

Construire vector

Sunt probleme în care vectorul nu se citește de la tastatură, ci se construiește prin atribuire de valori. Identificăm două situații:

1.Când se cunoaște de la început numărul elementelor care se introduc în vector. Sunt problemele în care se cere pentru fiecare element citit introducerea în vector a unui element corespunzător. Așadar numărul elementelor din vector va fi egal cu numărul elementelor citite.

Algoritmul de construire al unui astfel de vector este următorul:

- se parcurg elementele date (care pot fi elementele unui alt vector sau elemente care atunci se citesc de la tastatură)
- pentru fiecare element citit se adaugă în vector pe poziția corespunzătoare elementul corespunzător (în funcție de ceea ce se cere în problemă)
- se afișează vectorul, care va avea tot atâtea elemente câte s-au citit

Codul C++:

```
//se parcurg elementele date; fie i variabila care indică la al câtelea element s-a ajuns
```

```
    v[i] = .....; // se completeaza cu valoarea corespunzatoare in functie de cerinta problemei
```

```
for(i=0;i<n;i++)
```

```
    cout<<v[i]<<" "; //se afiseaza elementele vectorului; n este numarul elementelor din vector
```

Exemplu: Se citesc de la tastatură n numere naturale nenule cu cel mult 8 cifre fiecare ($n \in [2, 10^4]$) și un număr natural nenul k cu cel mult 8 cifre. Se cere să se construiască și să se afișeze un vector care să conțină pentru fiecare dintre cele n numere citite cel mai mare divizor comun dintre numărul respectiv și k .

```
#include<iostream>
using namespace std;
int v[10000],n,i,x,k,ck,r;
int main()
{cin>>n>>k;
  for(i=0;i<n;i++) //se citesc cele n valori; incepem parcurgerea de la 0 pentru a incepe vectorul de la 0
  {cin>>x;          // x este numarul curent
    ck=k;          //se face o copie a lui k pentru ca se distruge atunci cand calculam cmmdc
    while(ck!=0)   //si nu s-ar mai putea calcula cmmdc dintre urmatorul numar si k
    {r=x%ck;      // se calculeaza cmmdc dintre numar si k
      x=ck;
      ck=r;
    }
    v[i]=x; //se memoreaza in vector cmmdc care este memorat in x
  }
  for(i=0;i<n;i++) //se afiseaza vectorul
    cout<<v[i]<<" ";
  return 0;
}
```

2.Când nu se cunoaște de la început numărul elementelor care se introduc în vector. Sunt problemele în care se cere introducerea în vector doar a valorilor care îndeplinesc anumite condiții date. Nu se poate ști de la început numărul valorilor care îndeplinesc condiția.

Algoritmul de construire al unui astfel de vector este următorul:

- o variabilă m care în final va memora numărul elementelor din vector se inițializează cu 0 (pentru că la început vectorul nu conține niciun element)
- se parcurg elementele date (care pot fi elementele unui alt vector sau elemente care atunci se citesc de la tastatură)
- se verifică dacă elementul curent îndeplinește condiția cerută în problemă
- în caz afirmativ:
 - se adaugă elementul în vector pe poziția m
 - crește variabila m cu o unitate (pentru că s-a adăugat un nou element în vector)
- în final se verifică dacă $m=0$
 - în caz afirmativ se afișează un mesaj care arată că vectorul este vid (niciun element nu a îndeplinit condiția dată); acest mesaj este omis dacă nu o cere problema
 - în caz negativ se afișează elementele vectorului (sunt m elemente în vector și primul element este pe poziția 0)

Codul C++:

```
m=0;
//se parcurg elementele date
//daca elementul curent x indeplineste conditia data
{v[m]=x; //se adauga elementul in vector
 m++;} //creste numarul elementelor din vector
if(m==0)
    cout<<"vector vid";
else
    for(i=0;i<m;i++)
        cout<<v[i]<<" ";
```

Exemplu: Se citesc de la tastatură n numere naturale cu cel mult 8 cifre fiecare ($n \in [2, 10^4]$). Se cere să se construiască și să se afișeze un vector care să conțină doar numerele pare dintre cele citite. În cazul în care niciunul dintre numerele citite nu este par se afișează mesajul NU EXISTA.

```
#include<iostream>
using namespace std;
int v[10000],n,i,x,m=0;
int main()
{cin>>n;
 for(i=1;i<=n;i++) // se citesc cele n valori
 {cin>>x; // x este numarul curent
 if(x%2==0) //daca numarul citit este par
 {v[m]=x; //se adauga elementul in vector
 m++;} //creste numarul elementelor din vector
 }
if(m==0) //daca nu sunt elemente in vector
    cout<<"NU EXISTA";
else //daca sunt elemente in vector se afiseaza vectorul
for(i=0;i<m;i++)
    cout<<v[i]<<" ";
return 0;
}
```