

## POLJA

- Polje je podatkovna struktura koja sadrži podatke istog tipa
- Često se koristi kada se radi sa nepoznatim brojem podatka istog tipa
- Polje može zauzimati različiti broj bajtova u memoriji
- Polja mogu biti različitih dimenzija, a najčešće su jednodimenzionalna
- Primjer deklaracije polja

long artikli [5];

Jednodimenzionalno polje artikli tipa long sa 5 elemenata. U memoriji zauzima 20 bajtova ( 5x4 bajta ), a možemo ga predstaviti tablicom:

0	1	2	3	4

- Elementima polja pristupa se preko indeksa koji su uvijek cijeli, nenegativni broj
- Indeksi svakog polja rasprostranjeni su u intervalu  $[0, n-1]$ , gdje  $n$  predstavlja broj elemenata polja
- Svaki element polja zasebna je varijabla koja ima neku vrijednost
- Polja se kao i varijable mogu inicijalizirati na neku početnu vrijednost, npr. `int ocjena[5]={1};` - nulti element imao bi vrijednost 1, a svi ostali 0
- Mogu se inicijalizirati i sve vrijednosti polja npr. `int ocjena[5]={1,2,3,4,5};`
- Ako ne inicijaliziramo vrijednosti polja tada su one neodređene ( a ne 0 )
- Zadatak 1.

Napiši program koji će učitati  $n$  prirodnih brojeva u intervalu  $[1,100]$ . Program treba ispisati koliko je puta unesen svaki od tih brojeva u zadanom intervalu  $[1,100]$ . Ako se unese broj izvan zadanog intervala treba ga zanemariti.

Naputak:

Za rješenje zadatka očito je potrebno 100 varijabli u kojima ćemo registrirati broj ponavljanja nekog broja u intervalu  $[1,100]$ . Stoga deklariramo polje tipa integer sa 100 elemenata koji će služiti kao brojači. Unesena vrijednost u zadanom intervalu ustvari će biti indeks polja.

Rješenje:

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main()
{
    int i,n,br,brojevi[100]={0};
    cout<<"Koliko prirodnih brojeva zelite
unijeti?"<<endl;
    cin>>n;
    for(i=0;i<n;i++){
        cout<<"UNESITE " <<(i+1)<<"broj";
        cin>>br;
        if((br>=1)&&(br<=100))
            brojevi[br-1]++;
    }
    for(i=0;i<100;i++)
        if(brojevi[i]!=0)
            cout<<(i+1)<<" -> " <<brojevi[i]<<"\t";

    system("PAUSE");
    return 0;
}
```