

- ```

citește x,y (numere naturale)
z←1
t←0
cât timp x≥z execută
┌   dacă x%z=y atunci
│       t←z
└   z←z+1
scrie t

```

**Subiectul II (30 de puncte)**

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

1. Graful neorientat **G** este dat prin matricea de adiacență alăturată.  
Câte vârfuri ale grafului **G** au gradul 1?

(4p.)

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |

- a. 1                                      b. 2                                      c. 3                                      d. 0
2. Într-o listă simplu înlănțuită nevidă, fiecare element reține în câmpul **urm** adresa elementului următor din listă sau **NULL** dacă nu are un nod următor. Știind că variabila **p** reține adresa primului element din listă, variabila **q** reține adresa ultimului element din listă, iar lista este formată din exact 3 elemente, care dintre următoarele secvențe de instrucțiuni determină, în urma executării, eliminarea celui de-al doilea element din listă? (4p.)
- a. `delete p->urm; | free(p->urm);`  
`p->urm=q;`
- b. `p->urm=q;`  
`delete p->urm; | free(p->urm);`
- c. `p=q;`  
`delete p->urm; | free(p->urm);`
- d. `p->urm=q;`  
`delete q; | free(p->urm);`

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

3. Pentru reprezentarea unui arbore cu rădăcină cu 9 noduri, etichetate cu numere naturale de la 1 la 9, se utilizează vectorul de „tați”: **T**=(2,0,1,7,3,1,2,4,1). Care sunt descendenții direcți ai rădăcinii și câte frunze are arborele dat? (6p.)
4. Variabila **a** memorează elementele unui tablou bidimensional cu 4 linii și 5 coloane, numere întregi. Care este cel mai mic element situat pe a doua linie a tabloului construit în urma executării secvenței de program alăturate? (6p.)
- ```
for (i=1; i<=4; i++)  
    for (j=1; j<=5; j++)  
        a[i][j]=i-j;
```
5. Scrieți programul **C/C++** care citește de la tastatură un șir de cel mult 40 de caractere, format doar din litere ale alfabetului englez, și care afișează pe ecran toate șirurile obținute prin eliminarea succesivă a câte unei singure litere din șirul citit, ca în exemplu. Șirurile obținute se vor afișa câte unul pe câte o linie a ecranului.  
**Exemplu:** dacă se citește șirul **abbcc** atunci pe ecran se va afișa:
- ```
bbc  
abc  
abc  
abb
```
- (10p.)

**Subiectul III (30 de puncte)**

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

1. Utilizând metoda backtracking se generează toate cuvintele de câte 3 litere din mulțimea  $\{a, b, c\}$ . Dacă primele patru cuvinte generate sunt, în această ordine: **aaa**, **aab**, **aac**, **aba**, care este cel de-al optulea cuvânt generat? (4p.)
- a. **acb**                      b. **acc**                      c. **aca**                      d. **bca**

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

2. Subprogramul **f** are definiția alăturată. Ce valoare are **f(4)**? Dar **f(11)**? (6p.)
- |  |                                                                                |
|--|--------------------------------------------------------------------------------|
|  | <pre>int f(int x) {     if(x&lt;1) return 1;     else return f(x-3)+1; }</pre> |
|--|--------------------------------------------------------------------------------|
3. Scrieți programul **C/C++** care citește de la tastatură numărul natural **n** ( $0 < n < 100$ ) și un șir format din **n** numere reale. Aceste numere au atât partea întreagă cât și partea fracționară formate din cel mult trei cifre. Programul determină și afișează pe ecran toate numerele din șir care apar o singură dată în acesta.
- Exemplu:** Dacă **n=7** iar șirul este format din elementele (3.4, -151, 0.291, 3.4, 4.09, 3.4, 0.291) atunci pe ecran se va afișa -151 4.09. (10p.)
4. Fișierul text **numere.txt** conține pe prima linie un număr natural **n** ( $0 < n < 100000$ ) iar pe a doua linie **n** numere naturale, formate din cel mult 2 cifre, separate prin câte un spațiu.
- a) Scrieți un program **C/C++**, eficient atât din punct de vedere al timpului de executare, care afișează pe ecran toate numerele situate pe a doua linie a fișierului, în ordinea crescătoare a valorilor lor, separate prin câte un spațiu.
- Exemplu:** dacă fișierul **numere.txt** are următorul conținut:
- ```
7
12 21 22 11 9 12 3
```
- atunci pe ecran se va afișa: 3 9 11 12 12 21 22 (6p.)
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)