



2. Variabila `c` memorează raza și coordonatele (abscisa și ordonata), în planul  $xOy$ , ale centrului unui cerc. Știind că expresiile `C/C++` de mai jos au valori reale, reprezentând raza, respectiv coordonatele centrului cercului, scrieți definiția unei structuri cu eticheta `cerc`, care permite memorarea datelor precizate, și declarați corespunzător variabila `c`.

`c.raza`      `c.centru.x`      `c.centru.y` (6p.)

3. Variabila `i` este de tip întreg, iar variabila `s` poate memora un șir de cel mult 20 de caractere. Scrieți ce se afișează în urma executării secvenței alăturate. (6p.)
- ```
strcpy(s,"stilou");
cout<<s+4<<endl; | printf("%s\n",s+4);
for(i=0;i<4;i++)
    s[i]=s[0]+(i-1)*(1-i%2)+3*(2*i/3-1)*(i%2);
s[4]='\0';
cout<<s; | printf("%s",s);
```

### SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul `putere` are trei parametri:
- `n`, prin care primește un număr natural din intervalul  $[2, 10^9]$ ;
  - `d` și `p`, prin care furnizează divizorul prim, `d`, care apare la cea mai mică putere, `p`, în descompunerea în factori primi a lui `n`; dacă există mai mulți astfel de divizori se afișează cel mai mic dintre ei.

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă `n=10780`, atunci, în urma apelului, `d=5` și `p=1` ( $10780=2^2 \cdot 5 \cdot 7^2 \cdot 11$ ). (10p.)

2. Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 20]$ , `n` și `k`, și construiește în memorie un tablou bidimensional cu `n·k` linii și `n` coloane, numerotate începând cu 1, astfel încât fiecare coloană `i` ( $i \in [1, n]$ ) memorează un șir crescător de termeni cu proprietatea că primul termen este `i`, fiecare valoare apare în șir de exact `k` ori și oricare doi termeni alăturați au valori egale sau consecutive. Programul afișează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu valorile aflate pe aceeași linie separate prin câte un spațiu.

**Exemplu:** dacă `n=4` și `k=3`, se afișează pe ecran tabloul alăturat.

(10p.)

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 5 |
| 2 | 3 | 4 | 5 |
| 2 | 3 | 4 | 5 |
| 3 | 4 | 5 | 6 |
| 3 | 4 | 5 | 6 |
| 3 | 4 | 5 | 6 |
| 4 | 5 | 6 | 7 |
| 4 | 5 | 6 | 7 |
| 4 | 5 | 6 | 7 |

3. Șirul de mai jos este definit astfel:  $f_1=1$ ,  $f_2=2$ ,  $f_n=3 \cdot f_{n-1}-2 \cdot f_{n-2}$  (unde `n` este un număr natural  $n \geq 3$ ).  
1, 2, 4, 8, 16, 32, ...

Se citește de la tastatură un număr natural `x` ( $x \leq 10^9$ ), valoare a unui termen al șirului dat, și se cere să se scrie în fișierul text `bac.txt`, în ordine descrescătoare, separați prin câte un spațiu, toți termenii șirului care sunt mai mici sau egali cu `x`. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă se citește numărul 16

fișierul `bac.txt` conține numerele 16 8 4 2 1

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

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