);
        до n>2
        кц;
        m←2;
        поки m<=n повт
                p←n;k←2;
                поки p&(k*k<=n) повт
                        p←n mod k<>0;
                        якщо p то k←k+1 кр
                кц;
                якщо p то показати(m,' просте') кр;
                m←m+1
        кц
ка.
```

3.37. Скласти алгоритм друку всіх простих дільників заданого натурального числа.

3.38. Скласти алгоритм, який визначає чи є задане натуральне число n досконалим, тобто рівним сумі всіх своїх (додатніх) дільників, крім самого цього числа (наприклад, число 6 — досконале: 6=1+2+3).

Вказівка. Шукаємо суму *S* всіх дільників заданого числа *n*. Якщо *S*=*n,* то число, яке перевіряємо, є досконалим. Перша ідея полягає в знаходженні дільників числа *n* в діапазоні [1*, n div* 2]. У відповідності з другою ідеєю пошук ведеться тільки між 1 та $\sqrt{n}$ і якщо дільник знайдений, то до суми *S* додаються як дільник, так і частка.

3.39. Дано натуральне число *k* . Скласти алгоритм одержання *к*-тої цифри послідовності

а) 110100100010000 ... , в якій виписані підряд степені 10;

б) 123456789101112 ... , в якій виписані підряд всі натуральні числа;

в) 149162536 ... , в якій виписані підряд квадрати всіх натуральних чисел;

г) 01123581321 ... , в якій виписані підряд всі числа Фібоначчі.

3.40. Скласти алгоритм знаходження кореня рівняння *tg x*=*x* на відрізку [0,001;1,5] із заданою точністю ε, використовуючи метод ділення відрізку навпіл.

Розв`язок.

```
алг корінь це
        змін x,l,r,eps:дійсн;
поч
        повт
                взяти (eps)
        до eps>0;
        l ←0.001; r ←1.5;
        {лівий та правий кінці відрізка з коренем: tg(l)<l,tg(r)>r}
        повт
                x ←(l+r)/2; {середина відрізка [l,r]}
                якщо sin(x)/cos(x)<x то
                        l ←x {[l,r]←[x,r]}
                інакше
                        r ←x {[l,r]←[l,x]}
                кр
        до l-r<eps
        кц;
        x ← (l+r)/2; {корінь — середина останнього відрізка [l,r]}
        показати (x)
ка.
```

3.41. Знайти корінь рівняння $x^3 + 4x^2 + x - 6 = 0$, який міститься на відрізку [0,2], з заданою точністю ε.

Вказівка. Одним з методів розв`язування рівняння є метод хорд, який полягає в обчисленні елементів послідовності

$$u_0 = a;$$

$$u_n = u_{n-1} - \frac{y(u_n - 1)}{y(b) - y(u_{n-1})} \cdot (b - u_{n-1})$$

до виконання умови $|u_n - u_{n-1}| < \varepsilon_0$. В умовах нашої задачі *a*=0, *b*=2, $y(x) = x^3 + 4x^2 + x - 6$.

3.4. Цикли з лічильником

3.42. Задані натуральне число *n,* дійсні числа $a_1, a_2, \ldots, a_n$. Скласти алгоритм для знаходження:

а) *max*($a_1, a_2, \ldots, a_n$); б) *min*($a_1, a_2, \ldots, a_n$);

в) *max*($a_2, a_4, \ldots$); г) *min*($a_1, a_3, \ldots$);

д) *min*($a_2, a_4, \ldots$)+*max*($a_1, a_3, \ldots$);

е) *max*($|a_1|, \ldots, |a_n|$); ж) *max*($-a_1, a_2, -a_3, \ldots, (-1)^n a_n$;

з) $\left(min\left(a_1, \ldots, a_n\right)\right)^2 - min\left(a_1^2, \ldots, a_n^2\right)$.

Розв`язок а) Позначивши через *A* чергове число, яке ми вводимо, а через *Min* — *min*($a_1, a_2, \ldots, a_n$), складемо слідуючий алгоритм:

```
Алг Мінімум це
    змін   i,N:нат;
           A,Min:дійсн;
поч
    взяти(N); взяти(A); Min←A;
    для i←1 до N повт
        взяти(A);
        якщо A<Min то Min←A  кр
    кц;
    показати(Min)
ка.
```

3.43. Дано натуральне число n, цілі числа $a_1, a_2, \ldots, a_n$. Скласти алгоритм знаходження

а) min($a_1, 2a_2, \ldots, na_n$);

б) min($a_1 + a_2, \ldots, a_{n-1} + a_n$);

в) max($a_1, \ldots, a_1 a_2 \ldots a_n$);

г) кількості парних серед $a_1, a_2, \ldots, a_n$;

д) кількості повних квадратів серед $a_1, a_2, \ldots, a_n$;

е) кількості квадратів непарних чисел серед $a_1, a_2, \ldots, a_n$.

Розв`язок. в). Позначимо через *A* — чергове число, що вводиться, *P* — добуток введених чисел $a_1, a_2, \ldots$, що передують *A*, включаючи саме це число, Max — шуканий максимум. Складемо алгоритм у вигляді:

```
Алг Max_P це
    змін   N,i :нат;
           A,P,Max :ціл;
поч
    взяти(N);
    взяти(A);
    P←A; Max←A;
    для i←2 до N повт
        взяти(A); P←P*A;
        якщо P>Max то Max←P кр
    кц;
    показати(Max)
ка.
```

3.44. Дано натуральне число *n*. Скласти алгоритм обчислення факторіала *y=n!*, використовуючи

а) суцільний зростаючий; б) суцільний спадний цикли з лічильником.

3.45. Скласти алгоритм обчислення подвійного факторіала натурального числа *n* *y=n!!*.

Розв`язок. За означенням

$$n!!=\begin{cases}1\cdot3\cdot5\cdot\ldots\cdot n\text{, якщо } n \text{ - непарне,}\\ 2\cdot4\cdot6\cdot\ldots\cdot n\text{, якщо } n \text{ - парне.}\end{cases}$$

В обох випадках ми маємо як співмножники всі члени спадної арифметичної прогресії з різницею -2, які містяться між *n* та 1. Звідси

```
Алг Факт2 це
        змін k,n :нат;
                y:дійсн;
поч
        взяти(n);
        y←1;
        для k←n до 1 через -2 повт
                y←y*k
        кц;
        показати(y)
ка.
```

розв`язує дану задачу. Зауважимо, що при непарному *n* останнім значенням *k* буде 1, а при парному — 2. Приклад виконання алгоритму наведений в табл. 3.2

Таблиця 3.2. Трасувальна таблиця до завдання 3.45.

	n	*y*	*k*
взяти(n)	5		
y←1		1	
для k←5 до 1 через -2 повт			
k=5			5
y←y* k		5	
k=3			3
y←y* k		15	
k=1			1
y←y* k		15	
кц			
показати(y)		15	

3.46. Скласти алгоритми обчислення факторіалів:

а) *y=(2n)!!*; б) *y=(2n+1)!!*; в) *y=n!n!!(n+1)!!*

3.47. Задане натуральне число *n*. Скласти алгоритми обчислення добутків

а) $P=\left(1+\frac{1}{1^2}\right)\left(1+\frac{1}{2^2}\right)\ldots\left(1+\frac{1}{n^2}\right)$; б) $P=\left(1-\frac{1}{2^2}\right)\left(1-\frac{1}{3^2}\right)\ldots\left(1-\frac{1}{n^2}\right)$, $n>2$

3.48. Скласти алгоритм друку таблиці значень функції *y=sin*(*x*) на відрізку [0,1] з кроком *h*=0.1.

3.49. Скласти алгоритм визначення кількості тризначних натуральних чисел, сума цифр яких дорівнює *n* (*n*>=1). Операцію ділення не використовувати.

3.50. Дано *n* цілих чисел. Скласти алгоритм, що визначає, скільки з них більші за своїх «сусідів», тобто попереднього та наступного чисел.

3.51. Задані натуральне число *n*, дійсні числа $y_1, \ldots, y_n$. Скласти алгоритм визначення

а) $\max(|z_1|, \ldots, |z_n|)$, де $z_i = \begin{cases} y_i \text{ при } |y_i| \le 2, \\ 0.5 \text{ у інших випадках} \end{cases}$;

б) $\min(|z_1|, \ldots, |z_n|)$, де $z_i = \begin{cases} y_i \text{ при } |y_i| \ge 1, \\ 2 \text{ у інших випадках} \end{cases}$;

в) $z_1 + z_2 + \ldots + z_n$, де $z_i = \begin{cases} y_i \text{ при } 0 < y_i < 10, \\ 1 \text{ у інших випадках} \end{cases}$.

*<u>Розв`язок</u>* <u>в)</u>. Позначивши через *S* шукану суму, а через *y* значення чергового введеного числа, подамо алгоритм розв`язку задачи в наступному вигляді:

```
Алг Сума_Z це
    змін i,n :нат;
        y,z,S :дійсн;
поч
    взяти(n); S←0;
    для i←1 до n повт
        взяти(y);
        якщо (y<10) & (y>0) то z←y
        інакше  z←1
        кр;
    S←S+z
    кц;
    показати(S)
ка.
```

4. НАЙПРОСТІШІ ВЛАСТИВОСТІ ПРОГРАМ

4.1. Дослідити цикл на скінченність

поки x>0 *повт* x ← x-1 *кц* .

Розв`язок. Необхідна умова скінченності циклу очевидно виконується. Розглянемо функцію $B(x)=[x]+1$. Оскільки $[x]\ge 0$ при $x>0$, то $x>0$ тягне $B(x)>0$. Оскільки $[x-1]=[x]-1$, то $B(x)=[x]+1>[x]=[x-1]+1=B(x-1)$. Отже $B(x)$ — обмежувальна функція і цикл скінченний.

4.2. Встановити умови скінченності циклів:

а) *поки* x<>0 *повт* x ← x-1 *кц;*

б) *поки* x>0 *повт* x ← x+1 *кц;*

в) *поки* x<=b *повт* x ← x+a *кц;*

г) *поки* x<>b *повт* x ← x+a *кц;*

д) *поки* x<y *повт* z ← x; x ← y; y ← z+2 *кц*.

4.3. Довести скінченність циклу в алгоритмі

Алг Circle_1 це
змін n: ціл;
поч
взяти(n);
поки n>1 повт
якщо n mod 2 *=0 то n* $\leftarrow$ *n div* 2
інакше n $\leftarrow$ *n*+1
кр
кц;
показати(n)
ка.

4.4. Довести нескінченність циклу в алгоритмі

Алг Circle_2 це
змін n: ціл;
поч
взяти(n);
поки n>1 повт
якщо n mod 2 *=0 то n* $\leftarrow$ *n div* 2+1
інакше n $\leftarrow$ *n*+1
кр
кц;
показати(n)
ка.

4.5. Встановити умови скінченності циклу у наступному фрагменті алгоритму
s$\leftarrow$ *0;*
поки n<>m повт
n $\leftarrow$ *n*+1*; s* $\leftarrow$ *s*+1
кц

4.6. Скласти алгоритм ділення натуральних чисел *n* і *m* націло з остачею:
$n=q*m+r,\ 0<=r<m.$
Розв`язок. Виберемо початкові значення $q=0$, $r=n$. В цьому випадку умова
$J=(n=m*q+r)\&(r>=0)$
очевидно істинна. Побудуємо цикл таким чином, щоб умова *J* була його інваріантом, а умову повторення циклу виберемо так, щоб його заперечення разом з інваріантом $\neg\alpha\ \&\ J$ давало б умову задачі. Очевидно, $\alpha=\neg\,(r<m) = (r>=m)$.

Будемо шукати дії, що зберігають умову *J* при умові α. Значення змінної *q* в результаті виконання цієї дії, очевидно, повинне збільшуватися. Розглянемо збільшення *q* на 1. Тоді *r* потрібно зменшити на величину $(q+1)*m-q*m=m$, щоб забезпечити збереження *J* .Таким чином, пара присвоєнь $q \leftarrow q+1;\ r \leftarrow r-m$ зберігає умову *J* при $r>=m$. Оскільки задача має зміст при натуральних *m* і невід`ємних цілих *n*, то передбачимо в ній захищене введення. Отримаємо алгоритм:

алгоритм Div це
змін m,n,q,r :ціл;
поч

__повт__

взяти (m,n)

__до__ (m>0)&(n>=0) ;

q← 0; r← n;

__поки__ r>=m __повт__

q← q+1; r← r-m

__кц__;

показати(q,r)

__ка__.

Спадною цілочисловою величиною, очевидно, є значення змінної r, оскільки $m>0$, а обмежувальною функцією — функція $B(r,m)=r-m$.

4.7. Скласти алгоритм обчислення найбільшого спільного дільника двох натуральних чисел *m, n* .

Розв`язок 1. Задача полягає в обчисленні функції, позначимо її НСД, двох натуральних аргументів: НСД(*m,n*). При $m=n$ обчислення НСД тривіальне, так як НСД$(m,m)=m$.

В силу адитивності функції НСД вона інваріантна відносно перетворень

$m \leftarrow m — n$ якщо $m>n$

$n \leftarrow n — m$ якщо $n>m$

а також дій $n \leftarrow n+m;$ $m \leftarrow m+n$.

Вибір потрібної дії серед перелічених вище необхідно здійснити одночасно з пошуком обмежувальної функції. Якщо за таку функцію взяти $B(m,n)=m+n$, а за умову повторення циклу умову $m<>n$, то функція $B(m,n)$ буде спадати під дією першої сукупності перетворень. Щоб привести цю функцію у відповідність з умовою скінченності, перейдемо до різниці $B1(m,n) = B(m,n) — 2НСД(m,n)$. Отримаємо алгоритм

__алгоритм__ НСД __це__

__змін__ m,n :нат;

__поч__

__повт__

взяти (m,n)

__до__ m>0 & n>0;

__поки__ m<>n __повт__

__якщо__ m>n __то__ m ← m— n

__інакше__ n ← n — m

__кр__

__кц__;

показати (m)

__ка__.

Показати, що функція $B1(m,n)$ дійсно є обмежувальною функцією отриманого циклу.

Розв`язок 2. Другий спосіб полягає у розгляді іншої умови тривіальності обчислення НСД, наприклад НСД$(m,0)=m$. Функція НСД інваріантна відносно перетворень $m \leftarrow n;$ $n \leftarrow m \ mod \ n$. Якщо за обмежувальну функцію взяти $B(n)=n$, а за умову повторення циклу умову $n<>0$, то функція $B(n)$ буде спадати під дією вказаних перетворень. Отримаємо алгоритм

__алгоритм__ НСД __це__

__змін__ m,n :нат:

__поч__

```
повт
        взяти (m,n)
до m>0 & n>0 ;
        поки n<>0 повт
        l← m; m← n; n← l mod n
кц;
показати (m)
ка.
```

4.8. Скласти алгоритм обчислення добутку двох натуральних чисел, користуючись операціями додавання та ділення навпіл.

4.9. Скласти алгоритм обчислення степеня $y=x^n$ натурального показника, використовуючи тільки операції множення та ділення навпіл.

Розв`язок. Розглянемо функцію $y=zx^k$. При $z=1$ і $k=n$ вона дає шукану величину $y=x^n$. При $k=0$ функція $y=zx^k$ обчислюється тривіально : $y=z$. Ця функція є інваріантною відносно перетворень

$x \leftarrow x \leftarrow x,\ k \leftarrow k/2;$

$z \leftarrow z \leftarrow x,\ k \leftarrow k\text{-}1.$

Першу пару дій можна застосовувати при парному k, другу доречно застосовувати при непарному k. Умова закінчення $k=0$. Отримаємо алгоритм

```
алг  степінь   це
        змін n,k: нат; x,z : дійсн;
поч
        взяти(n,x);
        z← 1; k← n;
        поки k <> 0 повт
                якщо k mod 2=0 то
                        x← x*x; k← k div 2
                інакше
                        z← z* k; k← k-1;
                кр
        кц;
        показати(z)
ка.
```

Обмежувальну функцію побудувати самостійно.

5. Прості типи даних

5.1 Арифметика наближених обчислень

5.1. Обчислити значення функції *round*

а) *round*(35.774873, 5, 10);

б) *round*(0.0000237927, 4, 10);

в) *round*($(110.0100110)_2$, 7, 2);

г) *round*($(3.77673)_8$, 3, 8);

д) *round*(1345271, 4, 10);

е) *round*($(21.4322)_5$, 3, 5).

5.2. Підібрати приклади, що показують порушення співвідношень

а) $x +_p (y +_p z)=(x +_p y) +_p z;$

б) $x*_p (y*_p z)=(x *_p y) *_p z;$

в) $x*_p (y +_p z)=x*_p y +_p x*_p z$

при p=8.

Відповідь

а) x=11111113., y=-111111., z=7.5111111.;

5.3. Вкажіть x такий, що $(x +_p x) /_p 2$ <> x при p=8.

5.4. Чи виконується рівність $x /_p y=x*_p (1 /_p y)$ для всіх дійсних чисел?

5.5. Доведіть або спростуйте співвідношення

а) $0 —_p (0 —_p x)=x;$

b) $1 /_p (1 /_p x)=x$.

5.6. Чи вірно, що u <= $(u +_p v) /_p 2$ <= v при u <= v ?

Розв`язок. В звичайній арифметиці при u <= v справедлива нерівність

$$u=\frac{1}{2}(u+u) <= \frac{1}{2}(u+v) <= \frac{1}{2}(v+v)=v.$$

В арифметиці наближених обчислень $x=(x +_p x) /_p 2$ справедливе не завжди, що призводить до порушення нерівностей

u <= $(u +_p v) /_p 2$ <= v при u <= v.

Достатньо взяти (p=8) u=9.2222222, v=9.2222223 .

5.7. Нехай p=4. Послідовність a_n (n>=1) задана співвідношенням a_1=1000, a_n=0.5 (n>=2). Скласти алгоритм обчислення суми $a_1 + a_2 + ... + a_n$ з точністю ε=1.

5.8. Нехай p=4. Послідовність $a_i(i \geq 1)$ задана співвідношенням $a_i=5000+i$. Скласти алгоритм обчислення середнього арифметичного $\frac{1}{n}(a_1+...+a_n)$ з точністю ε=1.

5.9. Розв`язати квадратне рівняння $ax^2+bx+c=0$, використовуючи наближені обчислення з чотирма ($p$=4) і шістьма ($p$=6) значущими цифрами. Порівняти отримані корені один з одним і з точним розв`язком. Знайти спосіб обчислення і скласти алгоритм, що дає результат з відносною похибкою δ.

Вказівка: Скористатися теоремою Вієта.

5.10. Якщо в системі рівнянь $\begin{cases} x + \sqrt{2}y = \sqrt{3} \\ \sqrt{2}x + 2y = \sqrt{6} \end{cases}$.замінити радикали їх наближеннями, то що можна сказати про розв`язок отриманої системи методом Крамера? Що вийде, якщо збільшити точність наближення радикалів?

5.2 Натуральний та цілий типи даних

5.11. Скласти алгоритм обчисення найбільшого спільного дільника

а) двох натуральних чисел;

б) двох цілих чисел

за допомогою метода Евкліда.

Розвязок 1а.

```
алгоритм НСД це
        змін M,N,V,U: нат;
поч
        взяти (M,N);
        U←M; V←N;
        поки U<>V повт
                якщо U<V то V←V-U
                інакше U←U-V
                кр
        кц;
        показати (U)
ка.
```

5.12. Скласти алгоритм обчислення найменшого спільного кратного двох натуральних чисел *m* і *n* та порівняти області визначеня алгоритма, побудованого за формулою

$$НСК(m,n)=\frac{m*n}{НСД(m,n)}$$

і алгоритма за формулою

$$НСК(m,n)=\frac{m}{НСД(m,n)}*n \, .$$

5.13. Дано натуральні числа *m* і *n*. Знайти такі натуральні числа *p* і *q*, не виключаючи спільних дільників, що $p/q = m/n$.

5.14. Знайти всі прості нескоротні дроби, що містяться між 0 і 1, знаменники яких не перевищують 7 (дріб задається двома числами- чисельником та знаменником).

5.15. Дано натуральне число *n*. Скласти алгоритм знаходження всіх таких натуральних *q*, що *n* ділиться на q^2 і не ділиться на q^3.

5.16. Дано натуральні числа *m* і *n*. Скласти алгоритм знаходження всіх їх натуральних спільних кратних , менших добутка *m*n*.

5.17. Дано натуральні числа *m* і *n*. Скласти алгоритм знаходження всіх їх спільних дільників.

5.18. Два натуральних числа називаються дружніми, якщо кожне з них дорівнює сумі всіх дільників іншого, крім самого цього числа. Скласти алгоритм знаходження всіх пар дружніх чисел, що лежать в діапазоні від 200 до 300.

5.19. Дано натуральне число *n*. Скласти алгоритм знаходження всіх піфагорових трійок натуральних чисел, кожне з яких не перевищує *n*, тобто всіх таких трійок натуральних чисел *a, b, c*, що $a^2 + b^2 = c^2$ $(a <= b <= c <= n)$.

5.20. Натуральне число з *n* цифр є числом Армстронга, якщо сума його цифр, піднесених до *n*-того степеня, дорівнює самому числу (наприклад, $153=1^3 + 3^3 + 5^3$). Скласти алгоритм знаходження всіх чисел Армстронга, що складаються з двох, трьох та чотирьох цифр.

5.21. Дано натуральне число *n*. Скласти алгоритм, що визначає, чи можна подати його у вигляді суми двох квадратів натуральних чисел. Якщо це можливо, то

а) вказати пару *a, b* таких натуральних чисел , що $n=a^2+b^2$;

б) вказати пару *a, b* таких натуральних чисел , що $n=a^2+b^2$,*a>=b.*

5.22. Дано натуральне число *n*. Скласти алгоритм,що дозволяє

а) переставити першу та останню цифри числа *n*;

б) приписати по одиниці в початок та кінець запису числа *n*;

в) одержати суму *m* останніх цифр числа *n*.

5.23. Дано натуральне число *n*. Скласти алгоритм, що визначає серед чисел 1, ..., *n* всі такі, запис яких співпадає з останніми цифрами запису їх квадратів (наприклад, $6^2=36,\ 25^2=625$ і т.д.).

<u>*Розвязок.*</u>

Очевидно, що якщо запис числа *x* співпадає з останніми цифрами його квадрата x^2, то різниця x^2-*x* ділиться на 10 у степені , що дорівнює кількості цифр у числі *x*. Враховуючи цю обставину, запишемо алгоритм у вигляді

```
алгоритм Квадрат це
    змін x,y,n,k: ціл;
поч
    взяти(n);
    для x ←1 до n повт
        k←1; y←x;
        поки y<>0 повт
            y←y div 10; k←k*10
        кц;
        якщо (x*x-x) mod k=0 то показати (x) кр
    кц
ка.
```

5.24. Назвемо натуральне число паліндромом , якщо його запис читається однаково зліва направо і справа наліво (наприклад, 1, 393, 4884). Скласти алгоритм, що визначає, чи є задане натуральне число *n* паліндромом.

5.25. Дано натуральне число *n*. Скласти алгоритм знаходження всіх менших *n* натуральних чисел

а) які при піднесенні до квадрату дають паліндром;

б) паліндромів, які при піднесенні до квадрату дають також паліндром.

<u>*Розв`язок.*</u> а) Для *x*=1, ..., *n* будемо порівнювати значення $y=x^2$ з числом *m*, що є записом числа *y* у зворотньому порядку.

```
алгоритм Пал_квадрат це
    змін n,x,y,m,p,q,a : ціл;
поч
    взяти (n);
    для x←1 до n повт
        y←x*x; m←0; p←y;
        поки p<>0 повт
            a←p mod 10; p←p div 10; m←m*10+a
        кц;
        якщо y=m то показати(x) кр
    кц
ка.
```

5.26. Задані натуральне число n, цілі числа $a, x_1, \ldots, x_n$. Скласти алгоритм визначення номера члена послідовності, який дорівнює a. Якщо такого немає, то відповіддю має бути число 0.

5.27. Задані натуральне число n, цілі числа $a, x_1, \ldots, x_n$. Якщо в послідовності $x_1, \ldots, x_n$ є хоча б один член, що дорівнює a, то отримати суму всіх членів, які слідують за таким членом послідовності, в протилежному випадку відповіддю має бути відповідне текстове повідомлення.

5.28. Задані натуральне число n, цілі числа $a_1, \ldots, a_n$. Отримати суму додатніх, кількість від'ємних і число нульових членів послідовності $a_1, \ldots, a_n$.

5.29. Задані натуральні числа n і p, цілі числа $a_1, \ldots, a_n$. Скласти алгоритм обчислення добутку членів послідовності, кратних p.

Розв'язок.

```
алгоритм  Добуток це
        змін n,p,i,k:нат;
                s,a:ціл;
поч
        взяти (n,p);
        s←1; k←1;
        повт n раз
                взяти(a);
                якщо a mod p=0 то s←s*a; k←k+1 кр
        кц;
        якщо k=1 то показати (‘таких немає’)
        інакше показати(s)
        кр
ка.
```

5.30. Дано натуральне число n. Скласти алгоритм, який з`ясовує, чи можна подати $n!$ у вигляді добутку трьох послідовних чисел.

5.3 Дійсний тип даних

5.31. Задані рекурентні співвідношення

$$a_0 = 1, c_0 = 1 - x,$$

$$a_i = a_{i-1} * (c_{i-1} + 1), c_i = c_{i-1}^2, i = 1,2,\ldots$$

Довести, що $\lim\limits_{n\to\infty} a_n = \frac{1}{x}, \ (0 < x < 2)$ та скласти алгоритм обчислення величини $\frac{1}{x}$, що не використовує операцію ділення

Вказівка. Скористатись співвідношеннями $c_n = c_0^{2^n}$ і $a_n = \frac{1 - c_n}{x}$,

останнє з яких одержується з рівностей

$$a_n = (1 + c_{n-1}) * \ldots * (1 + c_1) * (1 + c_0), \quad \frac{1 + c_{i-1}}{1 - c_i} = \frac{1}{1 - c_{i-1}}$$

5.32. Скласти алгоритм наближеного обчислення квадратного кореня $\sqrt{x}$, використовуючи рекурентні співвідношення

$$a_0 = x,\ c_0 = 1 - x,$$

$$a_i = a_{i-1} * (1 + \frac{1}{2} * c_{i-1}),\ c_i = \frac{1}{4} * c_{i-1}^2 * (3 + c_{i-1}),\ i = 1, 2, \ldots$$

та границю $\lim\limits_{n\to\infty} a_n = \sqrt{x}\ (0 < x < 2)$.

<u>*Вказівка*</u>. Використовуючи рівності

$$a_n = (1 + \frac{1}{2} * c_{n-1}) * (1 + \frac{1}{2} * c_{n-2}) * \ldots * (1 + \frac{1}{2} * c_0) * x,$$

$$1 + \frac{1}{2} * c_{i-1} = \frac{\sqrt{1 - c_i}}{\sqrt{1 - c_{i-1}}}$$

довести, що $a_n = \sqrt{x * (1 - c_n)}$.

5.33. Скласти алгоритм наближеного обчислення квадатного кореня $\sqrt{x}$ з заданою точністю, використовуючи рекурентні співвідношення

$$a_0 = \frac{x}{2},$$

$$a_i = \frac{1}{2} * (a_{i-1} + \frac{x}{a_{i-1}}),\ i = 1,2,...$$ та границю $\lim\limits_{ї\to\infty} a_n = \sqrt{x}$.

<u>*Вказівка*</u>. Скористатися рівністю

$$a_n - \sqrt{x} = \frac{1}{2 * a_{n-1}} * \left(a_{n-1} - \sqrt{x}\right)^2.$$

5.34. Користуючись розкладом в ряд Тейлора, скласти алгоритми обчислення функцій
а) $y=sin(x)$, б) $y=cos(x)$
з заданою точністю.

5.35. Скласти алгоритм обчислення інтегралу

$$\int_0^x e^{-x^2} dx = x - \frac{x^3}{3 \cdot 1!} + \frac{x^5}{5 \cdot 2!} - \frac{x^7}{7 \cdot 3!} + \ldots$$

з заданою точністю.

5.36. Скласти алгоритм наближеного обчислення кубічного кореня $\sqrt[3]{x}$, користуючись рекурентним співвідношенням

$$a_0 = 1,\ a_i = \sqrt[4]{x + a_{i-1}},\ i = 1,2,...$$

та границею $\lim\limits_{n\to\infty} a_n = \sqrt[3]{x}$.

Вказівка. Скористатись нерівністю

$$\left|a_n - \sqrt[3]{x}\right| < \left|a_1 - a_0\right| * \frac{q^n}{1-q}$$, де $q = \frac{1}{n}$.

5.37. Скласти алгоритм наближеного обчислення числа π, використовуючи добуток

$$\frac{2}{\pi} = \sqrt{\frac{1}{2}} * \sqrt{\frac{1}{2} + \frac{1}{2} * \sqrt{\frac{1}{2}}} * \sqrt{\frac{1}{2} + \frac{1}{2} * \sqrt{\frac{1}{2} + \frac{1}{2} * \sqrt{\frac{1}{2}}}} * \ldots$$

5.38. Скласти алгоритм наближеного обчислення золотого перетину c, використовуючи границю

$$c = \lim_{n \to \infty} \frac{F_n}{F_{n-1}},$$

де F_n-числа Фібоначчі.

Вказівка. Розглянути послідовність $c_n = \frac{F_n}{F_{n-1}}$ та знайти рекурентне співвідношення для c_n.

5.39. Скласти алгоритм обчислення з заданою точністю границь послідовностей, утворених за законами:

а) $a_n = \dfrac{n}{\sqrt{n^2+1} - \sqrt{n^2-1}};$

б) $a_n = \left(1 - \frac{1}{2}\right) * \left(1 - \frac{1}{3}\right) * \ldots * \left(1 - \frac{1}{n+1}\right);$

в) $a_n = \left(1 - \frac{1}{2!}\right) * \left(1 + \frac{1}{3!}\right) * \ldots * \left(1 + \frac{(-1)^n}{(n+1)!}\right).$

5.40. Скласти алгоритм наближеного обчислення кореня п'ятого степеню $\sqrt[5]{x}$, використовуючи рекурентні співвідношення

$$a_o = \begin{cases} \min(2x, 0.95), \text{ при } x <= 1 \\ \frac{x}{5}, \text{ при } 1 < x < 25 \\ \frac{x}{25}, \text{ у інших випадках} \end{cases}, \quad a_i = \frac{4}{5} * a_{i-1} + \frac{x}{5 * a_{i-1}^4}, i = 1, 2, \ldots$$

і границю

$\lim\limits_{n \to \infty} a_n = \sqrt[5]{x} (x > 0)$. Для першого члена a_i, що задовольняє умові $\frac{5}{4} x \left|a_{i+1} - a_i\right| < \varepsilon$, обчислити різницю $x - a_i^5$.

5.41. Скласти алгоритм наближеного обчислення границі послідовності, заданої рекурентним співвідношенням

$$\text{а) } a_0 = 0, a_i = \frac{a_{i-1}+1}{a_{i-1}+2}, i = 1,2...; \text{ б) } a_0 = 1, a_i = \frac{2 - a_{i-1}^3}{5}, i = 1,2,...$$

з заданою точністю.

5.42. Задане дійсне число *x*. Послідовність $a_1, a_2, \ldots$ утворена за наступним законом: $a_1 = x$; далі для $i=2,3...$ виконано :

$$\text{а) } a_i = \sqrt{\left|4 \cdot a_{i-1}^2 - 2x\right|}; \text{ б) } a_i = \frac{16+x}{1+\left|a_{i-1}^3\right|},$$

$$\text{в) } a_i = 2 \cdot a_{i-1} + \frac{x}{4+a_{i-1}^2}, \text{ г) } a_i = 3 + \frac{1}{2^i} cos^2\left(a_{i-1} - x\right).$$

Скласти алгоритм для обчислення границі послідовності a_n з заданою точністю.

5.43. Задані рекурентні співвідношення

$$x_1 = a, y_1 = b,$$

$$x_i = \frac{1}{2}\left(x_{i-1} + y_{i-1}\right), y_i = \sqrt{x_{i-1} \cdot y_{i-1}}, i = 2,3\ldots (a > b > 0)$$

Скласти алгоритм для обчислення першого члена послідовності x_i такого, що $|x_i - y_i| < \varepsilon$, $(\varepsilon > 0)$.

5.44. Скласти алгоритм наближеного обчислення числа π за формулою Грегорі

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots$$

5.45. Дано *n* дійсних чисел. Скласти алгоритм для знаходження номера того з них, яке найбільш близьке до цілого числа.

5.46. Задані дійсні числа *x,y* ($x>0$, $y>1$). Скласти алгоритм для обчислення цілого числа *k* (додатнього, відємного або рівного нулеві), що задовольняє умові $y^{k-1} \le x < y^k$.

5.47. Задана послідовність дійсних чисел $x_1, x_2, \ldots x_n$ (*n* заздалегідь невідоме), за якою слідує 0. Скласти алгоритм для обчислення суми

$$y = n \cdot x_1 + (n-1) \cdot x_2 + \ldots + 2 \cdot x_{n-1} + x_n.$$

5.48. Задана послідовність з *n* дійсних чисел $x_1, x_2, \ldots x_n$ ($n>=3$). Скласти алгоритм для обчислення

а) $\left(x_1 + 2 \cdot x_2 + x_3\right) \cdot \left(x_2 + 2 \cdot x_3 + x_4\right) \cdot \ldots \cdot \left(x_{n-2} + 2 \cdot x_{n-1} + x_n\right)$;
б) $\left(x_1 + x_2 + x_3\right) \cdot x_2 + \left(x_2 + x_3 + x_4\right) \cdot x_3 + \ldots + \left(x_{n-2} + x_{n-1} + x_n\right) \cdot x_{n-1}$

5.49. Задані натуральне число n, дійсні числа $a_1, \ldots, a_n$. Скласти алгоритм, що визначає у послідовності $a_1, \ldots, a_n$ кількість сусідств

а) двох додатніх чисел ; б) двох чисел різних знаків.

5.50. Скласти алгоритм обчислення суми

$$1 - \frac{1}{2} + \frac{1}{3} - \ldots + \frac{1}{9999} - \frac{1}{10000}$$

наступними чотирма способами:

а) послідовно зліва направо;

б) послідовно зліва направо обчислюється окремо сума додатніх та сума від'ємних доданків, потім ці суми складаються;

в) послідовно справа наліво;

г) послідовно справа наліво обчислюється окремо сума додатніх та сума від'ємних доданків, потім ці суми складаються.

Порівняти результат із даним значенням суми (30 десяткових знаків).

0.693097183059945296917232371458...

Який спосіб дає найменшу похибку обчислень і чому?

5.4 Символьний тип даних

5.51. Скласти алгоритм виведення великих літер латинського алфавіту в алфавітному порядку.

<u>*Розв'язок.*</u> Алгоритм виведення буде мати вигляд

<u>*алгоритм*</u> ABC <u>*це*</u>
    <u>*змін*</u> *S: симв;*
<u>*поч*</u>
    <u>*для*</u> *S*←*'A'* <u>*до*</u> *'Z'* <u>*повт*</u>
        показати(*S*)
    <u>*кц*</u>
<u>*ка*</u>.

5.52. Визначити, чи є даний символ латинською літерою (великою або маленькою), цифрою або ні тим ні іншим.

Надалі в цьому розділі під «текстом» будемо розуміти завдання непорожньої послідовності символів $c_1, \ldots, c_n$ (n заздалегідь не відоме), що передують символу '.' (тобто, в текст крапка не входить).

5.53. Визначити, яка з двох заданих літер у даному тексті зустрічається частіше.

5.54. Визначити, чи входить до даного тексту кожна з літер слова *key*.

5.55. Перевірити, чи правильно в заданому тексті розставлені круглі дужки (тобто, чи знаходиться справа від кожної відкриваючої дужки відповідна їй закриваюча дужка, а зліва від кожної закриваючої — відповідна їй відкриваюча).

Відповідь — «так» або «ні».

<u>*Розв'язок.*</u>

<u>*алг*</u> Дужки <u>*це*</u>
    <u>*змін*</u> *c*: *симв*;

```
                    k: ціл;
поч
          взяти(c); k ←0;
          поки (c<>'.')&(k>=0) повт
                    якщо c='(' то k←k+1
                    інякщо c=')' то k←k-1 кр
                    взяти(c)
          кц;
          якщо k=0 то показати('Так')
          інакше показати('Ні')
          кр
ка.
```

5.56. Визначити, чи є заданий текст правильним записом цілого числа (можливо зі знаком).
5.57. Надрукувати заданий текст:
а) виключивши з нього всі цифри і подвоївши знаки '+' та '-';
б) виключивши з нього всі знаки '+', безпосередньо за якими знаходиться цифра;
в) виключивши з нього всі літери '*в*', безпосередньо перед якими знаходиться літера '*с*';
г) замінивши в ньому всі пари '*ph*' на літеру '*f*';
д) виключивши з нього всі зайві пропуски, тобто з кількох, що йдуть підряд, залишити один.

Розв'язок б)

```
алг Виключеня це
          змін a,b:симв;
поч
          взяти(a,b);
          поки b<>'.' повт
                    якщо (a<>'+')∨(b<'0')∨(b>'9') то показати(a) кр;
                    a←b; взяти(b)
          кц;
          показати(a)
ка.
```

5.58. Дано текст, серед символів якого є принаймні одна кома. Знайти номер
а) першої по порядку коми;
б) останньої по порядку коми.

5.59. Виключити з заданого тексту групи символів, які знаходяться між '(' та ')'. Самі дужки теж мають бути виключені. Вважається, що дужки розставлено правильно (парами) та всередині кожної пари дужок немає інших дужок.

5.60. Заданий текст, серед символів якого міститься двокрапка ':'. Отримати всі символи, розміщені
а) до першої двокрапки включно;
б) після першої двокрапки;
в) між першою і другою двокрапкою. Якщо другої двокрапки немає, то отримати всі символи, розміщені після єдиної двокрапки.

5.61. Задана непорожня послідовність непорожніх слів з латинських літер. Словами називаються групи символів, які розділені одним чи кількома пропусками та не містять пропусків всередині себе. Визначити кількість слів, які:
а) містяться в даній послідовності;
б) починаються з заданої літери;
в) закінчуються заданою літерою;
г) починаються і закінчуються однією літерою;
д) містять принаймні одну задану літеру;
є) містять рівно три заданих літери.

5.62. В умовах попереднього завдання:

а) знайти довжину самого короткого слова;

б) підрахувати кількість входжень заданої літери в останнє слово даної послідовності.

5.63. Заданий текст надрукувати по рядках, розуміючи під рядком або наступні 60 символів, якщо серед них немає коми, або частину тексту до коми включно.

5.64. Використовуючи тільки символьне введення, ввести непорожню послідовність цифр, перед якою може знаходитись знак '+' чи '-' і за якою знаходиться крапка, і отримавши відповідне ціле число, присвоїти його цілій змінній *m*.

<u>*Розв'язок*</u>:

```
Алг Введення_цілого це
    змін s:симв;
        k0,m,zn:ціл;
поч
    взяти(s); zn←1;
    якщо s='-' то zn← -1; взяти(s)
    інякщо s='+' то взяти(s) кр
    m←0; k0←ord('0');
    поки s<>'.' повт
        m←m*10+ord(s) — k0;
        взяти(s)
    кц;
    m←zn*m;
    показати(m)
ка.
```

5.65. Використовуючи тільки символьне виведення, вивести на друк значення цілої змінної *k* (знак '+' не друкувати).

5.66. Використовуючи тільки символьне введення, ввести задане дійсне число із знаком, записане у форматі з фіксованою крапкою, за яким знаходиться символ '?'. Присвоїти його дійсній змінній *x*.

5.67. Використовуючи тільки символьне виведення, надрукувати дійсне число *x* у наступній формі:

$$\pm 0.d_1d_2...d_9E \pm p_1p_2,$$

де d_i, p_j — цифри, причому $d_1 \neq 0$, якщо $x \neq 0$.

5.68. Задана послідовність символів, яка має вигляд:

$$d_1 \pm d_2 \pm ... \pm d_n$$

(d_i-цифри, $n>1$), за якою знаходиться крапка. Обчислити значення цієї алгебраїчної суми.

5.69. Задане натуральне число *n*. Надрукувати в трійковій системі числення цілі числа від 0 до *n*.

5.70. В заданий текст входять тільки цифри та літери. Визначити, чи задовольняє він наступній властивості:

а) текст є десятковим записом числа, кратного 9 (6, 4);

б) текст починається з деякої ненульової цифри, за якою знаходяться тільки літери і їх кількість дорівнює числовому значенню цієї цифри;

в) текст містить (крім літер) тільки одну цифру, причому її числове значення дорівнює довжині тексту;

г) сума числових значень цифр, які входять в текст, дорівнює довжині тексту;

д) текст співпадає з початковим (кінцевим, будь-яким) відрізком ряду 0123456789;

е) текст складається тільки з цифр, причому їх числові значення складають арифметичну прогресію (наприклад, 3 5 7 9, 8 5 2, 2).

<u>*Розв'язок г)*</u>

```
Алг Сума_значень це
    змін s:симв;
        k,k0,m:ціл;
поч
    взяти(s); k0←ord('0');
    k←0; m←0;
    поки s<> '.' повт
        k←k+1;
        якщо (s<='9')&(s>='0') то
            m←m+ord(s)-k0
        кр;
        взяти(s)
    кц;
    показати('Сума числових значень цифр');
    якщо k<>m то показати('не') кр;
    показати('дорівнює довжині тексту')
ка.
```

5.71. Знайти у даному тексті символ та довжину найдовшої послідовності однакових символів, що йдуть підряд.

5.5 Типи перерахування та обмеження

5.72. Скласти алгоритм виведення назв:

а) днів тижня;

б) місяців у році;

в) кольорів спектру;

г) шахових фігур.

<u>*Розв'язок* а)</u>

```
Алг Дні_Тижня це
    тип тиждень=(пн,вт,ср,чт,пт,сб,нд);
    змін день:тиждень;
поч
    для день←пн до нд повт
        показати (день)
    кц
ка.
```

5.73. За назвою країни визначити назву її столиці.

<u>*Вказівка*</u>. Розглянути типи:

<u>*тип*</u> *країна*=(Австрія,Болгарія,Італія,Україна,Франція,Швеція);

столиця=(Відень,Софія,Рим,Київ,Париж,Стокгольм).

5.74. За українською назвою мови програмування визначити її англійську назву.

5.75. За назвою країни визначити назву континенту, де вона розташована.

5.76. За назвою місяця визначити сезон (пору року), на який цей місяць приходиться.

<u>*Розв'язок.*</u>

```
Алг Пора_року це
    тип сезон=(зима,весна,літо,осінь);
    місяць=(січ,лют,бер,квіт,трав,чер,лип,сер,вер,жов,лис,груд);
```

```
        змін m:місяць;
                s:сезон;
поч
        взяти(m);
        вибрати m із
        |груд,січ,лют:s←зима;
        |бер,квіт,трав:s←весна;
        |чер,лип,сер:s←літо;
        |вер,жов,лис:s←осінь
        кв;
        показати(s)
ка.
```

5.77. Довжину відрізка, задану в міліметрах, сантиметрах, дециметрах чи кілометрах замінити на величину цієї довжини в метрах.

5.78. Надрукувати слово з заданого переліку: (*степ*, *біль*, *зошит*, *двері*) у заданому відмінку і однині.

5.79. Корабель йшов за деяким курсом, потім його курс було змінено згідно даному наказу. Визначити новий курс корабля.

Вказівка. Розглянути типи

```
тип     курс =(північ,схід,південь,захід);
        наказ=(вперед, вправо, вліво, назад);
```

5.80. За літерою-цифрою *d* від '0' до '5' визначити назву цієї цифри.

5.81. Для натурального числа *k* від 1 до 99 надрукувати фразу «Мені *k* років», враховуючи при цьому, що при деяких значеннях *k* слово «років» треба замінити на слово «рік» або «роки».

Розв'язок.

```
Алг Вік це
        змін k:1..99; l:0..9;
поч
        взяти(k);
        показати('Мені ',k,' ');
        якщо (k<=19)&(k>=11) то
                показати('років')
        інакше
                l←k mod 10;
                вибрати k із
                |0,5..9: показати('років');
                |1: показати('рік');
                |2..4:показати('роки')
                кв
        кр
ка.
```

5.82. Для натурального числа *k* надрукувати фразу «Ми знайшли *k* грибів у лісі», погодивши закінчення слова «гриб» з числом *k*.

5.83. Визначити кількість днів в заданому місяці.

5.84. Визначити дату наступного дня.

5.85. Визначити день тижня у 1994 році за його датою, знаючи, що 7 січня 1994 року — субота.

5.86. Визначити дату *k*-го за рахунком дня високосного року.

5.87. Визначити порядковий номер того дня високосного року, який має задану дату.

Розв'язок

```
Алг Номер_дня це
тип місяць=(січ, лют, бер, квіт, трав, чер, лип, сер, вер, жов, лис, груд);
змін m,m1:місяць;
```

```
        k:0..366; d:1..31;
поч
        взяти(d,m);
        m1←січ; k←d;
        поки m1<m повт
                вибрати m1 із
                |січ,бер,трав,лип,сер,жов: k←k+31;
                |лют:k←k+29;
                |квіт,чер,вер,лис:k←k+30
                кв;
                m1←succ(m1)
        кц;
        показати(k)
ка.
```

5.6 Тип даних рядок

5.88. Скласти алгоритм підрахунку загального числа входжень символів '+', '-', '*' у рядок *A*.

5.89. Скласти алгоритм перетворення рядка *A*, замінивши у ньому всі знаки оклику '!' крапками '.', кожну крапку – трьома крапками '...', кожну зірочку '*'- знаком '+'.

5.90. Інверсія рядка *A* — це рядок *B*, записаний тими ж символами у зворотньому порядку. Інверсія порожнього рядка за означенням – порожній рядок. Побудувати інверсію рядка.

5.91. Рядок називається симетричним, якщо його символи, рівновіддалені від початку та кінця рядка, співпадають. Порожній рядок вважається симетричним. Перевірити рядок *A* на симетричність.

5.92. Скласти алгоритм видалення із рядка *A* всіх входжень заданої групи символів.

5.93. Скласти алгоритм перетворення слова *A*, видаливши у ньому кожний символ '*' та подвоївши кожний символ, відмінний від '*'.

5.94. Скласти алгоритм підрахунку найбільшої кількості цифр, що йдуть підряд, у рядку *A*.

5.95. Скласти алгоритм підрахунку числа входжень у рядок *A* заданої послідовності літер.

5.96. Скласти алгоритм, який за рядком *A* та символом *S* будує новий рядок, отриманий заміною кожного символу, слідуючого за *S*, заданим символом *C*.

5.97. Скласти алгоритм перетворення рядка *A* видаленням із нього всіх ком, які передують першій крапці, та заміною у ньому знаком '+' усіх цифр '3', які зустрічаються після першої крапки.

5.98. Скласти алгоритм виведення на друк усіх цифр, які входять в заданий рядок, та окремо — решту символів, зберігаючи при цьому взаємне розташування символів у кожній з цих двох груп.

5.99. Рядок називається монотонним, якщо він складається з зростаючої або спадної послідовності символів. Скласти алгоритм перевірки монотонності рядка.

5.100. Скласти алгоритм обчислення числа входжень у рядок *A* символів, перелічених у рядку *V*.

Знайти символ, кількість входжень якого у рядок *A*

а) максимальна;

б) мінімальна.

5.101. Виділити з рядка *A* найбільший підрядок, перший і останній символи якого співпадають.

5.102. Перевірити, чи складається рядок з

а) 2 симетричних підрядків;

б) n симетричних підрядків.

5.103. Видалити з рядка всі повторні входження символів.

5.104. Виділити з рядка найбільший монотонний підрядок, коди послідовних символів якого відрізняються на 1.

5.105. Замінити всі пари однакових символів рядка, які йдуть підряд, одним символом. Наприклад, рядок *'aabcbb'* перетворюється у *'abcb'*.

5.106. Побудувати рядок *S* з рядків *S1*, *S2* так, щоб у *S* входили

а) ті символи *S1*, які не входять у S2;

а) всі символи *S1*, які не входять у *S2*, та всі символи *S2*, які не входять у *S1*.

5.107. Видалити з рядка симетричні початок та кінець. Наприклад, рядок *'abcdefba'* перетворюється у *'cdef'*.

5.108. Скласти алгоритм виведення на друк тільки маленьких літер українського алфавіту, які входять в заданий рядок.

Розвязок. Наступний символ рядка виводиться на друк, якщо він співпадає з одним з символів слова *V* — послідовності всіх маленьких літер українського алфавіту.

```
Алг Ukr_lit це
        змін c,d:симв;
                A,P,V:рядок;
                q:бул;
поч
        взяти(A);
        V← 'абвгдеєжзіийклмнопрстуфхцчшщьюя';
        поки len(A)>0 повт
                P←V; c←hd(A);
                повт
                        q←c=hd(P); P←tl(P);
                до q∨(len(P)=0);
                якщо q то показати(c) кр;
                A←tl(A)
        кц;
ка.
```

5.109. Скласти алгоритм виведення на друк рядка *A*, складенного з маленьких літер українського алфавіту, відповідними великими літерами.

5.110. Заданий рядок, який складається з великих літер українського алфавіту. Скласти алгоритм перевірки впорядкованості цих літер за алфавітом.

5.111. Скласти алгоритм виведення на друк в алфавітному порядку усіх різних маленьких українських літер, які входять до даного рядка.

5.112. Словами називаються підрядки, які розділені одним чи кількома пропусками та не містять пропусків всередині себе. За рядком *A* скласти алгоритм виведення на друк:

а) всіх слів рядка у зворотньому порядку;

б) усіх слів, які зустрічаються у рядку по одному разу;

в) цього ж рядка, але з видаленням з нього повторних входжень слів.

5.113. Знайти

а) найкоротше слово рядка;

б) найдовше слово рядка;

5.114. Використовуючи операції роботи з рядками, скласти алгоритми, які реалізують наступні дії:

а) знищення *n* символів рядка *S*, починаючи з позиції *k*;

б) вставка рядка *A* у рядок *B*, починаючи з позиції *k*;

в) виділення із рядка *S* підрядка R довжиною *n* символів, починаючи з позиції *k*;

г) перетворення дійсного числа *d* у рядок *S*;

д) перетворення рядка *S* у дійсне число *d* з кодом перетворення *i*: *i*=0 у випадку успішного перетворення або *i* дорівнює номеру першого помилкового символу рядка;

е) виділення з рядка *A* підрядка довжиною *n* символів, починаючи з кінця рядка.

Зауваження. Задачі 5.53 – 5.71 з підрозділу «Символьний тип даних» (5.4) можуть бути сформульовані у термінах рядків і запропоновані для розв'язання у даному підрозділі. Тоді в умовах задач замість «текст» або «послідовність символів» слід читати «рядок». Так, наприклад, алгоритм розв'язку задачі 5.59 може бути складено наступним чином:

```
Алг Del_scob це
        змін d:симв;
```

```
                A,S:рядок;
поч
        взяти(A); S←' ';
        поки len(A)>0 повт
                d←hd(A);
                якщо d='(' то
                        повт
                                A←tl(A)
                        до hd(A)=')'
                        кц
                інакше S←app(S,d)
                кр;
                A←tl(A)
        кц;
        показати(S)
ка.
```

6. Підпрограми

6.1. Скласти алгоритм обчислення добутку

$$p=f_0*f_1*\ldots*f_n,$$

де $f_l=\dfrac{1}{l^2+1}+\dfrac{1}{l^2+2}+\ldots+\dfrac{1}{l^2+l+1}$

Вказівка. Використати для обчислення f_l функцію *Sum_drob* у вигляді

```
        функція Sum_drob (l:нат):дійсн  це
                змін i:нат;
                        m:ціл;
                        s:дійсн;
        поч
                s←0; m← l*l;
                для i← 0 до l повт
                        s←s+1.0/(m+i+1)
                кц;
                Sum_drob←S
        кф;
```

6.2. Два простих числа називаються «близнюками», якщо вони відрізняються один від одного на 2 (наприклад, числа 41 та 43). Скласти алгоритм виведення на друк всіх пар «близнюків» з відрізку [*n*,2**n*], де *n* — задане ціле число, яке більше 2.

6.3. Дано натуральне число *n* та послідовність натуральних чисел $a_1, a_2, \ldots, a_n$. Показати всі елементи послідовності, які є

а) повними квадратами;

б) степенями п’ятірки;

в) простими числами.

Визначити відповідні функції для перевірки, чи є число: повним квадратом, степенню п’ятірки, простим числом.

6.4. Дано натуральне число *n*. Для чисел від 1 до *n* визначити всі такі, які можна представити у вигляді суми двох повних квадратів. Описати функцію, яка перевіряє, чи є число повним квадратом.

6.5. Дано парне число n>2. Перевірити для нього гіпотезу Гольдбаха, яка полягає в тому, що кожне парне число n>2 можна представити у вигляді суми двох простих чисел. Визначити функцію, яка перевіряє, чи є число простим.

6.6. Скласти алгоритм обчислення величини

$$\frac{\sqrt[3]{a} - \sqrt[6]{a^2 + 1}}{1 + \sqrt[7]{3 + a}}$$

для заданого дійсного числа a>0. Визначити функцію обчислення коренів $y = \sqrt[k]{x}$ з точністю ε за наступною ітераційною схемою

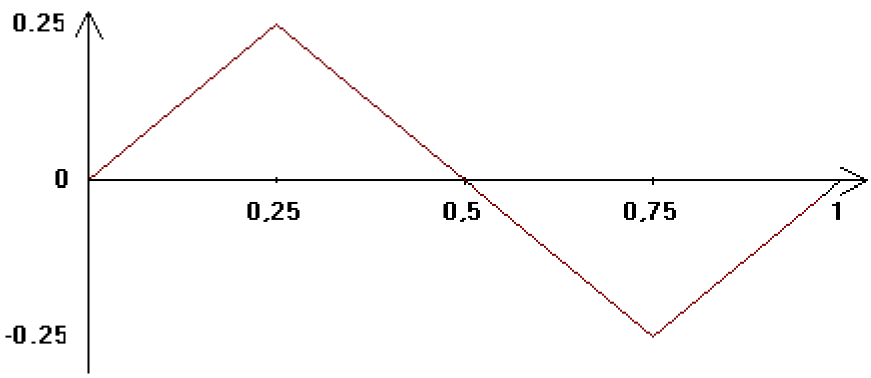
$$y_0 = 1;\ y_{n+1} = y_n + \left(x / \ y_n^{k-1} - y_n\right) / \ k \quad (n = 0,1,2,\ldots),$$

взявши за відповідь наближення y_{n+1}, для якого $\left|y_{n+1} - y_n\right| <$ ε.

6.7. Використовуючи функцію y=$arctg$(x), скласти підпрограму для обчислення функції, заданої співвідношенням

$$Arctg(x, y) = \begin{cases} arctg\dfrac{x}{y}, \text{ якщо } y > 0; \\ \dfrac{\pi}{2}, \qquad \text{якщо } x > 0 \text{ і } y > 0; \\ \pi + arctg\dfrac{x}{y}, \text{ якщо } x \geq 0 \text{ і } y < 0; \\ -\dfrac{\pi}{2}, \qquad \text{якщо } x < 0 \text{ і } y = 0; \\ -\pi + arctg\dfrac{x}{y}, \text{ якщо } x < 0 \text{ і } y < 0. \end{cases}$$

6.8. Скласти алгоритм обчислення значень функції *f(x)*, періодичної з періодом 1 і визначеної на всій числовій вісі. Графік функції зображено на малюнку 6.1



Мал. 6.1 — Графік періодичної функції до завдання 6.7.

Які допоміжні підпрограми будуть потрібні для розв'язку задачі ?

6.9. Визначити функцію для обчислення еліптичного інтегралу

$$I = \int_a^b \frac{dx}{a^2 \cdot \cos^2 x + b^2 \cdot \sin^2 x}, (a < b),$$

який, як показав Гаусс , рівний границі $I = \lim\limits_{n\to\infty} a_n = \lim\limits_{n\to\infty} b_n$ монотонно-збіжних послідовностей a_n і b_n, які визначаються рекурентними співвідношеннями

$$a_0 = a; b_0 = b; a_{n+1} = \sqrt{a_n \cdot b_n}, b_{n+1} = \frac{a_n + b_n}{2}, n = 1,2,\ldots$$

Вказана границя називається арифметико-геометричним середнім чисел a і b.

<u>*Вказівка.*</u>При виборі умови повторення циклу врахувати , що

$$a = a_0 < a_1 <\ldots< a_n <\ldots< b_n <\ldots< b_0 = b$$

6.10. Визначити функції для обчислення

а) синуса ; б) косинуса

використовуючи їх розклади в ряд Тейлора.

6.11. Дано координати вершин трикутника і точки всередині його. Використовуючи функцію для обчислення площі трикутника через три його сторони, визначити відстань від даної точки до найближчої сторони трикутника.

<u>*Вказівка.*</u>Врахувати що площа трикутника обчислюється також через основу і висоту.

6.12. Скласти функцію перевірки заданого рядка на симетричність.

Скласти функцію для побудови інверсії заданого рядка.

<u>*Розв`язок*</u>.

<u>*функція*</u> *Інв(S: рядок): рядок* <u>*це*</u>
    <u>*змін*</u> *A: рядок;*
<u>*поч*</u>
    A←'';
    <u>*поки*</u> *len(S)>*0 <u>*повт*</u>
        A ← add(A, hd(S));
        S ← tl(S)
    <u>*кц;*</u>
    Інв ← A
<u>*кф*</u>;

6.13. Перевірити, чи є даний рядок ідентифікатором, натуральним числом, чи ні тим ні іншим. Скласти функції, які визначають чи є заданий символ літерою та чи є даний символ цифрою.

6.14. Скласти функцію, яка визначає позицію першого (останнього) входження заданого символа в заданий рядок.

6.15. Скласти процедуру, яка замінює в початковому рядку символів всі одиниці на нулі, а всі нулі — на одиниці. Заміна повинна виконуватись, починаючи з заданої позиції рядка.

6.16. Скласти процедуру, в результаті звернення до якої з першого заданого рядка видаляється кожний символ, який належить і другому заданому рядку.

6.17. Скласти підпрограму для обчислення значення натурального числа за заданим рядком символів, який є записом цього числа у системі числення за основою b ($2<b<16$). Використати функцію, яка за заданим символом повертає відповідну цифру у системі числення за основою b.

6.18. Скласти підпрограму для отримання за заданим натуральним числом рядка символів, який є записом цього числа у системі числення за основою b ($2<b<16$). Використати функцію, яка за заданою цифрою у системі числення за основою b повертає символ, що відповідає цій цифрі.

6.19. Скласти алгоритм додавання «у стовпчик» двох чисел, записаних у вигляді рядків, що є позиційними записами цих чисел у десятковій системі числення. Використати підпрограми:

1) функцію *GetDigit(c)* отримання цифри за символом *c*;

2) функцію *GetSymbol(d)* отримання символа за цифрою *d*;

3) процедуру *AddDigit(n1, n2, p, n)* додавання двох цифр *n1*, *n2* з урахуванням перенесення р та отримманням останньої цифри результату *n*;

4) функцію додавання двох рядків у стовпчик *AddColumn*(*S1*, *S2*).

6.20. Скласти алгоритм множення «у стовпчик» двох чисел, записаних у вигляді рядків, що є позиційними записами цих чисел у десятковій системі числення. Використати підпрограми:

1) функцію *GetDigit(c)* отримання цифри за символом *c*;

2) функцію *GetSymbol(d)* отримання символа за цифрою *d*;

3) процедуру *MulDigit(n1, n2, p, n)* множення двох цифр *n1*, *n2* з урахуванням перенесення *p* та отримманням останньої цифри результату *n*;

4) підпрограму *MulStrChar*(*S*, *c*) множення рядка *S* на символ *c*;

5) підпрограму *AddString*(*S1*, *S2*, *n*) додавання двох рядків у стовпчик зі «зсувом» другого рядка на *n* позицій ліворуч.

6.21. Скласти процедуру «стискання» рядка: кожний підрядок, який складається з кількох входжень одного і того ж символа, замінюється самим цим символом.

6.22. Скласти процедури, які виділяють

1) перше слово рядка, залишаючи рядок без першого слова;

2) n-те слово рядка.

Використати одну з цих процедур для

а) підрахунку кількості слів рядка;

б) отримання найдовшого слова;

в) отримання найкоротшого слова;

г) отримання всіх слів, які є паліндромами (симетричними);

д) отримання всіх слів, які є ідентифікаторами;

д) отримання всіх слів, які є натуральними числами.

6.23. Скласти рекурсивні підпрограми для обчислень значень функцій

а)
$$f(x)=\begin{cases} f\left(\dfrac{x}{2}\right)+\sqrt{x}, \text{ якщо } x \ge 10^{-6}, \\ 0, \text{ якщо } x < 10^{-6}. \end{cases}$$

б)
$$f(x)=\begin{cases} f(x+1)+\dfrac{1}{x}, \text{ якщо } 10^{-6} \le x \le 10^{3}, \\ 0, \text{ якщо } x > 10^{3}, x < 10^{-6}. \end{cases}$$

в)
$$f(x)=\begin{cases} f(\ln x)+\ln x, \text{ якщо } x > 10^{-6}, \\ 0, \text{ якщо } x \le 10^{-6}. \end{cases}$$

г)
$$f(x)=\begin{cases} f(2*x)+f\left(\dfrac{x}{2}\right), \text{ якщо } \dfrac{1}{2} \le x \le 2^{10}, \\ x, \text{ якщо } x < \dfrac{1}{2}, x > 2^{10}. \end{cases}$$

6.24. Скласти рекурсивну функцію для обчислення многочленів Ерміта (див. завдання 3.15 б) з теми 3.2. «Програмування рекурентних співвідношень») і порівняти кількість дій у рекурсивному та нерекурсивному варіантах.

6.25. Визначити рекурсивну функцію обчислення *НСД*(*n,m*) натуральних чисел, яка грунтується на співвідношенні *НСД*(*n,m*)=*НСД*(*m,r*), де *r* — остача від ділення *n* на *m*.

6.26. Визначити рекурсивну процедуру представлення натурального числа *Z* у вісімковій системі числення.

Розв'язок. Переведення числа *Z* у вісімкову систему числення здійснюємо шляхом ділення його на 8 і виведення залишків від ділення у зворотній послідовності:

```
процедура Convert(Z:цiл) це
    поч
        якщо Z>1 то Convert(Z div 8);
        показати(Z mod 8)
    кп.
```

6.27. Визначити рекурсивну функцію обчислення степеня дійсного числа з цілим показником x^n згідно з формулою

$$x^n = \begin{cases} 1, \text{ якщо } n = 0, \\ \dfrac{1}{x^{|n|}}, \text{ якщо } n < 0, \\ x * x^{n-1}, \text{ якщо } n > 0. \end{cases}$$

6.28. Визначити рекурсивну функцію для обчислення біноміального коефіцієнту C_n^m, $0 \le m \le n$, за такою формулою:

$$C_n^0 = C_n^n = 1; C_n^m = C_{n-1}^m + C_{n-1}^{m-1}, \text{ при } 0<m<n.$$

6.29. Визначити рекурсивну функцію для знаходження суми додатніх дійсних чисел, які складають непорожню послідовність, за якою слідує від'ємне число.

6.30. Визначити рекурсивну функцію для обчислення числа Фібоначчі F_n для заданого натурального *n* (див. завдання 10 з теми «Арифметичний цикл»). Порівняти працемісткість рекурсивного і нерекурсивного варіантів.

6.31. Задані натуральні числа *a,c,m*. Визначити рекурсивну функцію для обчислення *f*(*m*) за формулою

$$f(m) = \begin{cases} m, \text{ якщо } 0 \le m \le 9, \\ g(m) * f(m - 1 - g(m)) + m \text{, у інших випадках} \end{cases},$$

g(*m*) — остача від ділення *a***n*+*c* на 10.

Розв'язок.

```
функція F(a,c,m:нат):нат це
    змін z,y:нат;
    поч
        якщо (m<=9) & (m>=0) то
            y ← m
        інакше
            z ← (a * n+c) mod 10; y ← z*F(a,c,m-1-z)+m
        кр;
```

$F \leftarrow y$

<u>*кф*</u>;

6.32. Визначити рекурсивні функції

а) перевірки заданого рядка на симетричність;

б) побудови рядка, інвертованого по відношеню до заданого;

в) заміни у вихідному рядку всіх входжень даного символа даним рядком;

г) перевірки, чи є один рядок початком іншого;

д) перевірки на входження одного рядка у інший.

<u>*Вказівка*</u>. Нехай Λ, A, $B \in W$ (Λ — порожній рядок), $x,y \in Ch$.

Для побудови рекурсивних функцій використати співвідношення

а) *сим*(Λ) = *Іст, сим*(*add*(*x*, Λ) = *Іст,*
сим(*app*(*add*(*x, A*)*, y*)) = (*x*=*y*) & *сим*(*A*);

б) *інв*(Λ) = Λ,
інв(*add*(*x, A*) = *app*(*інв*(*A*), *x*);

в) *зам*(Λ*, x, B*) = Λ,
зам(*add*(*y, A*)*, x, B*) = *add*(*y, зам*(*A, x, B*)),
зам(add(*x, A*)*, x, B*) = *B*+*зам*(*A, x, B*);

г) *поч*(Λ*, B*) = *Іст*, *поч*(*add*(*x*, *A*), Λ) = *Хиб*,
поч(*add*(*x*, *A*), *add*(*y*, *B*)) = (*x* = *y*) & *поч*(*A*, *B*);

д) *вход*(Λ, *B*) = *Іст*, *вход*(add(*x*, *A*), Λ) = *Хиб*,
вход(*add*(*x*, *A*), *add*(*y*, *B*)) = *поч*(*add*(*x*, *A*), *add*(*y*, *B*))$\vee$*вход*(add(*x*, *A*)*, B*).

<u>*Розв`язок*</u> в) Виходячи з наведених співвідношень, функцію заміни побудуємо таким чином

```
функція Zam(A: рядок; x: симв; B: рядок): рядок це
    поч
        якщо len(A)=0 то Zam ← A
        інякщо hd(A) = x то Zam ← B+Zam(tl(A), x, B)
        інакше Zam ← add(c, Zam(tl(A), x, B)) кр
    кф;
```

6.33. Скласти рекурсивну функцію для обчисленя функції Аккермана *Акк*(*n,m*), заданої співвідношенням

Акк(*0,m*)=*m+1;*
Акк(*n,0*)=*Акк*(*n-1,1*)*;*
Акк(*n,m*)=*Акк*(*n-1,Акк*(*n,m-1*)).

Обчислити *Акк*(0,5),*Акк*(1,2),*Акк*(2,2).

<u>*Розв`язок*</u>.Обчислимо функцію Аккермана за допомогою рекурсивної функції

```
функція Акк(N,M:нат):нат це
    змін Y:нат;
    поч
        якщо N=0 то
            Y← M+1
        інякщо M=0 то
            Y← Акк(N-1,1)
        інакше
            Y← Акк(N,M-1);Y← Акк(N-1,Y)
        кр;
        Акк← Y
    кф;
```

Покажемо спосіб обчислення функції Аккермана на прикладі:

Акк(1,2)=*Акк*(0,*Акк*(1,1))=*Акк*(0,*Акк*(0,*Акк*(1,0)))=
Акк(0,*Акк*(0,*Акк*(0,1)))=*Акк*(0,*Акк*(0,2))=*Акк*(0,3)=4.

6.34. Скласти алгоритм обчислення суми:

$$S_{ij} = \sum_{k_1=j-1}^{i-1} \sum_{k_2=j-2}^{i-2} \dots \sum_{k_{j-1}=1}^{k_{j-2}-1} \sum_{k_j=0}^{k_{j-1}-1} 1$$

6.35. Ханойські вежі. Дошка має три стрижні. На перший нанизано *N* дисків спадного догори діаметра. Потрібно, перекладаючи диски по одному, розмістити їх в початковому порядку на другому стрижні. При цьому більший диск ніколи не повинен розміщуватись над меншим.

Скласти підпрограму, яка ілюструє порядок переміщення дисків. Викликати її при *N*=3. Підрахувати кількість ходів, які потрібні для переміщення дисків. Знайти її залежність від *N*.

Розв`язок. Припусимо, що ми вміємо переносити *N*-1 диск. Тоді правило переміщення *N* дисків із стрижня *A* на стрижень *B* з використанням стрижня *C* для тимчасового зберігання дисків можна записати таким чином:

1)перенести *N*-1 диск з *A* на *C*;
2)перенести 1 диск з *A* на *B*;
3)перенести *N*-1 диск з *C* на *B*.

Очевидно алгоритм має зміст при *N*>1. Одержуємо рекурсивну процедуру

```
процедура Ханой(арг N,A,B,C:нат) це
    поч
        якщо N>0 то
            Ханой(N-1,A,C,B);
            показати(A,`→ `,B);
            Ханой(N-1,C,B,A)
        кр
    кп.
```

яку можна викликати так: *Ханой*(*N*,1,2,3).

При *N*=3 одержимо 1→ 3, 1 → 2, 3→ 2, 1→ 3, 2→ 1, 2→ 3, 1→ 3.

Кількість ходів, потрібних для переміщення 3 дисків, дорівнює 7, для *N* дисків — $2^N - 1$

6.36. Скласти алгоритм, який відображає всі перестановки цілих чисел від 1 до *N*.

Вказівка. Множину перестановок цілих чисел від 1 до *N* можна отримати з множини всіх перестановок цілих чисел від 1 до *N*-1, вставляючи *N* в усі можливі позиції в кожній перестановці.

7. Структури даних

7.1 Множини

7.1. Описати функцію *card*(*A*), яка підраховує кількість елементів у числовій множині

7.2. Визначити функію, що визначає мінімальний елемент чілсової множини.

7.3. Визначити процедуру виведення елементів числової множини в порядку зростання.

7.4. Визначити функцію доповнення множини.

7.5. Вагою числової множини називається сума модулів всіх його елементів. Вага порожньої множини вважається рівною нулеві. Скласти функцію для обчислення ваги числової множини.

7.6. Діаметром числової множини називається величина

$$d(A) = \max_{x,y \in A} |x - y|$$

Визначити функцію обчислення діаметра *d*(*A*).

7.7. Дано *n* цілих чисел від 1 до 50. Визначити, скільки серед них чисел Фібоначчі та скільки чисел, перша значуща цифра в десятковому запису яких 1 чи 2. Надрукувати всі ці числа (без повторів) у порядку зростання.

7.8. Визначити

а) функцію обчислення кількості різних (значущих) цифр у десятковому записі натурального числа *n*;

б) процедуру друку у порядку зростання всіх цифр, які не входять до десяткового запису натурального числа *n*.

7.9. Скласти функцію визначення тривалості місяця у році *Y*.

<u>*Вказівка*</u>. Викорискати в якості базового типу в означенні множини

<u>*тип*</u> *місяць* = (*січ, лют, берез, квіт, трав, черв, лип, серп, верес, жовт, лист, груд*);.

7.10. Дана непорожня послідовність символів (рядок) *S*. Описати функцію, яка визначає загальну кількість цифр та знаків операцій, що входять до рядка *S*.

7.11. Дано рядок з малих латинських літер. Надрукувати:

а) перші входження літер до рядка, зберігаючи початковий взаємний порядок;

б) всі літери,що входять до рядка не менше двох раз;

в) всі літери, що входять до рядка по одному разу.

7.12. Дано рядок символів *S*. В алфавітному порядку надрукувати (по одному разу) всі малі українські голосні літери, що входять до *S*.

7.13. Дана непорожня послідовність слів із малих українських літер; між сусідніми словами — пропуск, за останнім символом — крапка. Надрукувати в алфавітному порядку:

а) всі голосні літери, які входять до кожного слова;

б) всі приголосні літери, які не входять до жодного слова;

в) всі дзвінки приголосні літери, які не входять до жодного слова;

г) всі глухі приголосні, які не входять до жодного слова;

д) всі приголосні, які входять тільки в одне слово;

е) всі глухі приголосні літери, які не входять тільки в одне слово;

є) всі голосні літери, які входять не більше ніж в одне слово;

ж) всі дзвінки приголосні літери, які входять в кожне непарне слово і не входять в жодне парне слово;

з) всі глухі приголосні літери, які входять хоча б в одне непарне слово та не входять в жодне непарне слово.

<u>*Вказівка.*</u> До голосних літер належать а, е, и, і, о, у, є, ю, ї; до приголосних — всі інші крім й, ь; дзвінкі приголосні — б, в, г, д, ж, з, л, м, н, р; глухі приголосні: к, п, с, т, ф, х, ц, ч, ш, щ.

7.14. Задача пошуку елемента множини може бути розв’язана з використанням процедури пошуку, в якій використовується бульова функція *Q*(*x*), що задає умову успішності пошуку.

<u>*процедура*</u> *Пошук*(<u>*арг*</u> *A*:<u>*множ із*</u> *a..b*; <u>*рез*</u> *x:a..b*; *знайд:бул*) <u>*це*</u>
        <u>*змін*</u> *вичерп:бул;*
<u>*поч*</u>
        x← *a*; *вичерп*← *x*>*b*; *знайд*←*Хиб*;
        <u>*поки*</u> ¬(*вичерп* ∨ *знайд*) <u>*повт*</u>
                <u>*якщо*</u> *x* <u>*в*</u> *A* <u>*то*</u> *знайд* ← *Q*(*x*) <u>*кр*</u>;
                <u>*якщо*</u> ¬*знайд* <u>*то*</u>
                        вичерп ← *x*=*b*;
                        <u>*якщо*</u> ¬*вичерп* <u>*то*</u> *x*←*succ*(*x*) <u>*кр*</u>
                <u>*кр*</u>
        <u>*кц*</u>
        { <u>*якщо*</u> *знайд* <u>*то*</u> *Q*(*x*) <u>*інакше*</u> *«(x* <u>*в*</u> *A)* ¬*Q*(*x*) }
<u>*кп*</u>;

Спростити процедуру пошуку, якщо відомо,що *a* та *b* належать цілому або натуральному типу, причому *b* менше максимально представимого числа.

7.15. Запропонувати інший спосіб організації циклу по множині, не використовуючи інструкцію одержання елемента.

7.2 Записи та об'єднання

7.16. Нехай задано тип для представлення комплексних чисел у алгебраїчній формі:

```
тип Компл_Алг = запис
                    Re: дійсн;
                    Im: дійсн
                кз;
```

Визначити функції для обчислення

а) суми; б) різниці; в) добутку; г) частки двох комплексних чисел.

Розв'язок б).

```
функція Різниця(X,У: Компл_Алг): Компл_Алг це
        змін z:Компл_Алг;
поч
        Z.Re←X.Re-Y.Re;
        Z.Im←X.Im-Y.Im;
        Сума←Z
кф;
```

7.17. Визначити процедури

а) введення;   б)виведення комплексного числа.

Розв'язок б).

```
процедура Вивед_К(арг X:Компл_Алг) це
поч
        показати(X.Re);
        якщо X.Im>=0 то показати ('+') кр;
        показати (X.Im,'.*i')
кп;
```

7.18. Визначити програмовані функції для обчислення

а) аргументу;   б) модуля комплексного числа.

Розв'язок б).

```
функція Модуль_К(X:Компл_Алг):дійсн це
        змін A:дійсн;
поч
        A←sqrt(X.Re*X.Re+X.Im*X.Im);
        Модуль_К←A
кф;
```

7.19. Визначити програмовану функцію степеня комплексного числа з натуральним показником.

7.20. Визначити тип «комплексне число у тригонометричній формі» й підпрограми для операцій та інструкцій з завдань 7.16, 7.17, 7.19.

7.21. Визначити функції переведення комплексного числа з алгебраїчної форми у тригонометричну та навпаки.

7.22. Визначити процедуру обчислення коренів квадратного тричлена з заданими комплексними коефіцієнтами.

7.23. Визначити функцію обчислення значень квадратного тричлена з комплексними коефіцієнтами в заданій комплексній точці.

7.24. Визначити програмовані функції обчислення суми всіх доданків, модуль яких не менше $\varepsilon > 0$, у комплексній точці z.

а) $e^z = 1 + \frac{z}{1!} + \frac{z^2}{2!} + ... + \frac{z^n}{n!} + ...;$

б) $sh\, z = z + \frac{z^3}{3!} + \frac{z^5}{5!} + ... + \frac{z^{2n+1}}{(2n+1)!} + ...;$

в) $ch\, z = 1 + \frac{z^2}{2!} + \frac{z^4}{4!} + ... + \frac{z^{2n}}{(2n)!} + ...;$

г) $sin\, z = z - \frac{z^3}{3!} + \frac{z^5}{5!} - ... + (-1)^n \frac{z^{2n+1}}{(2n+1)!} + ...;$

д) $cos\, z = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - ... + (-1)^n \frac{z^{2n}}{(2n)!} + ...;$

е) $ln(1+n) = z - \frac{z^2}{2} + \frac{z^3}{3} - ... + (-1)^{n-1} \frac{z^n}{n} + ... \quad (|z| < 1);$

є) $arctg\, z = z - \frac{z^3}{3} + \frac{z^5}{5} - ... + (-1)^n \frac{z^{2n+1}}{2n+1} + ... \quad (|z| < 1)$

7.25. Визначити типи запису для представлення наступних понять:
а) ціна в гривнях і копійках;
б) час в годинах, хвилинах і секундах;
в) дата (число, місяць, рік);
г) адреса (місто, вулиця, будинок, квартира);
д) семінар (предмет, викладач, № групи, день тижня, години занять, аудиторія);
е) бланк вимоги на книгу (відомості про книгу: шифр, автор, назва; відомості про читача: № читацького білета, прізвище; дата заказа);
є) поле шахової дошки (*a*5, *b*8 і т.д.);
ж) коло (радіус, координати центру).

Розв'язок б), в).

Визначимо типи таким чином

б) *тип* *Час*=(*h*: 0..23; *mn*,*sec*: 0..59);

в) *тип* *Дата* = *запис*

D:1..31;
M: *місяць*;
Y : *ціл*

кз;

де тип *місяць* описано у завданні 7.9.

7.26. Визначимо тип «карта» наступним чином :

тип *масть* = (*піки, хрести, бубни, чирви*);
значення = (*шість,сім, вісім, дев`ять, десять, валет, дама, король, туз*);
карта=(*m*: *масть*; *d*: *значення*).

Описати бульову функцію, яка перевіряє, чи «б'є» карта *K1* карту *K2* враховуючи те, що масть *KM* є козирною .

7.27. Визначивши тип «*Поле*» (задача 7.25 є)), описати бульову функцію, яка перевіряє, чи може ферзь за один хід перейти з поля *h1* шахової дошки на поле *h2*.

7.28. Визначимо тип «раціональне число»:

тип *Рац* = (*ch*: *ціл*; *zn*: 1..*maxint*),

де *maxint*-максимально представиме ціле число.

Якщо змінна *X* має тип *Рац*, то в прийнятих позначеннях *X.ch*-чисельник, *X.zn*-знаменник дробу.

Визначити програмовані функції для
а) обчислення суми двох раціональних чисел ;
б) обчислення добутку двох раціональних чисел ;
в) порівняння двох раціональних чисел ;
г) зведення раціонального числа до нескортного виду .

7.29. Використовуючи опис типу «*Дата*» (завдання 7.25 в)), визначити функції обчислення:
а) дати вчорашнього дня ,
б) дня тижня по його даті в поточному році .
Вказівка 14 в). Визначимо тип шуканої функції наступним чином
тип *назва_дня* = (*пн, вт, ср, чт, пт, сб, нд*).

7.30. Визначимо універсальний комплексний тип, який допускає як алгебраїчне, так і тригонометричне представлення:

```
тип    Предст = (алг, трг);
       Компл = вибір C: Предст із
                   алг: (Re, Im: дійсн;);
                   трг: (R, Phi: дійсн;)
             кв;
```

Визначити програмовану функцію для обчислення модуля комплексного числа.
Розв'язок.

```
функція Модуль(Z: Компл): дійсн це
поч
    якщо Z.C=трг то Модуль ← Z.R
    інакше Модуль ← Sqrt(Z.Re*Z.Re+Z.Im*Z.Im) кр
кф.
```

7.31. Визначити основні функції для універсального комплексного типу (див. завдання 7.16 – 7.19). Для двомісних операцій передбачити усі можливі випадки.

7.32. Визначити тип *Плоска_Фігура*, який включає трикутник, прямокутник, трапецію та круг. Побудувати функції обчислення периметру та площі плоских фігур.

7.33. Визначити універсальний тип, який допускає представлення точки на площині у прямокутній або полярній системі координат. Побудувати функцію обчислення площі трикутника з вершинами *A*, *B*, *C*.

7.3 Масиви

7.34. Визначити функцію для обчислення суми:
а) компонент з парними номерами;
б) компонент з непарними номерами;
в) додатніх компонент;
г) від'ємних компонент
дійсного вектора.

7.35. Визначити процедури для
а) введення ;
б) виведення дійсного вектора.

7.36. Визначити функції обчислення
а) середнього арифметичного компонент дійсного вектора;
б) норми дійсного вектора;
в) відстані між двома точками в n-вимірному евклідовому просторі;
г) скалярного добутку двох дійсних векторів.

7.37. Визначити функції обчислення кількості компонент дійсного вектора
а) більших заданого числа;

б) які належать заданому відрізку прямої.

7.38. Визначити функції обчислення

а) похідної від многочлена $P_n(x) = a_0 * x^n + a_1 * x^{n-1} + \ldots + a_{n-1} * x + a_n$ в заданій точці x;

в) інтеграла многочлена $P_n(x)$ на заданому відрізку.

7.39. Використавши підпрограму обчислення значення многочлена $P_n(x)$ з завдання 7.38 а) в заданій точці x, скласти алгоритм обчислення значень двох многочленів $P(x)=1+x+x^2+...+x^{20}$, $Q(x)=1-x+x^2-...-x^{15}$.

7.40. Скласти підпрограми для обчислення функцій

$$а)\ f(x_1,...,x_n) = \begin{cases} 0, & \text{якщо } x_1 > x_2 > ... > x_n, \\ \sum_{i=1}^{n} |x_{i+1} - x_i| & \text{в інших випадках;} \end{cases}$$

$$б)\ f(x_1,...,x_n) = \begin{cases} 1, & \text{якщо } x_1 <= x_2 <= ... <= x_n, \\ \sum_{i=1}^{n-1} 2^{x_i + x_{i+1}} & \text{в інших випадках;} \end{cases}$$

$$в)\ f(x_1,...,x_n) = \begin{cases} 0, \text{якщо } x_1 <= 2^1 <= x_2 <= 2^2 <= ... <= x_n <= 2^n \\ \prod_{i=1}^{n} x_i, \text{ в інших випадках;} \end{cases}$$

$$г)\ f(x_1,...,x_n) = \begin{cases} 1, & \text{якщо } \max_i x_i > \sum_{i=1}^{n} x_i, \\ \sum_{x_i > 0} x_i, & \text{в інших випадках;} \end{cases}$$

$$д)\ f(x_1,...,x_n) = \begin{cases} 0, & \text{якщо } \min_i x_i < \prod_{i=1}^{n} x_i, \\ \sum_{i=1}^{n-1} x_i * x_{i+1}, & \text{в інших випадках;} \end{cases}$$

$$е)\ f(x_1,...,x_n,y_1,...,y_n) = \sum_{i=1}^{n} (x_i^2 + y_i^2 + x_i * y_i);$$

$$є)\ f(x_1,...,x_n,y_1,...,y_n) = (x_1^2 + ... + x_n^2) * (y_1^2 + ... + y_n^2)$$

$$ж)\ f(x_1,...,x_n,y_1,...,y_n) = \prod_{i=1}^{n} (x_i^3 + y_i^3);$$

з) $f(x_1,...,x_n,y_1,...,y_n) = (x_1 + \frac{1}{y_1}) * ... * (x_n + \frac{1}{y_n});$

і) $f(x_0, x_1,...,x_n,y_1,...,y_n) = x_0 + \sum_{j=1}^{n}(x_j + \prod_{i=1}^{j} y_i);$

и) $f(x_0,x_1,...,x_n,y_1,...,y_n) = x_0 + x_1^2 * y_1^2 + x_2^2 * y_1^2 * y_2^2 + ... + x_n^2 * y_1^2 * y_2^2 * ... * y_n^2$

7.41. Слід квадратної матриці — це сума її діагональних елементів. Визначити функцію обчислення сліду матриці .

7.42. Визначити функцію для обчислення

а) визначника квадратної матриці;

б) ранга квадратної матриці.

Розв'язок а). Розв`язувати задачу будемо зведенням матриці до трикутного вигляду. Якщо в i-му рядку a_{ii}=0, то будемо шукати такий рядок j, в якому a_{ii}<>0 Якщо такий рядок відсутній, то визначник дорівнює нулю, інакше міняємо місцями i-ий та j-ий рядки. Маємо

```
функція  Визначник(A: масив[1..n,1..n] із дійсн): дійсн це
        змін i,j,k,l:ціл ;c,p,b:дійсн;
поч
        p←1; i←0;
        поки (p<>0)&(i<n) повт
                i←i+1; j←n;
                якщо A[i,i]=0 то
                        поки (A[j,i]=0)&(j>i) повт
                                j←j-1
                        кц;
                        якщо j>i то
                                для l←i до n повт
                                        b←A[i,l]; A[i,l]←A[j,l]; A[j,l]←b
                                кц
                        кр
                кр;
                для k←i+1 до j повт             {цикл до j}
                        c←A[k,i]/A[i,i];
                        для l←i до n повт
                                A[k,l]←A[k,l]-A[i,l]*c
                        кц
                кц;
                p←p*A[i,i]
        кц;
        Визначник←p
кф;
```

Цикл, відмічений коментарем, виконується до j, а не до n, оскільки у рядках з номерами більшими, ніж j (якщо є такі), в i-му стовпчику стоїть нульовий елемент.

7.43. Визначити функції для обчислення

а) суми усіх недіагональних елементів матриці;

б) кількості нульових елементів матриці.

7.44. Визначити функції для обчислення:

а) суми двох векторів;

б) добутку вектора на число.

7.45. Розглядаючи вектори A та B як послідовності цифр десяткового запису деяких невід`ємних цілих чисел, отримати вектор C — аналогічне представленя для суми цих двох чисел.

7.46. Циклічним k-зсувом вектора $(a_1,a_2,...,a_n)$ вліво називається вектор $(a_{k+1},...,a_n,a_1,...,a_k)$. Визначити функцію для його обчислення.

7.47. Визначити циклічний k-зсув вправо та функцію для його обчислення.

7.48. Скласти підпрограми
а) обміну значень двох векторів;
б) перестановки компонент вектора в зворотньому порядку.

7.49. Визначити функцію обчислення вектора B за формулами $b_i = a_1 + a_2 + \ldots + a_i,\ i = 1,2,\ldots n$, де a_i-компоненти заданого вектора A.

7.50. Визначити функції для обчислення
а) суми; б) добутку
двох многочленів $A(x)=a_0+a_1*x+...+a_n*x^n$ і $B(x)=b_0+b_1*x+...+b_m*x^m$.

7.51. Задано многочлен $P(x)$ з нульовим вільним членом. Знайти перші n коефіцієнтів розкладу $1/P(x)$ в ряд Тейлора в нулі.

7.52. Скласти підпрограму обчислення частки $Q(x)$ та залишку $R(x)$ від ділення многочлена

$$P(x)=p_0+p_1*x+...+p_n*x^n$$

на многочлен

$$T(x)=t_0+t_1*x+...+t_m*x^m$$

$P(x)=T(x)*Q(x)+R(x)$, $deg(R)<deg(T)$, де deg — степінь многочлена.

7.53. Визначити функцію для обчислення найбільшого спільного дільника двох многочленів.

7.54. Задана дійсна матриця розміру $m\times n$. Знайти вектор B, k-та компонента якого b_k це:
а) сума абсолютних величин елементів k-го рядка матриці;
б) добутку елементів k-го рядка;
в) значення середнього арифметичного елементів k-го рядка;
г) число від`ємних елементів в k-му рядку;
д) добутку квадратів тих елементів k-го рядка, модулі яких належать відрізку [1;1,5] (якщо таких елементів немає, то покласти b_k=1);
е) значення першого по порядку додатнього елемента k-го рядка (якщо таких елементів немає, то покласти b_k =10);
є) сума елементів, які знаходяться за першим від`ємним елементом в k-му рядку (якщо таких елементів немає, то покласти b_k=100).

7.55. Скласти алгоритм «швидкого» обчислення числа Фібоначі (див. задвдання 3.10 з розд. 3) за його номером n, досліджуючи матричне відображення

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix},\ n\geq 2.$$

Вказівка. Довести співвідношення

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} F_1 \\ F_0 \end{pmatrix},$$

яке дозволяє звести задачу обчислення чисел Фібоначі до задачі обчислення степені заданої матриці 2-го порядку. Для швидкого обчислення степені використати метод, який було розглянуто у завданні 4.9 (розд. 4), замінюючи операцію множення чисел векторно-матричними операціями.

7.56. Визначити підпрограми:

а) транспонування матриці;

б) множення матриці на вектор;

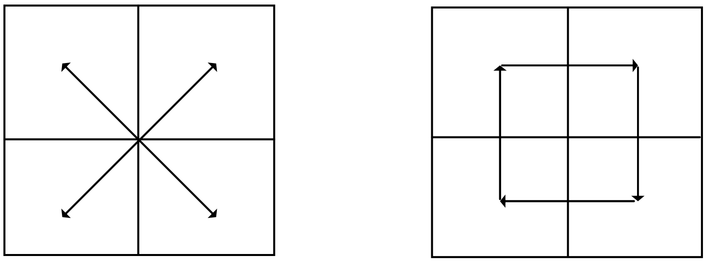
в) перестановки двох заданих рядків (стовпчиків) матриці;

г) перестановки заданого рядка матриці з заданим її стовпчиком;

д) побудови цілочисельної квадратної матриці порядку 7, елементами якої є числа 1, 2, ..., 49, розташовані в ній по спіралі;

е) видалення із матриці заданого рядка і заданого стовпчика.

7.57. Задана дійсна квадратна матриця порядку *2n.* Побудувати нову матрицю, переставляючи її блоки розміру $n \times n$



а) б)

Мал. 7.1

а) відповідно мал. 7.1 а);

б) відповідно мал. 7.1 б).

7.58. Задані натуральне число *m*, цілі числа $a_1, a_2, \ldots, a_m$ і цілочисельна квадратна матриця порядку *m*. Рядок з номером *i* матриці назвемо відміченим, якщо $a_i>0$, і невідміченим у протилежному випадку.

а) всі елементи, розташовані у відмічених рядках матриці, перетворити за правилом: додатні замінити на -1, від'ємні — на 1, а нульові залишити без зміни;

б) підрахувати кількість від'ємних елементів матриці, розташованих у відмічених рядках.

7.59. Задано дійсні матриці *A, B, C, N* порядку *n,* причому *N* — нульова матриця. Побудувати за цими матрицями матрицю *D* розміру $2n \times 3n$ у вигляді

$$D=\begin{pmatrix} A & B & C \\ B & N & A \end{pmatrix}.$$

7.60. Побудувати функцію, яка виконує теоретико-множинні операції над підмножинами скінченної універсальної множини, що складається з *n* елементів (*n* — задана константа).

Вказівка. Підмножини скінченної універсальної множини представити у вигляді бульового вектора *X* довжини *n,* поклавши *X(i)=Iст* тоді і тільки тоді, коли, $i \in X$.

7.61. Визначити процедуру пошуку заданого елемента матриці.

7.62. Визначити процедуру пошуку

а) максимальної ; б) мінімальної

компоненти вектора.

7.63. Визначити процедуру пошуку в заданому векторі компоненти:

а) більшої за задане число;

б) такої, що належить заданому відрізку на прямій.

7.64. Визначити процедуру підрахунку кількості максимальних і мінімальних компонент вектора.

7.65. Визначити процедуру одночасного обчислення максимальної і мінімальної серед компонент з парними і непарними номерами.

7.66. Визначити функцію для обчислення суми компонент дійсного вектора, які розташовані між максимальною та мінімальною компонентами (всі компоненти вектора різні).

7.67. Задані координати *n* точок на площині $(x_1,y_1),...,(x_n,y_n)$. Знайти номери двох точок, відстань між якими найбільша (вважати, що така пара точок єдина).

7.68. Дано два дійсних вектори довжини *n*. Визначити процедуру пошуку найменшої серед тих компонент першого вектора, які входять до другого вектора.

7.69. Визначити процедуру пошуку спільної компоненти двох векторів.

7.70. Визначити процедуру пошуку в заданому векторі

а) двох компонент, що дорівнюють заданому числу;

б) двох нульових компонент, які йдуть підряд;

в) двох рівних компонент;

г) двох рівних компонент, які йдуть підряд;

д) компонент, що є числами Фібоначі.

7.71. Визначити процедуру обчислення кількості інверсій в заданому векторі (тобто таких пар компонент, в яких більше число знаходиться зліва від меншого: $x_i>x_j$ при $i<j$).

7.72. Визначити функцію, яка перевіряє впорядкованість вектора за зростанням. (Вектор $(a_1,a_2,...,a_n)$ впорядкований за зростанням, якщо $a_1<a_2<...<a_n$.).

7.73. Визначити процедуру пошуку спільної компоненти двох впорядкованих векторів.

7.74. Визначити процедуру перетворення дійсного вектора за наступним правилом: всі від'ємні компоненти вектора перенести в його початок, а всі інші — в кінець, зберігаючи початкове взаємне розташування як серед від'ємних, так і серед інших компонент.

7.75. Визначити два впорядкованих за неспаданням вектори. Визначити процедуру об'єднання

а) всіх компонент;

б) спільних компонент

цих двох векторів у третій, щоб він знову став впорядкованим за неспаданням.

7.76. Задана цілочисельна квадратна матриця порядку *n* і цілочисельний вектор довжини *m*. Замінити нулями в матриці ті елементи, для яких існують рівні серед компонент даного вектора.

7.77. Заданий вектор, компонентами якого є множини всіх продуктів із вказаного асортименту, що знаходяться у відповідному магазині. Визначити процедуру пошуку множини продуктів

а) що є в кожному магазині;

б) кожен з яких є хоча б в одному магазині;

в) яких немає в жодному магазині.

7.78. Заданий вектор ГРУПА, компонентами якого ГРУПА(x) є множини імен людей із вказаного списку тих, що були в гостях у людини з ім'ям X ($X \notin$ГРУПА[X]).

Визначити процедуру пошуку хоча б одної людини, яка була в гостях в усіх інших людей, імена яких містяться в компонентах вектора ГРУПА.

7.79. Заданий вектор розмірності *n*, компонентами якого є записи, що містять відоммості про вершини гір. У відомостях про кожну вершину вказуються назва гори і її висота. Визначити процедуру пошуку найвищої вершини.

7.80. Заданий вектор ГР, компонентами якого ГР(x) є записи, що містять дані про людину з іменем X із вказаного списку. Кожне дане складається з вказаної статі людини та її зросту. Визначити підпрограми для

а) обчислення середнього зросту жінок з групи ГР;

б) пошуку найвищого чоловіка;

в) перевірки, чи є в групі ГР дві людини, однакові на зріст.

7.81. Заданий вектор ГР розмірності n, компонентами якого є записи, що містять анкети службовців деякого закладу. В кожній анкеті вказується прізвище службовця, його стать, дата народження у вигляді: число, місяць, рік. Визначити процедуру пошуку

а) найстаршого з чоловіків групи ГР;

б) людей з групи ГР, прізвища яких починаються з заданої літери.

7.82. Заданий вектор розмірності n, компоненти якого містять інформацію про студентів деякого вузу. Відомості про кожного студента складаються з указання його прізвища, ім'я, по батькові, статі, віку, курсу. Визначити процедуру пошуку

а) найбільш поширених чоловічих та жіночих імен;

б) прізвищ та ініціалів усіх студентів, вік яких є найбільш поширеним.

7.83. Заданий вектор розмірності n, компонентами якого є відомості про складання іспитів студентами деякого вузу. Інформація про кожного студента задана в слідуючому вигляді: прізвище, номер групи, оцінка_1, оцінка_2, оцінка_3. Визначити процедуру пошуку

а) студентів, що мають заборгованості хоча б з одного з предметів;

б) предмета, який було здано краще за усі інші;

в) студентів, що склали всі іспити на 5 і 4.

7.84. Задано вектор C розміру n, компонентами якого є відомості про мешканців деяких міст. Інформація про кожного мешканця містить його прізвище, назву міста мешкання; адреси мешкання у вигляді: вулиця, будинок, квартира. Визначити процедуру пошуку двох будь-яких жителів із списку C, що мешкають в різних містах за однаковою адресою.

7.85. Задано вектор Рейс, компонентами якого Рейс(x) є множини міст із вказаного списку, в які можна за один рейс доїхати автобусом з міста x. Визначити процедуру пошуку множини міст, в які можна дістатися автобусом (за один рейс або через інші міста) із заданого міста.

7.86. Визначити функцію обчислення норм дійсної матриці порядку n

а) $\|A\|_1 = \max\limits_{1\le i\le n} \sum\limits_{j=1}^{n} \left|a_{ij}\right|$, б) $\|A\|_2 = \max\limits_{1\le j\le n} \sum\limits_{i=1}^{n} \left|a_{ij}\right|$

7.87. Визначити процедуру обчислення у матриці

а) найменшого елемента;

б) найбільшого елемента;

в) суми елементів рядка, в якому розташований елемент з найменшим значенням;

г) суми найбільших значень елементів її рядків;

д) суми елементів рядків з від'ємним елементом головної діагоналі.

7.88. Визначити процедуру перестановки місцями рядка матриці, що містить елемент з найменшим значенням із рядком, в якому міститься елемент з найменшим значенням.

7.89. Визначити підпрограму пошуку найменшого серед найбільших елементів рядків квадратної дійсної матриці порядку n, тобто

$$\min_{1\le i\le n} \max_{1\le j\le n} \left\{a_{ij}\right\}.$$

<u>Розв'язок</u>. Оформимо підпрограму у вигляді функції, аргумент якої — дійсна матриця A

```
функція Мін_Макс(A:масив [1..n,1..n] із дійсн): дійсн це
        змін i,j:1..n; min,max:дійсн;
поч
        для i ← 1 до n повт
                max← A[i,1];
                для j← 2 до n повт
                        якщо A[i,j]>max то
                                max←A[i,j]
                        кр
                кц;
                якщо i=1 то min← max кр;
                якщо min>max то
```

min←max

<u>*кр*</u>

<u>*кц*</u>;

Мін_Макс← min

<u>*кф*</u>;

7.90. Визначити процедуру пошуку в матриці

а) індексів усіх її ненульових елементів;

б) кількості усіх її різних елементів.

7.91. Дана дійсна матриця розміру $m \times n$. Визначити підпрограму побудови вектора, компонентами якого є

а) найменше значення елементів рядків;

б) різниці між найбільшим і найменшим значеннями елементів рядків;

в) найбільші із значень елементів рядків від початку до головної діагоналі включно.

7.92. Елемент матриці називається «особливим», якщо:

1) він більший за суму інших елементів свого стовпчика;

2) в його рядку зліва від нього знаходяться елементи, менші за нього, а справа — більші.

Визначити процедуру пошуку числа «особливих» елементів матриці.

7.93. Елемент матриці називається слабким локальним максимумом, якщо з нього не можна зсунутись на крок в жодному з чотирьох напрямів в бік більшого елемента. Елемент називається сильним локальним максимумом, якщо з нього не можна зсунутись по жодному з восьми напрямків.

Визначити процедуру обчислення кількості сильних і слабких максимумів матриці.

7.94. Визначити функцію перевірки матриці на симетричність.

<u>*Розв'язок*</u>. Умова симетричності квадратної матриці має вигляд: $a_{ij}=a_{ji}$ для всіх $i<j$, $i=1, 2, ..., n-1$; $j=i+1, i+2, ..., n$. Тому задачу можна розглядати як задачу пошуку таких i, j, що $a_{ij} \neq a_{ji}$. Якщо таких індексів немає, то матриця симетрична. Одержуємо

<u>*функція*</u> *Сим_матр (A*: <u>*масив*</u> [1..*n*,1..*n*] <u>*із*</u> *дійсн*): *бул* <u>*це*</u>

<u>*змін*</u> *i,j*: *ціл*; *Q*: *бул*;

<u>*поч*</u>

Q← i; *i←* 0;

<u>*поки*</u> *Q & (i<n-1)* <u>*повт*</u>

i← i+1; *j← i*;

<u>*поки*</u> *Q & (j<n)* <u>*повт*</u>

j← j+1;

Q←A[i,j]=A[j,i]

<u>*кц*</u>

<u>*кц*</u>;

Сим_матр← Q

<u>*кф*</u>;

7.95. Визначити процедуру побудови за заданою матрицею *A* цілочисельного вектора *b*, присвоївши його *k*-ій компоненті значення 1, якщо виконується вказана нижче умова, і значення 0 в протележному випадку:

а) всі елементи *k*-го стовпчика однакові;

б) елементи *k*-го рядка матриці впорядковані за спаданням;

в) *k*-ий рядок матриці симетричний;

г) елемент *k*-го рядка, що належить головній діагоналі, симетричний;

д) елементи *k*-го рядка не перебільшують заданого числа *x*;

е) в *k*-му рядку матриці є хоча б один від'ємний елемент.

7.96. Визначити процедуру перестановки рядків і стовпчиків матриці так, щоб найбільший елемент її опинився у верхньому лівому куті.

Вказівка. Знаходимо номер рядка k і номер стовпчика l, що містять найбільший елемент матриці. Потім міняємо місцями 1-ий і k-ий рядки, 1-ий і l-ий стовпчикі матриці.

7.97. Елемент матриці назвемо сідловою точкою, якщо він є найменшим в своєму рядку і водночас найбільшим в своєму стовпчику. Для заданої цілочисельної матриці визначити процедуру пошуку індексів усіх її сідлових точок.

7.98. Дана дійсна квадратна матриця, всі елементи якої різні. Визначити процедуру обчислення скалярного добутку рядка, в якому знаходиться найбільший елемент матриці, на стовпчик з найменшим елементом.

7.99. Визначити функцію, яка перевіряє, чи є задана цілочисельна квадратна матриця ортонормованою, тобто. такою, в якій скалярний добуток кожної пари різних рядків дорівнює 0, а скалярний добуток кожного рядка на себе дорівнює 1.

7.100. Визначити функцію, що перевірює чи є задана цілочисельна квадратна матриця магічним квадратом, тобто такою, в якій суми елементів в усіх рядках і стовпчиках однакові.

7.101. Дана бульова матриця. Відомо, що множина істинних елементів є об'єднанням прямокутників, що не перетинаються. Скласти процедуру пошуку їхньої кількості.

7.102. В умовах попереднього завдання скласти процедуру підрахунку кількості прямокутників, що є квадратами.

7.103. Задане деяке розташування чорних і білих пішаків (в довільній кількості) на шаховій дошці. Скласти процедуру підрахунку

а) кількості можливих ходів білими;

б) кількості взять білими;

в) кількості чорних і білих пішаків під боєм.

7.104. Задані n точок площини. Провести коло, на якому лежить найбільша кількість точок даної множини.

7.105. Задані n точок площини. Визначити процедуру підрахунку кількості рівнобічних трикутників з вершинами у ціх точках.

7.106. Задані n точок площини. Визначити процедуру вибору трьох з них таких, щоб периметр трикутника з вершинами в цих точках був найбільшим.

7.107. Таблиця футбольного чемпіонату задана квадратною матрицею порядку n, в якій всі елементи, що належать головній діагоналі, дорівнюють нулеві, а кожний елемент, що не належить головній діагоналі, дорівнює 3, 1 або 0 (числу очок, набраних у грі: 3 — виграш, 1-нічия, 0-програш). Скласти процедуру

а) знаходження кількості команд, що мають перемог більше, ніж поразок;

б) визначення номерів, що пройшли чемпіонат без поразок;

в) пошуку команд, що виграли більше половини ігор.

7.108. Задана бульова матриця СУСІДИ, елементи якої СУСІДИ(a,b) рівні істині, якщо країни a і b мають спільний кордон, і хибності — в протилежному випадку. Скласти процедуру пошуку країни, що має найбільшу кількість сусідів серед перелічених країн.

7.109. Задана дійсна матриця T, елементи якої $T(x,y)$ означають середньомісячну температуру на острові x в місяці y. Скласти процедуру, що визначає який місяць і на якому острові серед переліченіх островів є найхолоднішим.

7.110. Задана непротирічна таблиця спорідненості ТС, елементи якої ТС(x,y) можуть приймати значення із списку (*син*, *дочка*, *батько*, *мати*, *ні*). Наприклад, ТС(x,y)=*ні*, якщо y не є ні одним з батьків, ні родичем, ні дитиною x'_a, ТС(x,y)=син, якщо y — син x'_a і т.п. Скласти процедуру пошуку

а) будь-якої з онучок;

б) будь-якого з дядьків;

в) кількості двоюрідних братів і сестер

людини з ім'ям n із заданого списку.

7.111. Знайти символ, який входить у рядок S найбільшу кількcть разів.

Вказівка: визначити масив з індексами символьного типу та елементами натурального типу, в якому обчислити та зберегти кількість входжень кожного символу у рядок S; знайти максимальний елемент цього масиву.

7.4 Файли

7.112. Скласти програму для обчислення значень многочлена, його похідної, використовуючи файл його коефіцієнтів.

7.113. Дано файл, компоненти якого є дійсними числами. Скласти підпрограму для обчислення:

а) суми компонент файлу;
б) кількості від'ємних компонент файлу;
г) останьої компоненти;
д) найбільшого із значень компонент файлу;
е) найменшого із значень компонент файлу з парними номерами;
є) суми найбільшого і найменьшого із значень компонент;
ж) різниці першої і останьої компоненти файлу;
з) кількості компонент файлу, які менші за середнє аріфметичне усіх його компонент.

7.114. Дано файл, компененти якого є цілими числами. Скласти підпрограму для обчислення:
а) кількості парних чисел серед компонент;
б) кількості квадратів непарних чисел серед компонент;
в) різниці між найбільшим парним і найменьшим непарним числами з компонент;
г) кількості компонент в найдовшій зростаючій послідовності компонент файлу.

7.115. За масивом A, елементи якого утворені за законом
а) $a_i=i$; б) $a_i=i^2$; в) $a_i=i!$; г) $a_i=2^{i+1}$; д) $a_i=2^i+3^{i+1}$ (i=1,2,...,n)
побудувати файл,який містить елементи масиву A.

7.116. Послідовність $x_1, x_2, ...$ утворена за законом

$$x_i = \frac{i-0.1}{i^3+|tg\,2i|}$$

(i=1,2,...). Дано дійсне $\varepsilon>0$. Побудувати файл, який містив би усі члени послідовності $x_1, x_2, ...$, які стоять перед першим членом, для якого виконано: $|x_i| < \varepsilon$.

7.117. Скласти процедуру запису в файл всіх чисел Фіббоначі, які не перевищують натуральне число n.

7.118. Дано файл f, компоненти якого є цілими числами. Побудувати файл g, який містив би всі компоненти файлу f:
а) які є парними числами;
б) які діляться на 3 і на 5;
в) які є точними квадратами;
г) записані у зворотньому порядку;
д) за виключенням повторних входжень одного і того самого числа.

7.119. Використовуючи файл f, компоненти якого є цілими числами, побудувати файл g, який містить всі парні числа файлу f і файл h — всі непарні. Послідовність чисел зберігається.

7.120. Задані натуральне число i файл f, компоненти якого є цілими числами. Побудувати файл g, записавши в нього найбільше значення перших n компонент файлу f, потім-наступних n компонент і т.д. Розглянути два випадки:
а) число компонент файлу ділиться на n;
б) число компонент файлу не ділиться на n.
В цьому випадку останнія компонента файлу g повинна дорівнювати найбільшій із компонент файлу f, які утворюють останню (неповну) групу.

7.121. Дано файл f, компоненти якого є цілими числами. Файл f містить рівне число додатніх та від'ємних чисел. Використовуючи допоміжний файл h, переписати компоненти файлу f в файл g так, щоб у файлі g :
а) не було двох сусідніх чисел одного знаку;

б) спочатку йшли додатні, потім від'ємні числа;

в) числа йшли таким чином: два додатніх, два від'ємних і т.д. (припускаються, що число компонент в файлі *f* ділиться на 4).

7.122. Назвемо файл, компоненти якого є символами, символьним файлом. По рядку *S* побудувати файл, який містить символи цього рядка.

7.123. Дано символьний файл *f*. Побудувати файл *g*, утворений із файлу f:

а) зміною всіх його великих літер однойменними малими;

б) записом його компонент у зворотньому порядку.

7.124. Дано символьний файл, який складається не менш, ніж із 2 компонент. Визначити, чи є два перших символа файлу цифрами. Якщо так, то виявити, чи є число утворене цими цифрами парним.

7.125. Задано символьні файли *f1* і *f2*. Переписати із збереженням послідовності компоненти файлу *f1* у *f2*, а компоненти файлу *f2* — у *f1*. Використати допоміжний файл *h*.

7.126. Задано символьні файли *f* і *g*.Записати в файл *h* спочатку компоненти файлу *f,* потім — компоненти файлу *g* зі збереженням порядку.

7.127. Дано символьний файл. Скласти підпрограму для:

а) додавання в його кінець заданого символа;

б) додавання в його початок заданого символа;

в) підрахунку числа входжень у файл заданого символа;

г) визначення входження у файл заданої комбінації символів;

д) вилучення заданого символа;

е) вилучення других входжень кожного символа.

7.128. Файл *F* типу <u>*файл із*</u> (*ключ*:*t; s*:*q*) називається впорядкованим за ключами, якщо записи в ньому розташовуються в порядку зростання ключів. Скласти процедуру пошуку запису за ключем у впорядкованому файлі.

7.129. Скласти функцію перевірки рівності файлів, виконаної за один перегляд.їх змісту.

7.130. Скласти процедуру вилучення з файлу запису із заданим ключем.

7.131. Скласти процедуру вилучення з впорядкованого файлу запису із заданим ключем.

7.132. Скласти функцію побудови множини записів файлу.

<u>*Розв'язок.*</u> Визначимо типи

```
тип τ = запис
            Ключ: t;
            s: q
            кз;
      st = множ із τ;

Функція SetZap(F :файл із τ): st це
      змін s: st; x: τ;
поч
      s ← [ ]; відкрити(F);
      поки ¬eof(F) повт
            читати(F,x); s←s+[x]
      кц;
      SetZap←s
кф;
```

7.133. Файл *H* називається злиттям файлів *F* і *G,* якщо *SetZap*(*H*)=*SetZap*(*F*)∪*SetZap*(*G*) (див. попереднє завдання). Скласти процедуру злиття двох впорядкованих файлів у новий впорядкований файл.

7.134. Багаж пасажира характеризується номером пасажира, кількістю речей та їх загальною вагою. Дано файл пасажирів, який містить прізвища пасажирів, та файл, який містить інформацію про багаж декількох пасажирів (номер пасажира – це номер запису у файлі пасажирів). Скласти процедуру для

а) знаходження пасажира, у багажу якого середня вага однєї речі відрізняється не більш як на 1 кг від загальної середньої ваги речей;

б) визначення пасажирів, які мають більше двох речей, і пасажирів, кількість речей яких перебільшує середнє число речей;

в) видачі відомостей про пасажира, число речей у багажі якого не менш, ніж у будь-якому іншому багажі, а вага речей не більша, ніж у будь-якому іншому багажі з цим же числом речей;

г) визначення, чи мають хоча б два пасажири багажі, які не відрізняються кількістю речей та відрізняються вагою не більше, ніж на 1 кг (якщо такі пасажири є, то показати їх прізвища);

д) визначення пасажира, багаж якого складається з однієї речі, вагою не менше 30 кг.

7.135. Відомості про учня складаються з його імені, прізвища та назви класу (рік навчання та літери), в якому він вчиться. Дано файл, який містить відомості про учнів школи. Скласти підпрограми, які дозволяють

а) визначити, чи є в школі учні з однаковим прізвищем;

б) визначити, чи є учні з однаковим прізвищем в якихось паралельних класах;

в) визначити, чи є учні з однаковим прізвищем у якомусь класі;

г) відповісти на питання а)-в) по відношенню до учнів, у яких співпадають і ім'я, і прізвище;

д) визначити, в яких класах нараховується більше. ніж 35 учнів;

е) визначити, на скільки учнів в восьмих класах більше, ніж в десятих;

є) зібрати у файл відомості про учнів 9-х і 10-х класів, помістивши спочатку відомості про учнів класу 9а, потім 9б і т.д.

ж) отримати список учнів даного класу за наступними зразками:

прізвище ім'я
прізвище і.
і.прізвище.

7.136. Дано файл, який містить, ті ж самі відомості про учнів школи, що і в попередній задачі, та додатково оцінки, отримані учнями на екзаменах за заданими предметами. Скласти процедури для

а) визначення кількості учнів, які не мають оцінок нижче чотирьох;

б) побудови файлу, який містить відомості про кращих учнів школи, тобто про учнів, які мають оцінки не нижче чотирьох;

в) друку відомостей про учнів, які мають хоча б одну незадовільну оцінку, у вигляді: прізвище та ініціали, назва класу, предмет.

7.137. Відомості про автомобіль складаються з його марки, номера і прізвища власника. Дано файл, який містить відомості про декілька автомобілей. Скласти процедури знаходження

а) прізвищ власниківі номерів автомобілей даної марки;

б) кількості автомобілей кожної марки.

7.138. Дано файл, який містить різні дати. Кожна дата — це число, місяць та рік. Скласти процедури пошука

а) року з найменьшим номером;

б) всіх весняних дат;

в) найпізнішої дати.

7.139. Дано файл,який містить відомості про книги. Відомості про кожну книгу — це прізвище автора, назва та рік видання. Скласти процедури пошуку

а) назв книг даного автора, виданих з 1960 р.;

б) книг із заданою назвою. Якщо така книга є, то надрукувати прізвища авторів та рік видання.

7.140. Дано файл, який містить номери телефонів співробітників установи: вказуються прізвище співробітника, його ініціали та номер телефона. Визначити процедуру пошуку телефона співробітника за його прізвищем та ініціалами.

7.141. Дано файл, який містить відомості про кубики: розмір кожного кубіка (довжина ребра в см), його колір (червоний, жовтий, зелений, синій) та матеріал (дерев'яний, металевий, картоний). Скласти процедури пошука

а) кількості кубіків кожного з перелічених кольорів та їх сумарний об'єм;

б) кількості дерев'яних кубіків із ребром 3 см та кількості металевих кубіків з ребром більшим за 5 см.

7.142. Дано файл, який містить відомості про речовину: вказується назва речовини, її питома вага та провідність (провідник, напівпровідник, ізолятор). Скласти процедури для

а) знаходження питомої ваги та назви всіх провідників;

б) вибору даних про напівпровідники та впорядкавування їх, шоб кожний наступний був з меншою питомою вагою, ніж попередній.

7.143. Дано файл, який містить відомості про товари, що експортуються. Вказано назву товару, країну, яка імпортує товар, та об'єм прибулої частини у штуках. Скласти процедуру пошуку країн, в які експортується даний товар та загальний об'єм його експорту.

7.144. Задано два файли *f1* і *f2*. Фвйл *f1* — це інвентарний файл, який містить відомості про те, скільки виробів і яких видів продукції зберігається на складі (вид продукції задається його порядковим номером). Файл *f2* — це допоміжний файл, який містить відомості про те, наскільки зменшилась чи збільшилась кількст виробів по деяких видах продукції. Допоміжний файл може містити декілька відомостей про продукції одного виду або не містити жодної такої відомості. Скласти процедуру оновлення інвентарного файлу за допоміжним, склавши новий файл *g*.

7.145. Дано файл, який містить відомості про іграшки: вказується назва іграшки (наприклад: м'яч, лялька, конструктор і т.д.), її вартість в гривнях та вікові границі дітей, для яких іграшка призначається (наприклад, для дітей від двох до п'яти років). Скласти процедури

а) пошуку назв іграшок, вартість яких не перевищує 40 грн. та які підходять дітем 5-и років;

б) пошуку назв іграшок, які підходять дітям і 4, і 10 років;

в) пошуку назв найдорожчих іграшок (ціна яких відрізняється від ціни найдорожчої іграшки не більше, ніж на 50 грн.);

г) визначення ціни найдорожчого конструктора;

д) визначення ціни всіх кубіків;

е) пошуку двох іграшок, які підходять дитині 3-х років, сумарна вартість яких не перевищує 20 грн.

є) пошуку конструктора ціною 22 грн., призначеного дітям від 5 до 10 років. Якщо такої іграшки немає, то занести відомості про неї у файл.

7.146. Дано файл, який містить відомості про прямокутники: вказується номер прямокутника у файлі, координати верхнього лівого кута прямокутника, координати нижнього правого кута прямокутника. Скласти процедуру пошуку прямокутника з найбільшою площею і визначення цієї площі.

Розв'язок. Компоненти файлу подамо у вигляді записів, які містять п'ять полів: перше поле — номер прямокутника у файлі, друге і третє — координати верхнього лівого кута, четверте і п'яте — координати нижнього правого кута прямокутника.

<u>*тип*</u> *Прям* = <u>*запис*</u>
    k: *цiл;*
    x1, y1, x2, y2: *дійсн*
<u>*кз*</u>;

Тоді процедура пошуку прямокутника з найбільшою площею може бути записана у вигляді:

<u>*процедура*</u> *Прямокутник*(<u>*арг*</u> *F*: <u>*файл із*</u> *Прям*; <u>*рез*</u> *i*: *цiл*; *s*: *дійсн*) <u>*це*</u>

```
        змін z Прям; s1:дійсн;
поч
        відкрити(F); s←0;
        поки ¬eof(F) повт
                читати(F,z);
                s1←abs((z.y2-z.y1)*(z.x2-z.x1));
                якщо s1>s то s←s1; i←z.k кр
        кц
кп;
```

7.147. У двох файлах міститься таблиця футбольного турніру. У першому файлі записано назви команд типу *Команда*. У другому файлі – результати матчів типу *Матч*.

```
тип     Команда = рядок;
        Матч = запис
                        n1, n2, m1, m2: нат
                кз;
```

У записі типу *Матч n1, n2* – номери першої та другої команд (номери записів у файлі команд); *m1, m2* – кількість м'ячів, забитих відповідно першою та другою командою.

Показати команду, яка є лідером, якщо за перемогу нараховується 3 очки, за нічию – 1, за поразку – 0.

З двох команд, які мають однакову кількість очок, першою вважається та, яка

1) має кращу різницю забитих та пропущених м'ячів;

2) при однаковій різниці має більше забитих м'ячів;

3) при всіх однакових попередніх показниках визначається жеребкуванням (використати для жеребкування генератор випадкових чисел).

Вказівка. Описати підпрограми створення файлів команд та матчів; додавання результату матчу; визначення лідера.

7.148. Дано текстовий файл. Групи символів, які відокремлені пропусками (одним або декількома) і не містять пропусків усередині себе, будемо називати словами. Скласти піпрограми для

а) знаходження найдовшого слова у файлі;

б) визначення кількості слів у файлі;

в) вилучення з файлу усіх слів, які складаються з однієї букви, та зайвих пропусків;

г) вилучення всіх пропусків на початку рядків, в кінці рядків та між словами (окрім одного);

д) вставки пропусків у рядки рівномірно між словами так, щоб довжина всіх рядків (якщо в них більше 1 слова) була 80 символів та кількість пропусків між словами у одному рядку відрізнялась не більше, ніж на 1 (вважати, що рядки файлу мають не більше, ніж 80 символів).

Результат записати у файл *h*.

7.149. Визначити процедуру, яка утворює текстовий файл із 9 рядків, в першому з яких одна літера «1», в другому — дві літери «2», ... , в дев'ятому – дев'ять літер «9».

7.150. Визначити процедуру, яка за заданою послідовністю символів формує текстовий файл із рядками по 40 літер (в останньому рядку літер може бути і менше).

7.151. Визначити функцію:

а) яка підраховує кількість порожніх рядків;

б) яка обчислює максимальну довжину рядків текстового файлу.

7.152. Визначити процедуру виведення

а) всіх рядків текстового файлу;

б) рядків, які містять більше 60 символів.

7.153. Визначити функцію, що обчислює кількість рядків текстового файлу, які

а) починаються із заданого символу;

б) закінчується на заданий символ;

в) починаються і закінчуються одним і тим же символом;

г) складаються з однакових символів.

7.154. Визначити процедуру, яка переписує у текстовий файл *g* всі рядки текстового файлу *f*
а) із заміною в них символа ‘0’ на ‘1’ і навпаки;
б) у інвертованому вигляді.

7.155. Визначити процедуру пошуку найдовшого рядка у текстовому файлі. Якщо таких рядків декілька, знайти перший з них.

7.156. Визначити процедуру, яка переписує компоненти текстового файлу *f* у файл *g,* вставляючи до початку кожного рядка один символ пропуску. Порядок компонент не повинен змінюватися.

7.157. Визначити процедуру, яка друкує по рядках зміст текстового файлу, вставляючи до початку кожного рядка його порядковий номер і символ пропуску.

7.158. Визначити процедуру пошуку у текстовому файлі рядків, фрагментом яких є заданий рядок.

7.159. Вважаючи, що довжина рядка текстового файлу *f* не перевищує 80, визначити процедуру, яка, доповнюючи короткі рядки файлу *f* пропусками справа, формує текстовий файл *g*, всі рядки якого мають довжину 80.

7.160. Визначити процедуру, яка, виключаючи пропуски, що стоять на початку рядків текстового файлу *f*, формує текстовий файл *g*.

7.161. У текстовому файлі записана непорожня послідовність дійсних чисел, які розділяються пропусками. Визначити функцію обчислення найбільшого з цих чисел.

7.162. У текстовому файлі *f* записана послідовність цілих чисел, які розділяються пропусками. Визначити процедуру запису у текстовий файл *g* всіх додатніх чисел із *f*.

7.163. У текстовому файлі кожний рядок містить декілька натуральних чисел, які розділяються пропусками. Числа визначають вигляд геометричної фігури (номер) та її розміри. Прийняті наступні домовленості:

- відрізок прямої задається координатами своїх кінців та має номер 1;
- прямокутник задається координатами верхнього лівого і нижнього правого кута та має номер 2;
- коло задається координатами центра і радіусом та має номер 3.

Визначити процедури обчислення
а) відрізка з найбільшою довжиною;
б) прямокутника з найбільшим периметром;
в) кола з найменшою площею.

7.164. Скласти алгоритм для перевірки правильності розстановки фігурних дужок у текстовому файлі – програмі на Сі.

7.165. Скласти алгоритм для перевірки правильності розстановки операторних дужок ‘begin’ та ‘end’ у текстовому файлі – програмі на Паскалі. Врахувати, що опис типу ‘record’ та оператор ‘case’ мають власний ‘end’.

Вказівка. Визначити підпрограми виділення з рядка підрядка, обмеженого з обох боків символами з заданої множини, та переведення рядка у верхній регістр.

7.5 Процедурний тип даних

7.166. Визначити функцію для обчислення кореня рівняння $f(x)=0$ на відрізку $[a,b]$, на якому $f(x)$ змінює знак, з заданою точністю ε методом ділення відрізка навпіл. Виконати обчислення кореня для функції $f(x)=x^3-7*x-1$.

7.167. Визначити функцію для обчислення кореня рівняння $f(x)=0$ на відрізку $[a, b]$, на якому $f(x)$ змінює знак, з заданою точністю ε методом хорд. Виконати обчислення кореня для функції $f(x)=x^3-7*x-1$.

Вказівка. Обчислюючи корінь методом хорд, з’єднують прямою точки $(a, f(a))$ та $(b, f(b))$ та знаходять точку x перетину цієї прямої з віссю абсцис. Якщо знаки $f(a)$ та $f(x)$ співпадають, далі пошук проводять на відрізку $[x, b]$, інакше – на відрізку $[a, x]$.

7.168. Скласти підпрограми для обчислення визначеного інтегралу
а) методом прямокутників; б) методом Сімпсона.

Вказівка б).Для обчислення інтегралу використати границю

$$I=\int_a^b f(x)dx=\lim_{n\to\infty} S_n,\ S_k=\frac{h_k}{3}\times(f_0+4\times f_1+2\times f_2+4\times f_3+\ldots+4\times f_{n-3}+2\times f_{n-2}+4\times f_{n-1}+f_n),$$

де $f_i=f(a+i*h_k)$, $h_k=\frac{b-a}{n}$, $n=2^k$, $i=0,1,\ldots,n$ і представлення S_k у вигляді

$$S_k=S_k^{(1)}+S_k^{(2)}+S_k^{(4)},$$

де

$$S_k^{(1)}=\frac{h_k}{3}*(f_0+f_n),\ S_k^{(2)}=\frac{h_k}{3}*(2*f_2+2*f_4+\ldots+2*f_{n-2}),\ S_k^{(4)}=\frac{h_k}{3}*(4*f_1+4*f_3+\ldots+4*f_{n-1}).$$

Для обчислення $S_k^{(1)}, S_k^{(2)}, S_k^{(4)}$ використати рекурентне співвідношення

$$S_k^{(1)}=\frac{1}{2}*S_{k-1}^{(1)},\ S_1^{(1)}=\frac{h}{3}*\left[f(a)+f(b)\right], h=\frac{b-a}{2}$$

$$S_k^{(2)}=\frac{1}{2}*S_{k-1}^{(2)}+\frac{1}{4}*S_{k-1}^{(4)},\ S_1^{(2)}=0,\ k=2,3,\ldots,$$

$$S_k^{(4)}=\frac{4*h_k}{3}*\left[f(a+h_k)+f(a+3*h_k)+\ldots+f(a+(n-1)*h_k)\right],\ S_1^{(4)}=\frac{4h}{3}*f\left(\frac{a+b}{2}\right).$$

7.169. Нехай *M1,M2,...,*M_n — матеріальні точки, положення яких на площині задано координатами $(x_1,y_1),\ldots,(x_n,y_n)$, а маси визначаються за допомогою вагової функції *g(x,y)*. Положення центру ваги цих точок задано формулами:

$$x=\frac{\sum_{i=1}^{n} g(x_i,y_i)*x_i}{\sum_{i=1}^{n} g(x_i,y_i)};\ y=\frac{\sum_{i=1}^{n} g(x_i,y_i)*y_i}{\sum_{i=1}^{n} g(x_i,y_i)}.$$

Визначити процедуру обчислення точки центру ваги (*x, y*) при заданій ваговій функції *g*(*x,y*).

Складену процедуру використати для знаходження положення центра ваги *n* точок при $g(x,y)=\sqrt{x^2+y^2}$.

7.170. Скласти підпрограму для знаходження елемента дійсного вектора, який задовольняє умову, задану булівською функцією *Q*(*x*). Виконати пошук, коли умовою *Q*(*x*) є:

а) *x*=*a*; б) *x*>*a*; в) *a*<=*x*<=*b*;

де *a*, *b* – задані числа.

7.171. Дано файл з дійсних чисел. Скласти підпрограму для знаходження компонента файлу, який задовольняє умову, задану булівською функцією *Q*(*x*). Виконати пошук, коли умовою *Q*(*x*) є:

а) *x*=*a*; б) *x*>*a*; в) *a*<=*x*<=*b*;

де *a*, *b* – задані числа.

8. Пошук та сортування

8.1. Дано вектор a з n цілих компонент. Знайти зростаючу підпослідовність компонентів вектора найбільшої довжини.

Вказівка. Використати вектор b, у який записувати останні елементи зростаючих підпослідовностей компонент з a довжини 1, 2, ..., k. Якщо на деякому кроці циклу $a[i] > b[k]$, то записати у (k+1)-й елемент вектора b компонент $a[i]$ та збільшити значення k на 1. Інакше знайти перший елемент вектора b, $b[j]$ такий, що $a[i] <= b[j]$ та записати $a[i]$ на місце $b[j]$. Оскільки елементи вектора b впорядковані, використати бінарний пошук для знаходження індекса j. Після того, як закінчимо прохід вектора a останній елемент вектора b буде останнім елементом шуканої підпослідовності, а кількість елементів вектора b, k — її довжиною. Для того, щоб показати саму підпослідовність (в оберненому порядку), можна використати цикл:

```
i←n;
поки k>0 повт
    поки a[i]<b[k] повт
        i←i-1
    кц;
    показати(a[i]);
    k←k-1
кц
```

8.2. Визначити процедуру впорядкування рядків дійсної матриці

а) за неспаданням їх перших елементів;

б) за незростанням сум їх елементів;

в) за неспаданням їх найменших елементів;

г) за незростанням їх найбільших елементів.

Використати методи обмінного сортування та сортування злиттям.

8.3. Дано масив з n точок площини, заданих своїми координатами. Впорядкивати точки за неспаданням відстані від початку координат.

8.4. Дано текстові файли F та G. Відомо, що кількість слів у файлі F не більше n, де n — відома константа. Перевірити, чи всі слова з файлу G входять у файл F.

Вказівка. Використати масив рядків a з n елементів, у який записати всі різні слова файлу F, впорядковані за зростанням. Використати бінарний пошук для пошуку слова з файлу G у масиві a.

8.5. В умовах попереднього завдання визначити кількість входжень кожного слова файлу F у файл G.

Вказівка. У масиві a з попереднього завдання зберігати разом зі словом кількість його входжень у файл G.

8.6. Даний текстовий файл F. Відомо, що кількість слів у файлі F не більше n, де n — відома константа. Побудувати за файлом F пару файлів G та H, щоб у компонентах файлу G зберігались усі різні слова файлу F, впорядковані за алфавітом, а у файлі H — цілі числа. Причому i-е число файлу H — це номер i-го слова з файлу F у файлі G.

Вказівка. Використати масив рядків a з n елементів, у який записати всі різні слова файлу F, впорядковані за зростанням.

8.7. Нехай файли A та B, компонентами яких є цілі числа, впорядковані з неспаданням. Отримати у файлі C всі числа з файлів A та B без повторень. Файл C повинен бути впорядкований за зростанням.

8.8. Даний файл F, компонентами якого є цілі числа. Отримати у файлі G всі непарні числя, що входять у F. Числа у файлі G повинні йти:

а) у порядку незростання;

б) у порядку спадання без повторень.

8.9. Даний текстовий файл F. Записати у файл G всі різні слова файлу F у порядку зростання.

9. Модулі

9.1. Описати модуль для реалізації універального комплексного типу (див. завдання 7.30). Реалізувати операції, відношення та інструкції:

1) взяти комплексне число;
2) показати комплексне число;
3) чи рівні два комплексних числа;
4) сума двох комплексних чисел;
5) різниця двох компдексних чисел;
6) добуток двох комплексних чисел;
7) частка від ділення двох комплексних чисел;
8) модуль комплексного числа;
9) піднесення комплексного числа до натурального степеня;
10) добуток комплексного числа та дійсного числа;
11) переведення комплексного числа до алгебраїчної форми;
12) переведення комплексного числа до тригонометричної форми.

З використанням модуля знайти корені рівняння $a*z^2+b*z+c=0$ з комплексними коефіцієнтами a, b, c.

9.2. Описати модуль роботи з відрізками на числовій вісі. Тип відрізку представити у вигляді запису:

```
тип Відрізок = запис
                        a, b: дійсн;
                        empty: бул
                  кз;
```

де a, b — границі відрізку, *empty* — ознака того, що відрізок порожній.

Реалізувати дії над відрізками:

1) зробити відрізок t порожнім;
2) чи порожній відрізок t;
3) покласти відрізок t рівним a, b;
4) покласти відрізок t рівним перетину відрізків $t1$, $t2$.

З використанням модуля скласти програму розв'язку системи квадратних нерівностей вигляду $x^2+p_ix+q_i>0$. Пари коефіцієнтів нерівностей p_i, q_i вводяться з пристрою введення.

9.3. Описати модуль для реалізації мультимножини цілих чисел на базі вектора. Мультимножина — це множина в якій для кожного елемента запам'ятовується не лише його входження, але й кількість входжень. Мультимножину описати як

тип Мультимножина = *масив* [0..n] *із нат*;

де n — відома константа. Кількість входжень елемента k ($0<=k<=n$) у мультимножину — це значення елемента масиву з індексом k.

Реалізувати дії над мультимножинами:

1) зробити мультимножину порожньою;
2) чи є мультимножина порожньою;
3) додати елемент до мультимножини;
4) забрати елемент з мультимножини (кількість входжень елемента зменшується на 1, якщо елемент не входить — відмова);
5) кількість входжень елемента у мультимножину;
6) об'єднання двох мультимножин (в результаті об'єднання кідькість входжень елемента визначається як максимальна з двох мультимножин);
7) перетин двох мультимножин (в результаті кількість входжень елемента визначається як мінімальна з двох мультимножин);

З використанням модуля розв'язати задачі:

а) знайти символ, який входить у рядок *S* максимальну кількість разів (див. завдання 7.111);

б) перевірити, чи складаються рядки *S1*, *S2* з одних і тих же символів, які входять у ці рядки однакову кількість разів;

в) перевірити, чи вірно, що всі символи рядка *S1*, входять також у рядок *S2*, причому не меншу кількість разів, ніж у *S1*.

9.4. Описати модуль для реалізації обмеженого стеку символів на базі вектора. Стек — це впорядкована сукупність однотипних елементів, в якій є доступ до одного елемента, що називається верхівкою стеку. Стек символів на базі масиву можна описати як:

<u>*тип*</u> *Стек* = <u>*масив*</u> [1..*n*] <u>*із*</u> *симв*;

де *n* — відома константа. Елементи стеку будуть займати початковий відрізок вектора до деякого індекса *top*, який вказує на верхівку стеку.

Реалізувати дії над стеком:

1) зробити стек порожнім;
2) чи порожній стек;
3) вштовхнути елемент у стек (додати новий елемент, який стає верхівкою стеку);
4) верхівка стеку (повернути значення верхівки стеку; для порожнього стеку — відмова);
5) забрати верхівку стеку (забрати верхній елемент стеку; для порожнього стеку — відмова).

З використанням модуля виконати інвертування вхідної послідовності символів.

9.5. Описати модуль роботи з точками та відрізками на площині. Типи точки та відрізку представити у вигляді записів:

```
тип    Точка = запис
                        x, y: дійсн;
               кз;
       Відрізок = запис
                        a, b: Точка;
               кз;
```

Реалізувати дії над точками:

1) взяти точку *t*;
2) покласти точку *t* рівною (*x*, *y*);
3) показати точку *t*.

Реалізувати дії над відрізками:

1) взяти відрізок *s*;
2) показати відрізок *s*;
3) покласти відрізок *s* рівним *a*, *b*;
4) довжина відрізку *s*;
5) чи лежить точка *t* на одній прямій з відрізком *s*;
6) чи лежить точка *t* всередині відрізку *s*;
7) площа трикутника, утвореного точкою *t* та відрізком *s*.

У файлі записано послідовність точок. З використанням модуля роботи з точками та відрізками на площині знайти:

а) трикутник з найбільшою площею, утворений точками послідовності;

б) коло найменшого радіуса, всередині якого лежать всі точки послідовності;

в) відрізок, на якому лежить найбільша кількість точок послідовності;

г) коло, на якому лежить найбільша кількість точок послідовності.

9.6. Описати модуль та скласти програму для реалізації гри у «хрестики-нолики» на полі розміром 3x3. У модулі реалізувати дії:

1) зробити хід гравця;
2) зробити хід комп'ютера;
3) показати ігрове поле.

9.7. Відома гра у «морський бій» полягає у наступному. Два гравці на двох полях 10х10 розставляють «кораблі» – прямокутники (4 – розміром 1х1, 3 – розміром 2х1, 2 – розміром 3х1, 1 – розміром 4х1). Кораблі не можуть мати сусідніх клітин по горизонталі, вертикалі або діагоналі. Гравці не бачать розстановку кораблів супротивника. Потім гравці по черзі роблять ходи (кожний хід – це вказання клітини на полі). Якщо гравець потрапляє на поле, яке займає корабель супротивника, він має право на позачерговий хід. Виграє той, хто першим знешкодить всі кораблі супротивника.

Описати модуль та програму для реалізації гри у «морський бій» між гравцем та комп'ютером. Передбачити у модулі реалізацію дій:

1) додати корабель;
2) зробити хід гравця;
3) зробити хід комп'ютера;
4) показати ігрове поле (поле гравця та стан поля комп'ютера).

10. Рекурсивні структури даних

<u>Зауваження</u>. Всі рекурсивні структури даних у цьому розділі реалізовувати на базі вказівників та динамічної пам'яті.

10.1. Описати модуль для реалізації стеку символів. Передбачити виконання дій над стеком:

1) Почати роботу.
2) Чи порожній стек?
3) Вштовхнути елемент у стек.
4) Верхівка стеку.
5) Забрати верхівку стеку.

Використовуючи цей модуль, розв'язати задачу: на вхід поступає послідовність символів. Впорядкувати цю послідовність за зростанням.

<u>Вказівка</u>. Для впорядкування використати 2 стеки.

10.2. Використовуючи модуль для реалізації стеку символів (див. завдання 10.1), скласти підпрограми:

а) довжина стеку;
б) змінити верхівку стеку;
в) інверсія стеку;
г) забрати n елементів стеку.

10.3. Описати модуль для реалізації черги цілих чисел. Передбачити виконання дій над чергою:

1) Почати роботу.
2) Чи порожня черга?
3) Додати елемент до кінця черги.
4) Взяти елемент з початку черги.

Використовуючи цей модуль, скласти підпрограму обчислення довжини черги. Запобігти знищенню черги після виклику підпрограми. Передбачити рекурсивний та нерекурсивний варіант.

10.4. Використовуючи модуль для реалізації черги цілих чисел (див. завдання 10.3), розв'язати наступну задачу. У черзі є m чисел. Проводять n випробувань, в результаті кожного з яких отримують випадкові числа 0 або 1. Якщо отримано 0, то треба взяти елемент з початку черги та показати його на екрані. Якщо отримано 1, то ввести число з клавіатури та додати до кінця черги. Після завершення випробувань показати залишок черги.

<u>Вказівка</u>. Використати генератор випадкових чисел (підпрограми *randomize* – ініціалізації генератора — та *random*(*k*) – отримання випадкового натурального числа у діапазоні від 0 до (*k*-1)).

10.5. Використовуючи модуль для реалізації черги цілих чисел (див. завдання 10.3), розв'язати задачу «лічилка», яка формулюється наступним чином. По колу розташовано n гравців, що мають номери від 1 до n. У лічилці m слів. Починають лічити з першого гравця. n-ий гравець вибуває, після чого знову починають лічити з гравця, що є наступним за тим, що вибув, і т. д. Треба показати послідовність номерів, що вибувають, при заданих значеннях n та m.

10.6. Використовуючи модуль для реалізації черги цілих чисел (див. завдання 10.3), розв'язати наступну задачу. У магазині стоїть черга з *m* покупців. Час обслуговування покупця з черги – це випадкове ціле число в діапазоні від 1 до t_1. Час додавання нового покупця до черги — це випадкове ціле число в діапазоні від 1 до t_2. Промоделювати стан черги (тобто показати час виникнення подій – обслуговування та додавання покупця) за період часу T ($T >> t_1$, $T >> t_2$). Показати залишок черги.

10.7. Використовуючи модуль для реалізації черги цілих чисел (див. завдання 10.3), скласти підпрограми:

а) інверсія черги;

б) забрати *n* елементів черги;

в) конкатенація двох черг;

г) порівняти 2 черги.

10.8. Описати модуль для реалізації деку цілих чисел. Передбачити виконання дій над деком:

1) Почати роботу.
2) Чи порожній дек?
3) Додати елемент до початку деку.
4) Взяти елемент з початку деку.
5) Додати елемент до кінця деку.
6) Взяти елемент з кінця деку.

Використовуючи цей модуль, скласти підпрограми: обчислення довжини деку, присвоєння для деків. Запобігти знищенню деку після виклику підпрограми обчислення його довжини.

10.9. Використовуючи модуль для реалізації деку цілих чисел (див. завдання 10.8), реалізувати стек цілих чисел на базі деку. Реалізувати стек на базі деку — означає описати модуль та реалізувати дії над стеком викликами відповідних підпрограм, що реалізують дії над деком. Для реалізованого стеку розв'язати задачу інвертування вхідної послідовності цілих чисел.

10.10. Використовуючи модуль для реалізації деку цілих чисел (див. завдання 10.8), реалізувати чергу на базі деку (див. попереднє завдання). Для реалізованої черги розв'язати задачу обчислення довжини черги.

10.11. Розв'язати задачу «лічилка» (див. завдання 10.5), використовуючи модуль для реалізації деку цілих чисел (див. завдання 10.8).

10.12. Використовуючи модуль для реалізації деку цілих чисел (див. завдання 10.8), розв'язати задачу 10.4, передбачивши однак чотири можливих результати кожного випробування (випадкові числа від 0 до 3):

- 0 – взяти елемент з початку деку та показати його на екрані;
- 1 – ввести число з клавіатури та додати його до початку деку;
- 2 – взяти елемент з кінця деку та показати його на екрані;
- 3 – ввести число з клавіатури та додати його до кінця деку.

10.13. Використовуючи модуль для реалізації деку цілих чисел (див. завдання 10.8), розв'язати задачу 10.6, передбачивши що через випадковий час від 1 до t_3 до початку черги додається «пільговий» покупець, який обслуговується першим, а через випадковий час від 1 до t_4 не витримує та йде з черги останній покупець.

10.14. Використовуючи модуль для реалізації деку цілих чисел (див. завдання 10.8), скласти підпрограми:

а) інверсія деку;

б) конкатенація двох деків;

в) порівняти 2 деки;

г) забрати *n* елементів з початку деку;

д) забрати *n* елементів з кінця деку;

е) замінити початок деку;

є) замінити кінець деку.

10.15. Описати модуль для реалізації класичного списку цілих чисел. Передбачити виконання дій над списком:

1) Почати роботу.
2) Чи порожній список?
3) Додати елемент.
4) Голова списку.
5) Хвіст списку.

Використовуючи цей модуль, скласти підпрограми:

а) обчислення довжини списку *Len*(*L*) та пошуку у списку елемента, рівного заданому числу *Search*(*L*, *x*);

б) конкатенації двох списків *Concat*(*L1*, *L2*) та інверсії списку *Invert*(*L*);

в) сортування списку *Sort*(*L*);

г) перевірки, чи є один список початком другого *IsBeg*(*L1*, *L2*) та чи входить один список у другий *IsIn*(*L1*, *L2*);

д) заміни всіх входжень у список елемента m числом n *Replace*(*L*, *m*, *n*).

10.16. Використовуючи модуль для реалізації класичного списку цілих чисел (див. завдання 10.15), скласти підпрограми:

а) *IsSymm*(*L*) — перевірка списку на симетричність;

б) *Copy*(*L*, *m*, *n*) — виділити з списку *L* *n* елементів, починаючи з елемента з номером *m* у новий список;

в) *Del*(*L*, *m*, *n*) — видалення *n* елеменів списку *L*, починаючи з *m*-го.

10.17. Використовуючи модуль для реалізації класичного списку цілих чисел (див. завдання 10.15), скласти підпрограми для реалізації додаткових дій над списком: *App*(*L*, *x*) – дописати елемент у кінець списку, *Lst*(*L*) – повернути останній елемент списку, *Bgn*(*L*) – повернути початок списку без останнього елемента.

10.18. Описати модуль для реалізації списку цілих чисел з поточним елементом. Передбачити виконання дій над списком:

1) Почати роботу.
2) Чи порожній залишок списку?
3) Встати до початку списку.
4) Перейти до наступного елемента.
5) Поточний елемент.
6) Вставити елемент.
7) Видалити елемент.

Використовуючи цей модуль, скласти підпрограми:

а) пошуку у списку елемента, рівного заданому числу *Search*(*L*, *x*);

б) сортування списку *Sort*(*L*);

в) конкатенації двох списків *Concat*(*L1*, *L2*);

г) присвоєння для списків *Let*(*L1*, *L2*) та обчислення довжини списку *Len*(*L*);

д) перевірки, чи є один список початком другого *IsBeg*(*L1*, *L2*) та чи входить один список у другий *IsIn*(*L1*, *L2*);

е) інверсії списку *Invert*(*L*).

10.19. Використовуючи модуль для реалізації списку цілих чисел з поточним елементом (див. завдання 10.18), скласти підпрограми:

а) *Change*(*L*, *n*) — замінити поточний елемент списку з поточним елементом *L* числом *n*;

б) *IsSymm*(*L*) — перевірка списку на симетричність;

в) *Copy*(*L*, *m*, *n*, *L1*) — виділити з списку *L* *n* елементів, починаючи з елемента з номером *m* у новий список *L1*;

г) *Del*(*L*, *m*, *n*) — видалення *n* елеменів списку *L*, починаючи з *m*-го.

10.20. Описати модуль для реалізації кільцевого списку цілих чисел. Передбачити виконання дій над списком:

1) Почати роботу.
2) Довжина списку.
3) Перейти до наступного елемента.

4) Поточний елемент.
5) Вставити елемент.
6) Видалити елемент.

Використовуючи цей модуль, розв'язати задачі:

а) «лічилка» (див. завдання 10.5);

б) пошук у списку елемента, рівного заданому числу *Search*(*L*, *x*) та присвоєння для списків *Let*(*L1*, *L2*);

в) знайти послідовність рівних елементів списку, що йдуть підряд, максимальної довжини;

г) видалити із списку всі повторні входження елементів;

д) знайти пару елементів списку, різниця між якими є максимальною за абсолютною величиною для всіх пар елементів списку.

10.21. Використовуючи модуль для реалізації кільцевого списку цілих чисел (див. завдання 10.20), скласти підпрограми:

а) *Change*(*L*, *n*) — замінити поточний елемент списку *L* числом *n*;

б) *IsSymm*(*L*) — перевірка списку на симетричність;

в) *Copy*(*L*, *m*, *n*, *L1*) — виділити з списку *L* *n* елементів, починаючи з елемента з номером *m* у новий список *L1*;

г) *Del*(*L*, *m*, *n*) — видалення *n* елеменів списку *L*, починаючи з *m*-го, по відношенню до поточного, елемента кільцевого списку.

10.22. Описати модуль для реалізації двозв'язного списку цілих чисел. Передбачити виконання дій над списком:

1) Почати роботу.
2) Чи порожній список?
3) Чи порожній початок списку?
4) Чи порожній кінець списку?
5) Встати до початку списку.
6) Встати до кінця списку.
7) Перейти до наступного елемента.
8) Перейти до попереднього елемента.
9) Поточний елемент.
10) Вставити елемент перед поточним.
11) Вставити елемент після поточного.
12) Видалити елемент.

Використовуючи цей модуль, розв'язати задачі:

а) пошуку у списку елемента, рівного заданому числу *Search*(*L*, *x*) та обчислення довжини списку *Len*(*L*);

б) сортування списку *Sort*(*L*);

в) конкатенації двох списків *Concat*(*L1*, *L2*) та присвоєння для списків *Let*(*L1*, *L2*);

г) перевірки, чи є один список початком другого *IsBeg*(*L1*, *L2*) та чи входить один список у другий *IsIn*(*L1*, *L2*).

10.23. Використовуючи модуль для реалізації двозв'язного списку цілих чисел (див. завдання 10.22), реалізувати на базі двозв'язного списку:

а) стек (розв'язати для стеку задачу інвертування послідовності символів);

б) чергу (розв'язати для черги задачу «лічилка» (див. завдання 10.5));

в) дек (розв'язати для деку задачу «лічилка» (див. завдання 10.5));

г) список з поточним елементом (розв'язати для списку задачу пошуку елемента у списку).

10.24. Використовуючи модуль для реалізації двозв'язного списку цілих чисел (див. завдання 10.22), скласти підпрограми:

а) *Change*(*L*, *n*) — замінити поточний елемент списку *L* числом *n*;

б) *Copy*(*L*, *m*, *n*, *L1*) — виділити з списку *L* *n* елементів, починаючи з елемента з номером *m* у новий список *L1*;

в) *Del*(*L*, *m*, *n*) — видалення *n* елеменів списку *L*, починаючи з *m*-го.

10.25. Описати модуль для реалізації бінарного дерева цілих чисел. Передбачити виконання дій над деревом:

1) Почати роботу.
2) Чи порожнє дерево?
3) Створити дерево.
4) Корінь дерева.
5) Лівий син.
6) Правий син.

Використовуючи цей модуль, розв'язати задачі:

а) виведення дерева на екран *Print*(*t*);

б) пошуку у дереві елемента, рівного заданому числу *Search*(*t*, *x*);

в) побудови бінарного дерева пошуку та пошуку елемента у ньому (бінарне дерева називають деревом пошуку, якщо для будь-якого його піддерева корінь цього піддерева не менше кожної вершини лівого сина та не більше кожної вершини правого сина);

г) обчислення висоти дерева *Height*(*t*);

д) перевірки, чи входить одне дерево у другие *IsIn*(*t1*, *t2*).

10.26. Описати модуль для реалізації сильно розгалуженого дерева цілих чисел. Передбачити виконання дій над деревом:

1) Почати роботу.
2) Чи порожнє дерево?
3) Створити вершину.
4) Корінь дерева.
5) Список синів.
6) Видалити вершину.
7) Почати роботу із списком синів.
8) Чи порожній залишок списку синів?
9) Встати до початку списку синів.
10) Перейти до наступного елемента списку синів.
11) Поточний елемент списку синів.
12) Вставити елемент у список синів.
13) Видалити елемент списку синів.

Використовуючи цей модуль, розв'язати задачі:

а) побудови бінарного дерева за сильно розгалуженим деревом;

б) пошуку у дереві елемента, рівного заданому числу *Search*(*t*, *x*);

в) обчислення висоти дерева *Height*(*t*);

г) перевірки, чи входить одне дерево у другие *IsIn*(*t1*, *t2*).

10.27. Описати модуль для реалізації графів. Передбачити виконання дій над графом:

1) Створити вершину.
2) Навантаження вершини.
3) Список попередників.
4) Список наступників.
5) Змінити список попередників.
6) Змінити список наступників.
7) Видалити вершину.
8) Почати роботу із списком вершин.
9) Чи порожній залишок списку вершин?
10) Встати до початку списку вершин.
11) Перейти до наступного елемента списку вершин.
12) Поточний елемент списку вершин.
13) Вставити елемент у список вершин.
14) Видалити елемент списку вершин.

Використовуючи цей модуль, розв'язати задачі:

а) додати вершину графа;

б) видалити вершину графа;

в) перевірити, чи існує шлях між двома вершинами;

г) знайти найкоротший шлях між двома вершинами.

11. ОБ’єктно-орієнтоване програмування

11.1. Описати клас *ЗНАЙОМИЙ* на базі класу Person

type Person = Object
Name,Surname: *String*;
BYear: *Integer*;
Procedure Input;
Procedure Output;
end;

та реалізувати програму пошуку телефону у телефонній книзі.

11.2. Описати клас *СПІВРОБІТНИК* на базі класу Person (див. завдання 11.1) та реалізувати програму розрахунку прибуткового податку.

11.3. Описати клас *ГІСТЬ* на базі класу Person (див. завдання 11.1) та реалізувати програму розрахунку плати за проживання у готелі.

11.4. Описати клас *ПАСАЖИР* на базі класу Person (див. завдання 11.1) та реалізувати програму розрахунку плати за білет в залежності від відстані маршруту.

11.5. Описати клас *ПОЛІНОМ* та реалізувати методи взяття похідної та добутку поліномів.

11.6. Описати клас *СЛОВО* та реалізувати програму пошуку однокореневих слів у файлі.

11.7. Описати клас *ДИНАМІЧНИЙ_МАСИВ*, реалізувати методи створення та видалення масиву, читання та зміни елемента. З використанням динамічних маисвів розв’язати наступну задачу. У двох масивах містяться коефіцієнти поліномів степеней m та n відповідно. Отримати коефіцієнти добутку цих поліномів.

11.8. Описати клас *СТЕК* з довільних об’єктів та реалізувати дії над стеками (див. завдання 10.1). Описати клас *СИМВОЛ* на базі початкового об’єкта та виконати інвертування послідовності символів.

11.9. Описати клас *ЧЕРГА* з довільних об’єктів та реалізувати дії над чергами (див. завдання 10.3). Описати клас *ПОКУПЕЦЬ* на базі початкового об’єкта та розв’язати задачу 10.6.

11.10. Описати клас *СПИСОК* з довільних об’єктів та реалізувати дії над списками (див. завдання 10.18). Описати клас *ЕЛЕМЕНТ* на базі початкового об’єкта та розв’язати задачу 10.18 е).

11.11. Побудувати об’єктно-орієнтований проект для виведення на екран графіків функцій однієї змінної.

11.12. Побудувати об’єктно-орієнтований проект для виведення на екран та переміщення фігур та реалізувати програму розміщення меблів.

11.13. Побудувати об’єктно-орієнтований проект для виведення на екран та переміщення об’ємних тіл.

11.14. Побудувати об’єктно-орієнтований проект для реалізації двозв’язного списку рядків та реалізувати перегляд текстових файлів.

11.15. Побудувати об’єктно-орієнтований проект для графічного зображення функцій 2 змінних.

11.16. Побудувати об’єктно-орієнтований проект для побудови опуклої оболонки точок тривимірного простору.