

2. Pentru un cerc se memorează coordonatele reale (abscisa și ordonata) ale centrului cercului, în sistemul de coordonate xOy . Variabilele $c1x$ și $c1y$ memorează abscisa, respectiv ordonata centrului unui cerc, iar variabilele $c2x$ și $c2y$ memorează abscisa, respectiv ordonata centrului unui alt cerc. Declarați corespunzător cele patru variabile și scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran, separate printr-un spațiu, coordonatele centrului cercului care are centrul pe axa absciselor. Dacă ambele cercuri au centrul pe această axă, se afișează mesajul **ambele**, iar dacă niciunul dintre cercuri nu are centrul pe această axă se afișează mesajul **niciunul**. (6p.)
3. Variabila **a** este de tip **char**, iar variabilele **i** și **k** sunt de tip întreg; de la tastatură se citesc 16 litere mari ale alfabetului englez. Fără a utiliza alte variabile, scrieți secvența înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila **k** să memoreze numărul de litere mari citite, cu excepția celor din mulțimea {**A, C, E, I**}.

Exemplu: dacă se citesc literele de mai jos, variabila **k** memorează valoarea 8.

P, R, A, S, L, E, A, C, E, L, V, O, I, N, I, C

```
k=.....  
for(i=1;i<=16;i++)  
{ cin>>a; | scanf("%c",&a);  
.....  
} (6p.)
```

SUBIECTUL al III-lea

(30 de puncte)

1. Se citesc două numere naturale, **n** și **c** ($c \leq 9$), și se cere să se scrie numărul obținut din **n**, prin eliminarea din acesta a tuturor cifrelor egale cu **c**, sau -1 dacă toate cifrele lui **n** sunt egale cu **c**. Cifrele nule nesemnificative sunt ignorate, ca în exemplu. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. **Exemplu:** dacă $n=50752$ sau $n=72$ și $c=5$ se scrie 72, dacă $n=500$ și $c=5$ se scrie 0, iar dacă $n=55$ și $c=5$ se scrie -1. (10p.)
2. Scrieți un program C/C++ care citește de la tastatură două numere naturale, **n** și **k** ($n \in [3, 20]$, $k \in [2, n]$), și construiește în memorie un tablou unidimensional cu **n** elemente, având proprietățile:
- al **k**-lea element are valoarea 0;
 - tabloul conține, începând cu prima poziție, până la al **k**-lea element inclusiv, de la stânga la dreapta, un șir strict descrescător de numere consecutive, iar începând cu al **k**-lea element, până la ultima poziție, de la stânga la dreapta, un șir strict crescător de numere consecutive.
- Programul afișează pe ecran elementele tabloului construit, separate prin câte un spațiu. **Exemplu:** dacă $n=10$ și $k=3$ se obține tabloul (2, 1, 0, 1, 2, 3, 4, 5, 6, 7). (10p.)
3. Fișierul **bac.in** conține un șir de cel puțin trei și cel mult 10^5 numere întregi nenule din intervalul $[-10^9, 10^9]$, dintre care două sunt negative, iar restul pozitive. Numerele sunt separate prin câte un spațiu. O secvență este formată din termeni aflați pe poziții consecutive în șir, iar lungimea secvenței este egală cu numărul de termeni ai acesteia. Se cere să se afișeze pe ecran lungimea unei secvențe din șirul aflat în fișier care conține o singură valoare negativă și un număr maxim de valori pozitive. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare. **Exemplu:** dacă fișierul conține numerele 15 21 -61 9 870 -23 11 5 8 pe ecran se afișează 6 (corespunzător secvenței 9 870 -23 11 5 8).
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)