

UNIVERSITATEA TEHNICĂ A MOLDOVEI

**PROGRAMAREA
CALCULATORULOR**

Indicații de laborator

Lucrarea de laborator nr.2

Programe ramificate

1. Scopul

Studierea, utilizarea și obținerea deprinderilor practice de elaborare și depanare a programelor ramificate (if-else, switch-break).

2. Descrierea temei

2.1. Instrucțiunea *if* și *if-else*

Instrucțiunea *if* permite programarea unei structuri de decizie în care o condiție (rezultatul evaluării unei expresii) determină executarea sau neexecutarea secvenței de instrucțiuni.

Sintaxa instrucțiunii <i>if</i>	Schema bloc
<pre>if (expresie) { instrucțiuni; }</pre>	<pre>graph TD Start(()) --> Decision{expresie ?} Decision -- ADEVĂR --> Instructions[Instrucțiuni;] Decision -- FALS --> Instructions Instructions --> End(())</pre>

Modul de funcționare: la început se calculează valoarea expresiei din parantezele rotunde ale instrucțiunii **if**. Dacă valoarea acesteia este **ADEVĂR**, atunci se vor executa toate instrucțiunile cuprinse de parantezele ondulate("{ }"), pe care ulterior le vom numi corp al funcției. În caz contrar se ignoră instrucțiunile din corpul lui **if** și se trece la următoarele rânduri de cod.

Instrucțiunea *if-else* permite programarea unei structuri de decizie în care o condiție determină executarea unei secvențe de program din două alternative.

Sintaxa instrucțiunii <i>if-else</i>	Schema bloc
<pre> if (condiție) { Instrucțiune-1; } else { Instrucțiune-2; } </pre>	<pre> graph TD Entry(()) --> Cond{Condiție ?} Cond -- ADEVĂR --> I1[Instrucțiune-1;] Cond -- FALS --> I2[Instrucțiune-2;] I1 --> Exit(()) I2 --> Exit </pre>

Modul de funcționare: În primul rând se calculează valoarea expresiei din parantezele rotunde ale instrucțiunii *if*, dacă valoarea acesteia este **ADEVĂR**, atunci se vor executa toate instrucțiunile din corpul lui *if*. În caz contrar se ignoră instrucțiunile din corpul lui *if* și se execută cele din corpul lui *else*.

Exemplu: Să se scrie un program ce va utiliza instrucțiunile *if-else*:

```

#include<conio.h>
#include<stdio.h>
void main()
{ int n;
  printf("Introdu rezultatul testului:");
  scanf("%d", &n);
  if(n>=50)
  {
    printf("Ati trecut testul");
  }
  else
  {
    printf("Nu ati trecut testul");
  }
}

```

```
}  
}
```

În practica programării, din când în când apare necesitatea alegerii unei operațiuni din mai multe posibile, pentru aceasta se permite utilizarea lanț ului de instrucțiuni *if-else-if* sau instrucțiunea *switch()*, pe care o vom examina ceva mai târziu.

Exemplu: Să se scrie un program ce va utiliza lanțul de instrucțiuni *if-else-if*:

```
#include<conio.h>  
#include<stdio.h>  
void main()  
{  
    int n;  
    printf("Introdu rezultatul testului: ");  
    scanf("%d",&n);  
    if(n>=48)  
    {  
        printf("Rezultat maxim");  
    }  
    else  
    {  
        if(n>=40)  
        {  
            printf("Rezultat satisfacator");  
        }  
        else  
        {  
            printf("Ati picat testul");  
        }  
    }  
}
```

2.2. Instrucțiunea SWITCH

Instrucțiunea **switch** permite executarea unei singure secvențe din mai multe alternative, în funcție de valoarea unei expresii. Sintaxa instrucțiunii **switch** este următoarea:

Sintaxa instrucțiunii <i>switch</i>	Schema bloc
<pre> switch (expresie) { case const-1: { instructiune-1; break; } ... case const-n: { instructiune-n; break; } default: { instructiune-def; } } </pre>	<pre> graph TD Start(()) --> Expresie[expresie] Expresie --> Const1{const-1} Const1 -- Da --> Instr1[instructiune-1] Const1 -- Nu --> ConstN{const-n} ConstN -- Da --> InstrN[instructiune-n] ConstN -- Nu --> Default{default} Default -- Da --> InstrDef[instructiune-def] Default -- Nu --> Exit(()) Instr1 --> Exit InstrN --> Exit InstrDef --> Exit Exit --> End(()) </pre>

Modul de funcționare: Întâi se calculează valoarea expresiei „**expresie**”. Dacă aceasta este egală cu una din constantele corespunzătoare fiecărei etichete **case**, se execută instrucțiunea din corpul constantei respective. Fiecare grup **case** trebuie terminat cu o instrucțiune **break**. În cazul în care aceasta lipsește, se execută toate instrucțiunile, începând cu constanta selectată până la ultimul **case**. Când este întâlnită o instrucțiune **break** în construcția **switch**, programul execută un salt la linia de cod ce urmează după corpul lui **switch**. Instrucțiunea corespunzătoare cazului implicit (**default**) este executată când expresia nu ia nici una din valorile constantelor. Default este opțional și, dacă nu este prezent, nu are loc nici o acțiune dacă nu se găsește o constantă potrivită.

Pentru înțelegerea completă trebuie menționate următoarele:

- a) **switch** diferă de **if** prin aceea că testează doar egalitatea, în timp ce **if** poate să evalueze orice tip de expresie relațională sau logică;
- b) în același **switch** nu pot exista două constante **case** cu valori identice. Desigur, două instrucțiuni **switch**, una inclusă în cealaltă, pot să aibă aceeași constantă **case**.
- c) Dacă în instrucțiunea **switch** sunt utilizate constante de tip caracter, ele sunt automat convertite în întregi.

Exemplu: Să se scrie un program ce va efectua operații cu numere întregi de forma OPERAND1 operator OPERAND2 (2*4):

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
    int op1,op2,rez;
    char op;
    printf("\nScrieti expresia fara spatii intre
    operanzi si operator\n");
    scanf ("%d%c%d", &op1, &op, &op2);
    switch (op)
    {
        case '+':
            {
                rez=op1+op2; break;
            }
        case '-':
            {
                rez=op1-op2; break;
            }
        case '*':
```

```

    {
        rez=op1*op2; break;
    }
case '/':
    {
        if (op2!=0)
        {
            rez=op1/op2;
        }
        else
        {
            printf("Nu se imparte la zero!!!!");
        }
        break;
    }
default:
    {
        printf("Nu a avut loc o operatie!");
    }
}
printf("\n%d %c %d=%d", op1, op,op2, rez);
}

```

3. Variantele propuse spre elaborare

3.1. Să se scrie un program C pentru următoarele sarcini:

1. Se consideră două numere întregi distincte. Să se scrie un program care înlocuiește numărul mai mare cu produsul numerelor date, iar numărul mai mic – cu valoarea absolută a diferenței lor.
2. Se consideră trei numere întregi. Să se scrie un program care calculează media aritmetică a acestor numere dacă primul număr este mai mare decât al doilea, în caz contrar să se calculeze suma lor.
3. Se consideră două numere întregi. Să se scrie un program care calculează produsul acestor numere dacă numerele sunt nenule, în caz contrar să se determine și să se afișeze numărul mai mare dintre aceste numere.
4. Se consideră un număr natural N . Să se scrie un program care determină dacă numărul N este pozitiv sau negativ.
5. Se consideră două numere întregi. Dacă primul număr este negativ, atunci suma lor se va micșora de 5 ori, în caz contrar – se va calcula suma pătratelor acestor numere.
6. Se consideră două numere întregi. Dacă primul număr este mai mare decât modulul celui de al doilea număr, atunci se va calcula media lor aritmetică, în caz contrar – cubul sumei lor.
7. Se consideră trei numere întregi. Să se ridice la pătrat numerele pozitive, celelalte - să se mărească de 7 ori.
8. Se consideră trei numere întregi. Dacă suma primelor două este pozitivă, atunci să se calculeze media geometrică a acestor numere, în caz contrar – media aritmetică.
9. Se consideră trei numere întregi, două dintre care sunt egale.

Să se scrie un program care afișează numărul diferit de celelalte două.

10. Se consideră două numere întregi. Dacă primul este pozitiv, atunci se va calcula media geometrică a acestor numere, în caz contrar – al doilea număr se va mări cu 50.
11. Se consideră două numere întregi distincte. Să se scrie un program care înlocuiește numărul mai mare cu suma numerelor date, iar numărul mai mic – cu rădăcina pătrată a diferenței lor.
12. Se consideră trei numere întregi. Să se scrie un program care calculează rădăcina de ordinul 3 a sumei acestor numere dacă primele două numere sunt nenule, în caz contrar să se mărească de 3 ori suma lor.
13. Se consideră trei numere întregi. Să se scrie un program care calculează produsul acestor numere dacă numerele sunt nenule, în caz contrar să se calculeze suma pătratelor acestor numere.
14. Se consideră un număr natural N . Să se scrie un program care determină dacă numărul N este divizibil cu 3 sau nu.
15. Se consideră două numere întregi. Dacă primul număr este negativ, atunci se va calcula valoarea absolută a produsului acestor numere, în caz contrar – se va calcula suma pătratelor acestor numere.
16. Se consideră două numere întregi. Dacă primul număr este mai mare decât modulul celui de al doilea număr, atunci se va calcula media lor aritmetică, în caz contrar – pătratul sumei lor.

17. Se consideră trei numere întregi. Să se ridice la pătrat suma lor dacă primul număr este mai mare decât ultimul, în caz contrar - să se mărească de 5 ori fiecare dintre aceste numere.
18. Se consideră trei numere întregi. Dacă suma primelor două este pozitivă, atunci să se mărească cu 30 fiecare număr, în caz contrar – să se calculeze valoarea absolută a mediei aritmetice a acestor numere.
19. Se consideră trei numere întregi, două dintre care sunt egale. Să se scrie un program care afișează numărul diferit de celelalte două iar, pe celelalte numere să le mărească de 3 ori.
20. Se consideră un număr natural N . Să se scrie un program care determină dacă numărul N este divizibil cu 7 sau nu, în caz contrar – să se mărească acest număr cu pătratul lui.
21. Se consideră două numere întregi. Dacă primul este mai mare decât al doilea, atunci el se va micșora de 5 ori, în caz contrar – al doilea număr se va mări cu 150.
22. Se consideră trei numere naturale a , b și c . Să se determine dacă numărul b și c sunt divizori pentru numărul a .
23. Se dau două numere naturale. Să se determine dacă numerele date sunt vecine în șirul numerelor pare.
24. Se consideră trei numere întregi. Să se scrie un program care calculează produsul acestor numere dacă numerele sunt nenule, în caz contrar să se determine și să se afișeze numărul mai mic dintre aceste numere.

3.2. Să se scrie un program care va calcula valoarea funcției în funcție de condiții. În program să se utilizeze instrucțiunile *if*, *if-else*, *if-else-if*:

Nr.	Funcția	Condiția
1	$M = \begin{cases} \pi b^3 + a + x \\ e^{ax} + \sqrt[3]{xb} \\ 4tga \end{cases}$	$1 \leq x \leq 3$ $x < 1$ $x > 3$
2	$A = \begin{cases} \pi \sqrt{x^2 + b} - b^2 \\ \sin^3(x + a) + \sqrt[3]{x^2 + b} \\ bx \cos(x) + a \end{cases}$	$x < 1.5$ $x = 1.5$ $x > 1.5$
3	$L = \begin{cases} x^{y/x} - \sqrt[3]{y/x} \\ \cos^2 x^3 - e^{x/a} \\ \pi e^{-ax} \sqrt{a^2 + 1.5} \end{cases}$	$x < 1.2$ $x = 1.2$ $x > 1.2$
4	$D = \begin{cases} \pi a + x \\ (bx^2 - a) / (e^{ax} - 1) \\ x^2 a / 4 \end{cases}$	$x > a$ $x = a$ $x < a$
5	$Z = \begin{cases} e^{2y} \ln(a + x + y^2) \\ \sqrt[3]{x + y} \\ \cos^2 x + \sin y^4 + a^2 \end{cases}$	$-1 \leq x * y \leq 6$ $x * y < -1$ $x * y > 6$

6	$C = \begin{cases} 1.8 \cos^2 h + a \\ 1.8 \pi a h \\ \cosh + a \sin^2 h \end{cases}$	$\begin{aligned} h &> a \\ h &= a \\ h &< a \end{aligned}$
7	$V = \begin{cases} \pi d^3 - \sqrt{ d - g } \\ dg - e^{-d} \\ \sin dg + 1 \end{cases}$	$\begin{aligned} 1 \leq d^2 \leq 9 \\ d^2 < 1 \\ d^2 > 9 \end{aligned}$
8	$B = \begin{cases} \pi e^{-\sqrt{w}} \cos(bx/w) \\ \sqrt{wx \sin 2w + e^{-2x}(w+b)} \\ tg^2(w+x+b) \end{cases}$	$\begin{aligned} w &< 0.2 \\ w &= 0.2 \\ w &> 0.2 \end{aligned}$
9	$L = \begin{cases} 2 \cos(x - \pi/6) \\ x^2/a + \cos(x+b)^3 \\ x/2a + \sin^2(x+1) \end{cases}$	$\begin{aligned} x &< 1.2 \\ 1.2 \leq x \leq 3.9 \\ x &> 3.9 \end{aligned}$
10	$K = \begin{cases} 1.5 \cos^2 v + u \\ u^2/3 + v^2 + \pi \\ 3tgv + u^3 \end{cases}$	$\begin{aligned} -1 \leq v^*u \leq 5 \\ v^*u &< -1 \\ v^*u &> 5 \end{aligned}$
11	$J = \begin{cases} (a+b)/(e^x + \cos x) \\ (a+b)/(x+1) \\ e^x + \sin x + \pi \end{cases}$	$\begin{aligned} x &< 2.8 \\ 2.8 \leq x \leq 6 \\ x &> 6 \end{aligned}$
12	$G = \begin{cases} ca/i + bi^2 \\ i + 2i + 2 \\ ai + bi^3 \end{cases}$	$\begin{aligned} i &< 4 \\ 4 \leq i \leq 6 \\ i &> 6 \end{aligned}$

13	$F = \begin{cases} a \ln x + \sqrt[3]{ a - x } \\ 2a \cos x + 3x^2 \end{cases}$	$\begin{aligned} x &> 1 \\ x &\leq 1 \end{aligned}$
14	$T = \begin{cases} \sqrt{at^2} + b \sin t + a \\ at + b + \pi \\ 3ta^2 + \cos^3 t + 1 \end{cases}$	$\begin{aligned} t &< 0.1 \\ t &= 0.1 \\ t &> 0.1 \end{aligned}$
15	$K = \begin{cases} a \sin(i^2 + 1) \\ \cos(i + 1/n) \\ tgn + n^3 + \pi \end{cases}$	$\begin{aligned} i &< 1.5 \\ i &= 1.5 \\ i &> 1.5 \end{aligned}$
16	$WD = \begin{cases} \sin(2/k) \\ 1/k \\ k^2 + \pi \end{cases}$	$\begin{aligned} k &< 1 \\ k &= 1 \\ k &> 1 \end{aligned}$
17	$f(x) = \begin{cases} x^2 + \sqrt[3]{3x} \\ x + 1 \\ x^3 \end{cases}$	$\begin{aligned} x &< -5, \\ -5 \leq x &< 2, \\ x &\geq 2, \end{aligned}$
18	$f(x) = \begin{cases} 3x + x - 1 \\ 2\sqrt[3]{x + 5} \\ \log x \end{cases}$	$\begin{aligned} x &< 1, \\ 1 \leq x &< 6, \\ x &\geq 6; \end{aligned}$

19	$f(x) = \begin{cases} \cos x \\ 4 - \sqrt{ x - 6 } \\ \sin x \end{cases}$	$\begin{aligned} x &< 0, \\ x &= 0, \\ x &> 0; \end{aligned}$
20	$f(x) = \begin{cases} e^{2x} \ln(x + \cos^2 x) \\ x^2 + x \\ \sin \pi x \end{cases}$	$\begin{aligned} x &\leq 0, \\ 0 &< x \leq 2, \\ x &> 2. \end{aligned}$
21	$K = \begin{cases} \sqrt{nt^2} + \sin t \\ \cos(t + 1/n) \\ tgn + n^3 + \pi \end{cases}$	$\begin{aligned} t &< 0.1 \\ t &= 0.1 \\ t &> 0.1 \end{aligned}$
22	$K = \begin{cases} \pi a + x \\ (bx^2 - a)/(e^{ax} - 1) \\ x^2 a / 4 \end{cases}$	$\begin{aligned} 1 &\leq x \leq 3 \\ x &< 1 \\ x &> 3 \end{aligned}$
23	$L = \begin{cases} x^{y/x} - \sqrt[3]{y/x} \\ \cos^2 x^3 - e^{x/a} \\ \pi e^{-ax} \sqrt{a^2 + 1.5} \end{cases}$	$\begin{aligned} x &< 1.5 \\ x &= 1.5 \\ x &> 1.5 \end{aligned}$

24	$TZ = \begin{cases} e^{2y} \ln(a + x + y^2) \\ \sqrt[3]{x + y} \\ \cos^2 x + a^2 \end{cases}$	$\begin{aligned} -1 &\leq x*y \leq 6 \\ x*y &< -1 \\ x*y &> 6 \end{aligned}$
----	---	--

3.3. Să se scrie un program C pentru următoarele sarcini. În program să se utilizeze instrucțiunile **switch**:

1. De la tastatură se introduce un număr natural n, ce reprezintă numărul de ordine a lunii anului. Să se scrie un program C ce va afișa denumirea anotimpului din care face parte luna respectivă.
2. De la tastatură se introduce un număr natural n, ce reprezintă numărul de zile a lunii anului. Să se scrie un program C ce va afișa lunile cu numărul respectiv de zile.
3. Se citește o cifră de la tastatură. Să se scrie un program C ce va afișa denumirea în litere a cifrei respective în limba engleză.
4. De la tastatură se introduce un caracter (literă majusculă), ce reprezintă notația unei note muzicale. Să se scrie un program C ce va afișa denumirea notei muzicale respective. Indicație: Notele muzicale au notațiile C (do), D (re), E (mi), F (fa), G (sol), A (la), H (si).
5. De la tastatură se introduce un caracter (literă majusculă), ce reprezintă prefixul unui multiplu al unităților de măsură. Să se scrie un program C ce va afișa prefixul multiplului corespunzător literei date. Indicație: Literele și multiplii corespunzători sunt: K (kilo), M (mega), G (giga), T (tera), P

(peta).

6. De la tastatură se introduce un caracter, ce reprezintă o cifră a sistemului de numerație roman. Să se scrie un program C ce va afișa numărul zecimal corespunzător valorii cifrei respective. Indicație: Cifrele romane sunt reprezentate prin literele(mari și mici): I,V, X, L, C, D, M.
7. Conform calendarului japonez, fiecare an poartă numele unui animal. Fiecare denumire se repetă exact o dată la 12 ani. Deci, un ciclu este format din 12 ani cu următoarele denumiri de animale în această ordine: șobolan, bou, tigru, iepure, dragon, șarpe, cal, oaie, maimuță, cocoș, câine, porc. Știind că anul 2020 este anul șobolanului, să se scrie un program C care va citi de la tastieră anul (număr de patru cifre) și va afișa denumirea lui conform calendarului japonez.
8. De la tastatură se introduce un număr natural ce reprezintă data unei zile din luna octombrie, anul 2020. Să se scrie un program C ce va afișa denumirea zilei pentru data respectivă. Se știe că 1 octombrie 2020 este (a fost) joi.
9. De la tastatură se introduce unul din următoarele numere: 1, 5, 10, 50, 100, 500 sau 1000. Să se scrie un program C ce va afișa cifra romană corespunzătoare numărului citit.
10. De la tastatură se citește un caracter. Să se scrie un program C ce va afișa ce fel de caracter a fost citit, literă mare, literă mică, cifră sau un caracter oarecare. Se admit numai litere ale alfabetului latin.
11. Se citește de la tastatură numărul de ordine al zilei săptămânii. Să se scrie un program C ce va afișa orarul lecțiilor pe care îl aveți în ziua corespunzătoare.

12. Se citește o cifră de la tastatură. Să se scrie un program C ce va afișa denumirea în litere a cifrei respective.
13. Se citește de la tastatură un n umăr natural, $N \leq 100\,000$. Să se scrie un program C ce va afișa numărul de cifre a acestui număr.
14. Se citește de la tastatură un n umăr natural, $0 < N \leq 10$. Să se scrie un program C ce va afișa denumirea simbolului chimic aflat în tabelul periodic al lui Mendeleev corespunzător numărului dat.
15. De la tastatură se introduce un caracter, ce reprezintă categoria de vehicule pentru care permisul este valabil (A, B, C, D, E, F, H). Să se scrie un program C ce va afișa denumirea vehiculelor pentru care permisul este valabil în categoria dată.
16. Din cele mai vechi timpuri, ciclul vieții ființei umane a fost împărțit în trei perioade: vârsta I (dezvoltarea, copilăria, tinerețea, 0 - 18 ani); vârsta a II-a (adult, 19 - 64 ani); vârsta a III-a (vârstnicul, îmbătrânirea, ≥ 65 ani). Fiecare vârstă menționată conține, la rândul ei, alte subetape: < 1 an, copil sub un an; 1 - 3 ani, copil mic; 4 - 6 ani, copil preșcolar; 7 - 10 ani, școlar mic; 11 - 15 ani, pubertate; 16 - 18 ani, adolescență; 19 - 35 ani, adult tânăr; 36 - 50 ani, adult matur; 51 - 64 ani, presenescență; 65 - 75 ani, vârstnic; 76 - 85 ani, bătrân; > 85 ani, longeviv. De la tastatură se introduce un număr natural ce reprezintă vârsta în ani. Să se scrie un program C ce va afișa perioada vârstei și subetapa în care se află vârsta indicată.
17. Schimbările ce au loc în viața ființei umane împart ciclul vieții de familie în 8 etape bine definite: Etapa I: familia cu tânăr/ă îndrăgostit/ă ce părăsește familia nucleară; Etapa II:

căsătoria – etapa de formare a unei noi familii; Etapa III: nașterea primului copil – extensia familiei; Etapa IV: familia cu școlar mic – integrarea socială; Etapa V: familia cu adolescenți; Etapa VI: familia în perioada de contracție familială – „criza” de la mijlocul vieții; Etapa VII: familia cu persoane de vârstă a III-a, pensionarea – criza socială; Etapa a VIII-a: familia cu persoana rămasă singură, doliul, îmbătrânirea. De la tastatură se introduce un număr natural ce reprezintă etapa ciclului vieții de familie. Să se scrie un program C ce va afișa descrierea etapei respective.

18. Se introduce un număr natural de la tastatură, $0 < n \leq 10$. Să se scrie un program C ce va afișa denumirea destinației turistice aflat pe locul n printre cele top 10 atracții turistice din Republica Moldova.

19. Clasificarea spectrală Harvard categorizează stelele în următoarele clase cu următoarele caracteristici: O (temperatura 30000-60000 K, culoarea convențională - albastru), B (temperatura 10000-30000 K, culoarea convențională - alb-albastru), A (temperatura 7500-10000 K, culoarea convențională - alb), F (temperatura 6000-7500 K, culoarea convențională - alb-gălbui), G (grupa soarelui) (temperatura 5000-6000 K, culoarea convențională - galben), K (temperatura 3500-5000 K, culoarea convențională - portocaliu), M (temperatura 2000-3500 K, culoarea convențională - roșu). Se introduce un caracter de la tastatură ce va indica clasa stelară. Să se scrie un program C ce va afișa denumirea și descrierea clasei respective.

20. Se introduce un număr natural n ($1 \leq n < 9$). Să se scrie un program C ce va afișa denumirea facultății din cadrul UTM corespunzător numărului înscris:

1. Facultatea Electronică și Telecomunicații
2. Facultatea Energetică și Inginerie Electrică

3. Facultatea Calculatoare, Informatică și Microelectronică
4. Facultatea Tehnologia Alimentelor
5. Facultatea Inginerie Mecanică, Industrială și Transporturi
6. Facultatea Urbanism și Arhitectură
7. Facultatea Construcții, Geodezie și Cadastru
8. Facultatea Inginerie Economică și Business
9. Facultatea Textile și Poligrafie

21. Se introduce un caracter de la tastatură ce reprezintă una din mulțimile numerelor reprezentate mai jos. Să se scrie un program C care va afișa denumirea mulțimei conform caracterului introdus.

$$\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$$

N - numere naturale;

Z - numere întregi;

Q - numere raționale;

R - numere reale;

C - numere complexe.

22. De la tastatură se introduce un număr natural ce reprezintă data unei zile din luna septembrie, anul 2020. Să se scrie un program C ce va afișa denumirea zilei pentru data respectivă. Se știe că 1 septembrie 2020 a fost marți.

23. De la tastatură se introduce un număr natural n, ce reprezintă numărul de ordine a lunii anului. Să se scrie un program C ce va afișa personalități remarcabile născute în această lună (cel puțin o personalitate pentru o lună).

24. Se citește de la tastatură un număr natural, $10 < N \leq 20$. Să se scrie un program C ce va afișa denumirea simbolului chimic aflat în tabelul periodic al lui Mendeleev corespunzător numărului dat.