

Massiivid

- Kui meil on üksikud muutujad, siis võib igaühte deklareerida ja vaadata eraldi
- Kui meil on neid palju, nad on sama tüüpi ja me ei pruugi enne ülesande lahendamist teada palju neid täpselt vaja läheb, siis on otstarbekam grupeerida need üheks indekseeritud üksuseks
- Näiteks, eelmise tunni kolmnurga ülesanne oli kolme XY koordinaadiga, tülikas oli nende jaoks 6 muutujat kasutada

Massiivi loomine

- Massiivi loomisel tuleb määrata selle pikkus
- C keel ei luba lihtsasti massiivi pikkust hiljem muuta (saab teha mäluoperatsioonidega, aga neid vaatab aines Programmeerimine 2)
- Pikkus antakse loomise hetkel ette nurksulgudes

```
int arvud[10]; //loob 10 elemendilise massiivi arvud
```

- Konstandi asemel lubab C keel pikkuseks ette anda ka muutujat, aga sel juhul jääb kehtima muutuja väärtus, mis sel hetkel oli:

```
int n = 5;
```

```
int M[n]; //massiiv luuakse pikkusega 5
```

```
n = 7; //massiiv jääb ikka pikkusega 5
```

Massiivi elementide poole pöördumine

- Üksikutel elementidel on indeks, elemendi poole pöördumisel tuleb see indeks ette anda nurksulgudes
- Massiivi töötlemisel saab pöörduda suvalise elemendi poole
- C keeles algab indeks nullist, st esimene element on indeksiga 0
- Kui massiivi pikkus on n , siis indeksi väärtus võib olla vahemikus $0..n-1$
- Indeks võib olla ka muutuja, pöördutakse elemendi poole, mis on muutuja hetkeväärtus, indeksmuutuja
- C ei kontrolli, ega me massiivi piire pole ületanud, seda peab jälgima ise

```
int arvud[3] = {2, 3, 5};
```

```
arvud[1] // See on teine element väärtusega 3
```

```
arvud[7] // See oleks kaheksas element, aga  
         // piiridest väljas!
```

Massiivi pikkus

- Massiivi loomisel eraldatakse mälu, indeks viitab aadressile
Algus + Indeks * Ühe_elementi_suurus
- Kuna lihtsasti ei saa massiivi pikkust hiljem muuta, siis võib selle defineerida varuga ja hoida tegelikku pikkust mingis abimuutujas

```
#define MAX 100
int massiiv[MAX]; //massiiv loodud pikkusega 100
int n = 0; //abimuutuja n, palju on kasutatud
scanf("%d", &massiiv[n]); //loeme esimese elemendi
n++; //1 element on kasutatud
```

Ülesanded massiividega

- Ülesanded (järjestatud) hulkadega
- Hulga alamosa – filtreerimine
- Statistika hulga või alamhulga kohta – agregeerimine
- Hulga järjestamine vastavalt tingimusele – sortimine
- Mitu hulka, nende ühisosa, ühend, vahe jne – relatsioonilised operatsioonid
- Tavaliselt kombineeritud ülesanded

Algoritmid massiividega

- Algoritmides tuleb ütelda, kui mingi muutuja on massiivi tüüpi
- Massiivi elementide poole pöördumist peab näitama kas C stiilis või abstraktselt
 - indeks $i = 0$, massiivi element $M[i]$, i muutub (näiteks tsüklis), $i < n$ (massiivi pikkus)
 - võtame massiivi esimese elemendi, võtame järgmise elemendi, kas on veel elemente
- Vaatame mõnda näidet
 - Filtreerime paaritud arvud uude massiivi
 - Massiivi elementide summa