

# Alokacija.

## Argumenti komadne linije.

## Programi u više datoteka.

February 19, 2014

### 1 Alokacija

**Zadatak 1** Napisati funkciju kojom se od matrice  $A$  formira niz brojeva  $b[0], \dots, b[n-1]$  čiji su elementi jednaki razlici najvećih i najmanjih elemenata u vrsti matrice.

**Zadatak 2** Kao argument komadne linije zadata je dimenzija niza  $n$ . Alocirati prostor za niz i učitati  $n$  brojeva iz datoteke `brojevi.txt`

**Zadatak 3** Učitavaju se brojevi sa standardnog ulaza sve dok se ne učita 0. Nakon toga se unosi broj  $x$ . Udvojiti svako ponavljanje  $x$  u nizu i ispisati novi niz.

npr. 1 2 3 2 4 0 2  $\rightarrow$  1 2 2 3 2 2 4

### 2 Argumenti komadne linije

**Zadatak 4** Napisati program koji ispisuje broj navedenih argumenata komadne linije, a zatim i same argumente sa rednim brojevima.

**Zadatak 5** Ako su celi brojevi  $a$  i  $b$  argumenti komandne linije napraviti niz  $A[0] = a, A[1] = a+1, A[2] = a+2, \dots, A[b-a] = b$  i ispisati ga.

### 3 Program koji se sastoji iz više jedinica prevodjenja

**Zadatak 6** 1. Napisati funkcije

```
void unos(int a[], int n); (n se ne unosi u funkciji, već se prosleđuje prilikom poziva)
```

```
void ispis(int a[], int n);
```

koje služe za unos i ispis nizova.

2. Napisati f-ju koja računa sumu niza sa prototipom

```
int suma(int a[], int n);
```

3. Napisati f-ju koja računa skalarni proizvod dva vektora iste dimenzije sa prototipom

```
int skalarni(int a[], int b[], int n);
```

4. Unose se koeficijenti polinoma u niz, napisati f-ju koja ispisuje polinom u obliku  $a[0] + a[1]*x + a[2]*x^2 + \dots + a[n]*x^n$ . Funkcija ima prototip

```
void ispis_polinoma(int a[], int n);
```

gde je  $n$  stepen polinoma, a ne dužina niza. Na primer, za  $a = \{1,2,3\}$ ,  $n=2$ , funkcija treba da ispiše  $1 + 2*x + 3*x^2$ .

5. Napisati funkciju za sumiranje dva polinoma (u opštem slučaju različitog stepena):
- ```
int suma_polinoma(int a[], int n, int b[], int m, int c[]);
```
- gde je  $a$  niz koeficijenata prvog polinoma,  $n$  je stepen prvog polinoma,  $b$  je niz koeficijenata drugog polinoma,  $m$  je stepen drugog polinoma,  $c$  je rezultujući niz koeficijenata, i funkcija vraća veličinu niza  $c$ .
6. Formirati datoteke `polinom.h`, `polinom.c` i `glavni.c`, gde ce u `polinom.h` biti prototipi funkcija vezanih za polinome, u `polinom.c` će se "uvući" sa `#include` direktivom `polinom.h` i dati definicije ovih funkcija, a `glavni.c` ce biti primer "glavnog" programa koji koristi modul `polinom.c`.

Napomena:

Kompilacija može onda da ide na više načina:

- I način

```
gcc glavni.c polinom.c -o glavni
```

Ovaj način može biti loš ako ima mnogo `.c` fajlova, a samo jedan se promeni, posto se onda vrši ponovo kompilacija svega.

- II način, preko `.o` fajlova

```
gcc -c glavni.c (proizvodi glavni.o)
gcc -c polinom.c (proizvodi polinom.o)
gcc glavni.o polinom.o -o glavni (linkuje glavni.o i polinom.o)
```

Ovo je bolji način, pošto se samo linkuje, tj. ponovo se kompajlira samo ono što je promenjeno, a linkuje se sa ostatkom, pa je skupa operacija kompilacije izbegnuta za većinu fajlova.

7. Dodati novu funkciju u `polinom.c` i `polinom.h`, gde se polinom množi skalarom
- ```
void mnoz_skalarom(int a[], int n, int c);
```
8. Dodati novu funkciju u `polinom.c` i `polinom.h`, koja računa vrednost polinoma u tački  $x$  (koristiti Hornerovu šemu):
- ```
int vr_poly(int a[], int n, int x);
```
9. Dodati novu funkciju u `polinom.c` i `polinom.h` koja množi dva polinoma:
- ```
int mul_poly(int a[], int n, int b[], int m, int c[])
```
- a funkcija vraća dimenziju niza  $c$ .