

SUBIECTUL al II-lea

(30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

- Un arbore cu 9 noduri, numerotate de la 1 la 9, este reprezentat prin vectorul de „tați” (3, 3, 0, 5, 2, 5, 2, 5, 8). Descendenții direcți (“fii”) ai nodului cu eticheta 5 sunt: **(4p.)**
 - 2 7
 - 2 8
 - 3 3 0
 - 4 6 8
- Numărul de noduri ale unui graf neorientat fără cicluri, cu 26 de muchii și 12 componente conexe este: **(4p.)**
 - 18
 - 28
 - 38
 - 48

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

- Variabila `d`, declarată alăturat, memorează în câmpul `mic` cel mai mic divizor, strict mai mare decât 1, al numărului natural din intervalul $[2, 10^2]$, memorat în câmpul `nr`.

```
struct divizor  
{ int nr, mic;  
  } d;
```

Scrieți o secvență de instrucțiuni în urma executării căreia, pentru numărul memorat în câmpul `nr` al variabilei `d`, se afișează pe ecran mesajul `prim`, dacă numărul este prim, mesajul `patrat` dacă numărul este pătratul unui număr prim, sau două numere naturale, separate printr-un spațiu, reprezentând cel mai mic și cel mai mare dintre divizorii proprii pozitivi ai săi. Divizorii proprii pozitivi ai unui număr sunt divizori pozitivi diferiți de 1 și de el însuși.
Exemplu: dacă în câmpul `nr` se memorează numărul 12, iar în câmpul `mic` se memorează numărul 2, se afișează pe ecran
2 6
iar dacă în câmpul `nr` se memorează numărul 9, iar în câmpul `mic` se memorează numărul 3, se afișează pe ecran mesajul
`patrat` **(6p.)**
- Variabilele `i` și `j` sunt de tip întreg, iar variabila `a` memorează un tablou bidimensional cu 9 linii și 9 coloane, numerotate de la 1 la 9, având inițial toate elementele nule.
Fără a utiliza alte variabile, scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila `a` să memoreze tabloul alăturat.

```
for(i=1;i<=9;i++)  
  for(j=1;j<=9;j++)  
  .....
```

(6p.)

1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2
1	2	3	3	3	3	3	3	3
1	2	3	4	4	4	4	4	4
1	2	3	4	5	5	5	5	5
1	2	3	4	5	6	6	6	6
1	2	3	4	5	6	7	7	7
1	2	3	4	5	6	7	8	8
1	2	3	4	5	6	7	8	9
- Un text are cel mult 100 de caractere și este format din cuvinte, numere naturale și spații. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Cuvintele și numerele sunt separate prin câte un spațiu, ca în exemplu.
Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat mai sus și afișează pe ecran numărul din text care începe cu cea mai mare cifră, ca în exemplu. Dacă există mai multe astfel de numere, se afișează doar unul dintre acestea, iar dacă textul nu conține niciun număr, se afișează pe ecran mesajul `nu exista`.
Exemplu: pentru textul
`am 190 de nuci si 70 de castane`
se afișează
70 **(10p.)**

SUBIECTUL al III-lea

(30 de puncte)

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

1. Subprogramele `f1` și `f2` sunt definite mai jos.

```
int f1 (int x, int y)           int f2 (int x, int y)
{ if(x%2!=0 || y%2!=0) return 1; { if (x==y) return x;
  else return 2*f1(x/2,y/2);     else
}                                 if(x>y) return f2(x-y,y);
                                 else return f2(x, y-x);
}                                 }
```

Cel mai mare divizor comun al lui 30 și 50 se obține în urma apelului: **(4p.)**

- a. `f1(30,50)` b. `f2(30,50)` c. `f1(30/2,50)` d. `f2(30/2,50)`

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Utilizând metoda backtracking, se generează toate posibilitățile de a forma cutii cu bomboane de tipuri distincte din mulțimea {**fondante, caramele, dropsuri, acadele**}. Într-o cutie sunt cel puțin două tipuri de bomboane, dar nu pot fi și dropsuri și acadele simultan. Două cutii sunt distincte dacă ele conțin cel puțin un tip diferit de bomboane. Primele patru soluții generate sunt, în această ordine, (**fondante, caramele**), (**fondante, caramele, dropsuri**), (**fondante, caramele, acadele**), (**fondante, dropsuri**). Scrieți a cincea și a șasea soluție, în ordinea generării acestora. **(6p.)**

3. Un număr natural este numit **echilibrat** dacă suma cifrelor sale de pe poziții pare este un număr par, iar suma cifrelor sale de pe poziții impare este un număr impar. Pozițiile cifrelor sunt numerotate de la dreapta la stânga, astfel: cifra unităților este pe poziția 0, cifra zecilor este pe poziția 1 ș.a.m.d.

Subprogramul **echilibrat** are un singur parametru, **n**, prin care primește un număr natural ($n \in [10, 10^9]$). Subprogramul returnează valoarea 1 dacă **n** este echilibrat sau valoarea 0 în caz contrar.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **n=25163912**, subprogramul returnează valoarea 1, iar dacă **n=11211**, subprogramul returnează valoarea 0. **(10p.)**

4. Numim **secvență încadrată** a unui șir de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat, subșir care începe și se termină cu aceeași valoare. Lungimea secvenței este egală cu numărul de termeni ai acesteia.

Fișierul **bac.txt** conține un șir de cel puțin două și cel mult 10^6 numere naturale din intervalul $[0, 9]$. Numerele sunt separate prin câte un spațiu. În șir există cel puțin doi termeni egali.

Se cere să se determine secvențele încadrate din acest șir care au lungime maximă și să se afișeze pe prima linie a ecranului lungimea maximă determinată, iar pe următoarea linie, pentru fiecare astfel de secvență, valoarea primului său termen. Numerele de pe a doua linie sunt afișate în ordine strict crescătoare, separate prin câte un spațiu.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

Exemplu: dacă fișierul **bac.txt** conține numerele

3 1 5 2 4 5 5 2 5 9 5 7 4 6 8 0 8

atunci pe ecran se afișează valorile:

9

4 5

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8p.)**