

Varianta 2

Subiectul I

ex.1

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Variabilele x și y sunt tip `int`. Care dintre expresiile C/C++ de mai jos are valoarea 1 dacă și numai dacă valorile întregi nenule memorate în variabilele x și y sunt egale? (4p.)
- a. $(x\%y==0) \ \&\& \ (y\%x==0) \ \&\& \ (x*y>0)$ b. $(x<=y) \ \&\& \ (y<x)$
c. $(x<=y) \ || \ (y<=x)$ d. $x*x==y*y$

Raspuns: a)

ex.2

2. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg nenul y .

- a) Scrieți ce se va afișa dacă se citesc, în această ordine, numerele: 17 22 13 101 2 7 5 0. (6p.)

- b) Scrieți un șir de date de intrare, format doar din numere naturale cu cel mult două cifre fiecare, care să determine afișarea valorii 9877. (4p.)

- c) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura `cât timp...execută` cu o structură repetitivă de alt tip. (6p.)

- d) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```
citește x (număr natural nenul)
cât timp x>0 execută
    citește y (număr natural)
    dacă x>y atunci
        scrie x%10
    altfel
        scrie y%10
    x←y
```

a)

Se citeste 17

17>0 Adevarat

Se citeste y <- 22

17>22 fals

=> Afiseaza 22%10 = 2

x <- y => x = 22

Se citeste 13

22>13 adevarat

=> Afiseaza $22\%10 = 2$

x = 13

Se citeste 101

13>101 fals

=> Afiseaza $101\%10 = 1$

x = 101

Se citeste 2,7,5

101>2 adevarat

=> Afiseaza $101\%10 = 1$

x = 2

Se citeste 7

2>7 fals

=> Afiseaza 7

x = 7

Se citeste 5

7>5 adevarat

=> Afiseaza 7

x = 5

Se citeste 0

5>0 adevarat

=> Afiseaza 5

x = 0

x>0 fals (0>0 fals)

=> Se afiseaza 2211775

b)

39 38 37 37 0

```
17 22 13 101 2 7 5 0
2211775
Process returned 0 (0x0)   execution time : 4.917 s
Press any key to continue.
```

Motiv:

- Algoritmul verifica ultima cifra a fiecarui numar citit.
- Primul numar citit intra in variabila x
- La fiecare pas, algoritmul citeste o variabila y si verifica daca ultima cifra a numarului stocat in x este mai mare decat ultima cifra a numarului stocat in y. Daca ultima cifra a lui x este mai mare, aceasta se afiseaza si vice-versa.
- Algoritmul stocheaza in fosta variabila x variabila y. Astfel, x ia valoarea lui y.
- Acesti pasi se repeata pana cand algoritmul detecteaza ca numarul citit este 0. In acest caz, algoritmul va scrie numarul precedent lui 0 si va incheia rulara.
- De aici putem deduce faptul ca, pentru ca la final sa se afiseze sirul 9877:
 1. Ultima cifra a ultimului numar citit trebuie sa fie 7.
 2. Ultima cifra a penultimului numar trebuie sa fie, de asemenea, tot 7.
 3. Ultima cifra a numarului precedent penultimului numar trebuie sa fie 8
 4. Ultima cifra a primului numar trebuie sa fie 9.
- De aici putem forma o multitudine de siruri care ajung la afisarea dorita. Eu am dat spre exemplu sirul 39,38,37,37,0. Un alt exemplu poate fi sirul 9,8,7,7,0.

c)

citeste x (numar natural nenul)

```
daca x>0 atunci
|
|cat timp x>0 executa
||
||citeste y (numar natural)
||
||daca x>y atunci
|||
|||scrie x%10
|||altfel
|||  scrie y%10
||-
||x<-y
|-
-
```

d)

d) Scrieți programul C/C++ corespunzător algoritmului dat.

```
1 //variantebac.info
2 #include <bits/stdc++.h>
3 using namespace std;
4 int main()
5 {
6     int x,y;
7     cin >> x;
8     while(x>0){
9
10         cin >> y;
11         if(x>y)
12             cout << x%10;
13         else cout << y%10;
14         x = y;
15     }
16
17 }
18
```

Subiectul II

ex.1

1. Câte grafuri neorientate, distincte, cu 8 vârfuri se pot construi? Două grafuri se consideră distincte dacă matricele lor de adiacență sunt diferite. (4p.)
- a. 4^{14} b. 2^{14} c. 4^{28} d. 64

Raspuns: a)

Aplicam formula nazdravana: numarul de grafuri neorientate distincte care se pot forma cu n varfuri este: $2^{(n(n-1)/2)}$.

$$2^{((8*7)/2)} = 2^{28}$$

Nu gasim 2^{28} in raspunsurile posibile, asadar vom rescrie numarul. $2^{28} = (2^2)^{14} = 4^{14}$. Aceasta este varianta a).

ex.2

2. Variabila `t`, declarată alăturat, memorează în câmpurile `a`, `b` și `c` lungimile laturilor unui triunghi. Care dintre următoarele instrucțiuni atribuie câmpului `p` al variabilei `t` valoarea perimetrului triunghiului respectiv? (4p.)
- | | |
|--|--|
| <pre>a. p.t=t.a+t.b+t.b;
c. t.p=t.a+t.b+t.c;</pre> | <pre>struct triunghi
{
 float a,b,c,p;
}t;
b. p.t=a.t+b.t+c.t;
d. t.p==t.a+t.b+t.c;</pre> |
|--|--|

Raspuns: c)

Atentie la sintaxa.

ex.3

3. Se consideră o stivă în care inițial au fost introduse, în această ordine, elementele cu valorile 1, 2 și 3. Se notează cu **AD** (*x*) operația prin care se adaugă elementul cu valoarea *x* în vârful stivei și cu **EL** operația prin care se elimină elementul din vârful stivei. Asupra acestei stive se execută următoarea secvență de operații: **AD** (4) ; **EL** ; **AD** (5) ; **EL** ; **AD** (6) ; **EL** ; **EL**.
- a) Care este valoarea elementului din vârful stivei în urma executării acestei secvențe de operații? (3p.)
- b) Care este suma valorilor elementelor aflate în stivă în urma executării acestei secvențe de operații? (3p.)

Ordinea de operatii este in felul urmator:

- In stiva se afla elementele 1,2,3. De la varf in jos, acestea sunt 3,2,1.
- Se adauga elementul 4.
- Se elimina elementul de pe varful stivei. Acesta este ultimul element introdus, adica 4.

- Stiva devine 3,2,1.
- Se adauga elementul 5.
- Se elimina elementul de pe varful stivei. Acesta este ultimul element introdus, adica 5.
- Stiva devine 3,2,1.
- Se adauga elementul 6.
- Se elimina elementul de pe varful stivei. Acesta este ultimul element introdus, adica 6.
- Stiva devine 3,2,1
- Se elimina elementul de pe varful stivei. Acesta este ultimul element introdus, adica 3.
- Stiva devine 2,1

a)Elementul din varful stivei este 2. Raspunsul pentru subpunctul a este 2

b)Suma elementelor din stiva, la finalul executarii acestor operatii, este 2+1, adica 3.

ex.4

4. În secvența de program alăturată, variabila **a** memorează o matrice cu **n** linii și **n** coloane (numerotate de la 0 la **n-1**) cu elemente numere **întregi**, iar toate celelalte variabile sunt întregi. Știind că **n** este un număr natural **par**, nenul, scrieți instrucțiunile care pot înlocui punctele de suspensie din secvența de program alăturată astfel încât, în urma executării acesteia, să se afișeze ultima cifră a sumei elementelor **pozitive** de pe linia **k** ($0 \leq k < n$) a matricei **a**. (6p.)

```
s = 0;
for(j = 0; j < n/2; j++)
{
    .....
}
printf("%d", s); | cout<<s;
```

Raspuns:

```
12 for(int j = 0; j < n/2; j++){
13     if(a[k-1][j]>0)
14         s+= a[k-1][j];
15     if(a[k-1][n-j-1]>0)
16         s+=a[k-1][n-j-1];
17     s=s%10;}
```

ex.5

5. Să se scrie un program C/C++ care citește de la tastatură un cuvânt format din cel mult 20 de caractere, doar litere mici ale alfabetului englez. Programul determină transformarea cuvântului citit prin înlocuirea fiecărei vocale a cuvântului, cu un șir format din două caractere și anume vocala respectivă urmată de litera mare corespunzătoare, restul literelor nemodificându-se, ca în exemplu. Programul afișează pe ecran cuvântul obținut, pe o singură linie. Se consideră vocale literele din mulțimea {a,e,i,o,u}.

Exemplu: pentru cuvântul bacalaureat se va afișa pe ecran: baAcaAlaAuUreEaAt
(10p.)

```
1 //variantebac.info
2 #include <bits/stdc++.h>
3 using namespace std;
4 char c[42],aux[100];
5 int main()
6 {
7     int j =0;
8     cin.get(c,21);
9     char voc[]="aeiou";
10    for(int i=0;i<strlen(c);i++)
11    {
12
13        aux[j]=c[i];
14        j++;
15        if(strchr(voc,aux[j-1])){
16            aux[j++]=aux[j-1]-32;}
17    }
18    strcpy(c,aux);
19    cout << c;
20 }
21
```

Subiectul III

ex.1

1. Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea $A = \{a, b, c, d, e\}$, cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: **abab, abac, abad, abba, abbb, abbc, abbd, abbe**. Care este ultimul cuvânt generat? (4p.)

a. edcb b. eeee c. edde d. eded

Raspuns: d)

- a) Nu convine, metoda backtracking genereaza cuvintele in ordine lexicografica. "edcb" nu poate fi ultimul cuvânt generat. Ultimul cuvânt generat trebuie sa aibe ultima litera posibila in ultimul loc posibil ("d" pe pozitia 4)
- b) Nu convine, cuvântul contine doua vocale alaturate.
- c) Nu convine, acelasi motiv ca a)
- d) Convine.

ex.2

2. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(12345)**? (6p.)

<pre>//C void f(long n) { printf("%d",n%10); if(n!=0) { f(n/100); printf("%d",n%10); } }</pre>	<pre>//C++ void f(long n) { cout<<n%10; if(n!=0) { f(n/100); cout<<n%10; } }</pre>
--	--

Raspuns:

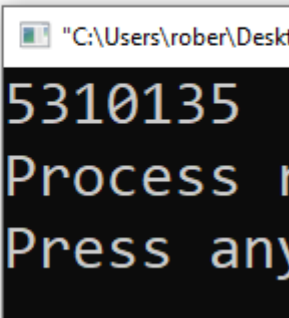
Se afiseaza 5310135

- $f(12345)$ -> se afiseaza $n\%10$, adica 5. $n!=0$ adevarat, astfel se apeleaza functia $f(n/100)$, adica $f(123)$
- $f(123)$ -> se afiseaza 3. $n!=0$ adevarat, se apeleaza functia $f(1)$
- $f(1)$ -> se afiseaza 1. $n!=0$ adevarat, se apeleaza functia $f(0)$
- $f(0)$ -> se afiseaza 0. $n!=0$ fals. Iesim din $f(0)$, ne intoarcem la functia $f(1)$
- $f(1)$, continuare: se afiseaza $n\%10$, adica 1. Iesim din $f(1)$, ne intoarcem la functia $f(123)$

- f(123), continuare: se afiseaza 3. Ne intoarcem la functia f(12345)
- f(12345), continuare: se afiseaza 5.

La final se va afisa 5310135.

```
1  #include <iostream>
2
3  using namespace std;
4  void f(long n) {
5      cout << n%10;
6      if (n!=0)
7          { f(n/100); cout << n%10; }
8
9  }
10 int main()
11 {
12     f(12345);
13 }
14
```



ex.3

3. Fișierul text **NR.TXT** conține pe o singură linie, separate prin câte un spațiu, cel mult 100 de numere **întregi**, fiecare număr având cel mult 4 cifre. Scrieți un program C/C++ care citește numerele din fișierul **NR.TXT** și afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, toate numerele **naturale nenule** din fișier. Dacă nu există astfel de numere se va afișa pe ecran mesajul **NU EXISTA**.

Exemplu: dacă fișierul **NR.TXT** conține numerele: -3 -10 0 7 -5 7 51 -800 6 3798, atunci pe ecran se va afișa: 6 7 7 51 3798 (10p.)

```
1  //variantebac.info
2  #include <bits/stdc++.h>
3  using namespace std;
4  ifstream fin("NR.TXT");
5  int v[101];
6  int main()
7  {
8      int q = 0;
9      int x = -1;
10     while(fin >> x) {
11         if(x>0)
12             v[q++] = x;
13     }
14     for(int i=0;i<q;i++)
15         for(int j=i+1;j<q;j++)
16             {if(v[j]<v[i]) swap(v[j],v[i]);}
17     for(int i=0;i<q;i++)
18         cout << v[i] << " ";
19 }
20
```

ex.4

4. Un număr n se numește **extraprim** dacă atât el, cât și orice număr obținut prin permutarea cifrelor lui n , sunt numere prime. De exemplu, numărul 113 este un număr **extraprim** deoarece 113, 311, 131 sunt numere prime.

a) Scrieți definiția completă a unui subprogram f , cu un parametru, subprogram care:

- primește prin intermediul parametrului a un număr natural cu cel mult 3 cifre ($a > 1$)
- returnează suma tuturor exponenților din descompunerea în factori primi a valorii parametrului a .

Exemplu: pentru $a=90$ subprogramul va returna valoarea 4, deoarece $a=2 \cdot 3^2 \cdot 5$ și $1+2+1=4$. (4p.)

b) Scrieți un program C/C++ care citește de la tastatură un număr natural n , $2 \leq n \leq 999$ și, folosind apeluri utile ale subprogramului f , verifică dacă n este un număr **extraprim**. În caz afirmativ, programul afișează pe ecran mesajul **DA**, în caz contrar afișând mesajul **NU**. (6p.)

a)

```
5  int f(int a) {
6      int s = 0;
7      for(int i=2; i<=a*a; i++) {
8          int exp = 0;
9          if(a%i==0) {
10             while(a%i==0)
11                 {a/=i; exp++;}
12             s+=exp;
13         }
14     }
15     return s;
16 }
17
```

b)

```
1 //variantebac.info
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 int f(int a){
6     int s = 0;
7     for(int i=2;i<=a*a;i++){
8         int exp = 0;
9         if(a%i==0){
10             while(a%i==0)
11                 {a/=i; exp++;}
12             s+=exp;
13         }
14     }
15     return s;
16 }
17
18 int n;
19 int main()
20 {
21     cin >> n;
22     //Primul if: daca n are 3 cifre si permutarile acestuia sunt toate numere prime (f(numar) = 1 => nr. este prim)
23     //Atunci afiseaza DA.
24     if(n>=100 && f(n)==1 && f(n/10%10*100+n%10*10+n/100)==1 && f(n%10*100+n/100*10+n/10%10)==1){
25         cout << "DA";
26     }
27     //Altfel afiseaza NU
28     else if(n>=100) cout << "NU";
29     //n are 2 cifre si permutarile acestuia sunt numere prime, afiseaza DA
30     else if(n/10>0 && f(n)==1 && f(n%10*10+n/10)==1) cout << "DA";
31     //Altfel afiseaza NU
32     else if(n/10>0) cout << "NU";
33     //n este cifra si este numar prim, atunci afiseaza DA
34     else if(n<10 && f(n)==1) cout << "DA";
35     //Altfel afiseaza NU
36     else cout << "NU";
37 }
38
```