

Examenul de bacalaureat simulare
Proba E. d)
Proba scrisă la INFORMATICĂ
Limbajul C/C++

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

SUBIECTUL I

(20 de puncte)

1. Indicați expresia care are valoarea 1 dacă și numai dacă numărul reala memorat în variabila x se afla în intervalul [-5,5].

- a. $x*x-25<0$ b. $25-x*x>=0$ c. $(5<=x)\&\&(x<=-5)$ d. $(x-5)*(x+5)$

2. Subprogramul f este definit alăturat. Scrieți ce se afișează în urma apelului de mai jos. f(12);

```
void f (int n)
```

```
{ int i;
```

```
for(i=2;i<=n/2;i++)
```

```
if(n%i==0) { cout<<i; f (n/i); } }
```

```
2346232
```

b. 2233246

c. 2346

d. 223323224362

3. Utilizând metoda backtracking, se generează toate cadourile formate din câte 3 obiecte distincte din mulțimea {**carte, tableta, joc, stilou, ceas, patine**} astfel încât să nu se găsească în același cadou **tableta si joc** sau **carte si stilou**. Primele patru soluții obținute sunt, în aceasta ordine: (**carte, tableta, ceas**), (**carte, tableta, patina**), (**carte, joc, stilou**), (**carte, joc, patine**). Indicați a șasea soluție generată:

- a. tableta,joc,patine b. joc, stilou, ceas c. carte, stilou, ceas d. tableta, stilou,ceas

4. Se considera urmatorul arbore cu 10 noduri dat prin vectorul de tati (8,9,9,9,4,4,5,5,0,5). Cate noduri pot fi alese ca radacina astfel incat inaltimea arborelui sa fie maxima?

- a. 3 b. 5 c. 2 d. 6

5. Un graf neorientat are 30 muchii si fiecare nod al grafului are gradul un numar nenul. Doar sase noduri au gradul un numar par, restul nodurilor avand gradele numere impare. Care este numarul maxim de noduri pe care le poate sa le aiba graful?

- a. 36 b. 54 c. 30 d. 60

SUBIECTUL II

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $a\%b$ restul împărțirii numărului natural a la numărul natural nenul b și cu [a] partea întreagă a numărului real a.

citește p,q (numere naturale nenule, $p \leq q$)

```
x ← p
```

```
┌cât timp x ≤ q execută
```

```
| y ← x
```

```
| c ← y%10
```

```
| ┌cât timp y ≠ 0 și y%10 = c execută
```

```
| | y ← [y/10]
```

```
| └─┘
```

```
| ┌dacă y = 0 atunci
```

```
| | scrie x, ' '
```

```
| └─┘
```

```
| x ← x+1
```

```
└─┘
```

a) Scrieți valorile afișate dacă se citește, în această ordine, numerele 65 și 80. (6p.)

b) Dacă pentru variabila p se citește numărul 1234, scrieți cel mai mare număr de patru cifre care poate fi citit pentru variabila q astfel încât, în urma executării algoritmului, să se afișeze 5 numere. (6p.)

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

d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura cât timp...execută cu o structură de tip pentru...execută. (6p.)

2. Variabilele f și fd, declarate alăturat, memorează în câmpurile x și y numărătorul, respectiv numitorul câte unei fracții.

```
struct fractie
{ int x; int y; } f,fd;
```

Scrieți o secvență de instrucțiuni care să memoreze în variabila fd fracția obținută prin scăderea fracției 1/2017 din fracția memorată în variabila f. (6p.)

3. În secvența alăturată variabilele n și i sunt de tip întreg, iar variabila s permite memorarea unui cuvânt, șir de cel mult 20 de caractere. Cuvintele citite sunt formate din litere mici ale alfabetului englez și sunt separate prin Enter.

```
n=.....;
for(i=1;i<=11;i++)
{ cin>>s;
.....
}
```

Fără a utiliza alte variabile, scrieți secvența înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila n să memoreze numărul de cuvinte citite pentru care subșirul format din ultimele trei caractere ale lor coincide cu șirul are, iar acesta NU mai apare pe alte poziții în cuvânt. Exemplu: dacă se citesc cuvintele alăturate, variabila n are valoarea 3. (6p.)

la
proiectare
se
are
in
vedere
evitarea
deformatiilor
areolare
si
liniare

SUBIECTUL III

(30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un număr natural, n ($n \in [2, 20]$), apoi n numere naturale din intervalul $[0, 10^4]$, reprezentând, de la stânga la dreapta, în această ordine, valorile elementelor aflate pe prima linie a unui tablou bidimensional cu n linii și n coloane. Programul construiește în memorie tabloul, inițializând celelalte elemente, astfel încât fiecare linie să se obțină prin permutarea circulară a elementelor liniei anterioare, de la stânga spre dreapta, cu o poziție, ca în exemplu. Programul afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu. Exemplu: dacă se citesc numerele n=4, apoi 1, 1, 3, 2, se obține tabloul alăturat. (10p.)

```
1 1 3 2
2 1 1 3
3 2 1 1
1 3 2 1
```

2. Scrieți un program C/C++ care citește de la tastatură, în această ordine, un număr natural n, apoi n cuvinte, separate prin Enter, urmate de un număr natural k. Numerele citite sunt din intervalul $[1, 20]$, iar fiecare cuvânt este format din cel mult 20 de caractere, numai litere mici ale alfabetului englez. Cel puțin unul dintre cuvinte are k litere. Programul construiește în memorie, apoi afișează pe ecran, un șir care să cuprindă cuvintele citite, separate prin câte un spațiu, astfel încât toate cele care au k litere să ocupe primele poziții, iar celelalte să se regăsească în continuarea acestora, într-o ordine oarecare. Exemplu: dacă se citesc, în această ordine, datele alăturate,

5
am
un
cires
de
mai
2

unul dintre șirurile obținute poate fi: am un de cires mai (10p.)

3. Un text are cel mult 100 de caractere, iar cuvintele sale sunt formate doar din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat mai sus și îl transformă în memorie prin înlocuirea fiecărui cuvânt format din număr par de litere cu simbolul #. Programul afișează pe ecran textul obținut sau mesajul nu exista dacă textul citit nu conține astfel de cuvinte. Exemplu: pentru textul:

anii de liceu sunt foarte frumoși

se afișează

liceu # # frumoși