

1. Subprogramul P este definit alăturat. Indicați ce se va afișa în urma apelului P(1,5).


```
void P(int i, int j)
{
    int k;
    if (i<=j)
    {
        P(i+1, j-1);
        for (k=j; k>=i; k--) cout<<k;
    }
}
```

a. 543214323 b. 343254321 c. 323412345 d. 34324321

2. Variabila m memorează simultan datele fiecăreia dintre cele **300** mașini înscrise în catalogul de vânzări 2022: numărul de km și prețul (numere naturale de maxim 6 cifre) respectiv consumul la 100 km (exprimat printr-un număr real). Scrieți definiția unei structuri cu eticheta **vauto**, care să permită memorarea datelor referitoare la o mașină și declarați variabila m astfel încât să fie corecte aceste accesări: `m[10].km`, `m[10].pret`, `m[10].consum`

3. Un text are cel mult 100 de caractere, iar cuvintele sale sunt formate numai din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un cuvânt de maxim 10 caractere și apoi un text de tipul precizat mai sus și afișează pe ecran cuvântul din stânga și din dreapta cuvântului citit, separate prin câte un spațiu și pe rânduri separate dacă există mai multe astfel de grupuri. Dacă în șir nu există unul din cele trei cuvinte (cuvântul dat, cuvântul din stânga, cuvântul din dreapta), se va afișa mesajul **NU EXISTA**
 Exemple :
 Pentru cuvântul `bac` și textul `el a fost la bac si apoi a plecat cu un bac la mare`
 datele afișate vor fi:
`la si`
`un la`
 Pentru cuvântul `bac` și textul `el a fost la bac`
 se va afișa mesajul **NU EXISTA**

4. Fișierul **bac.in** conține pe prima linie un număr natural **k** ($3 \leq k \leq 100$) și pe a doua linie un șir de minim **k** și maxim **10⁹** numere naturale, din intervalul **[0, 10⁹]**, separate prin câte un spațiu. Se cere să se afișeze pe ecran câte secvențe de **k** elemente consecutive din șirul citit de pe a doua linie a fișierului au proprietatea că valorile care le formează dau resturi distincte două câte două la împărțirea la **k**. Dacă nu există nicio secvență cu proprietatea cerută, se va afișa pe ecran valoarea **0**. Proiectați un algoritm de rezolvare eficient din punct de vedere al memoriei utilizate și al timpului de execuție.
Exemplu: dacă fișierul **bac.in** conține numerele
`4`
`9 4 4 3 16 50 101 2 16 3 16 85 3`
 pe ecran se va afișa **2** (secvențele sunt 3 16 50 101 respectiv 101 2 16 3).
 a. Scrieți programul C/C++corespunzător algoritmului proiectat.
 b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia.

5. Se consideră un tablou unidimensional în care elementele sunt, în această ordine, **(2,7,10,15,22,63,72)**. Pentru a verifica dacă în tablou există elementul cu valoarea **x**, se aplică metoda căutării binare. Scrieți toate valorile pe care le poate avea **x** astfel încât căutarea să se încheie după ce **x** a fost comparat cu exact două elemente ale tabloului.
 a. 7,22 b.63,7 c.10,63 d.15,63