

2. Variabila t memorează coordonatele reale (abscisa și ordonata), în planul xOy , ale fiecăruia dintre cele trei vârfuri A , B și C ale unui triunghi. Știind că expresiile $C/C++$ de mai jos au ca valori abscisa vârfului A respectiv ordonatele vârfurilor B și C ale triunghiului, scrieți definiția unei structuri cu eticheta `triunghi`, care permite memorarea datelor precizate, și declarați corespunzător variabila t .
- `t.A.x t.B.y t.C.y` (6p.)

3. În secvența alăturată, variabila a memorează un șir cu cel mult 100 de caractere, iar variabilele i și k sunt de tip întreg. Scrieți ce se afișează pe ecran în urma executării secvenței.
- ```

k=' a' -' A' ;
strcpy(a,"VICtorIE");
cout<<strlen(a); | printf("%d", strlen(a));
for(i=0;i<strlen(a);i++)
 if(a[i]>='A' && a[i]<='Z') a[i]=a[i]+k;
 else a[i]=a[i]-k;
cout<<a; | printf("%s",a);

```
- (6p.)

### SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul `putere` are trei parametri:
- $n$ , prin care primește un număr natural din intervalul  $[1, 10^9]$ ;
  - $d$  și  $p$ , prin care furnizează divizorul prim,  $d$ , care apare la cea mai mare putere,  $p$ , în descompunerea în factori primi a lui  $n$ ; dacă există mai mulți astfel de divizori se afișează cel mai mare dintre ei.
- Scrieți definiția completă a subprogramului.
- Exemplu:** dacă  $n=10780$ , atunci, în urma apelului,  $d=7$  și  $p=2$  ( $10780=2^2 \cdot 5 \cdot 7^2 \cdot 11$ ). (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  $n$  linii și  $n \cdot k$  coloane, numerotate începând cu 1, astfel încât fiecare linie  $i$  ( $i \in [1, n]$ ) 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.
- |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 |
| 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 |
| 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 | 6 | 6 | 6 |
| 4 | 4 | 4 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 |
| 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 8 | 8 | 8 |
- Exemplu:** dacă  $n=4$  și  $k=3$ , se afișează pe ecran tabloul alăturat. (10p.)
3. Se consideră șirul  $1, 1, 2, 5, 13, 34, 89, 233, 610 \dots$  definit astfel:  $f_1=f_2=1$ ,  $f_n=3 \cdot f_{n-1}-f_{n-2}$  (unde  $n$  este un număr natural  $n \geq 3$ ):
- Se citesc de la tastatură două numere naturale  $x$  și  $y$  ( $x \leq y \leq 10^9$ ), valorile a doi termeni aflați pe poziții consecutive în șirul 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  $y$ . Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
- Exemplu:** dacă se citesc numerele **89 233**
- fișierul `bac.txt` conține numerele **233 89 34 13 5 2 1 1**
- a. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)
- b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)