

Programiranje i programsko inženjerstvo

Predavanja
2014. / 2015.

8. Funkcije - Dodatak

Zašto se ulazni argumenti i rezultati
funkcija **ne smiju** prenositi preko
eksternih ili statičkih varijabli

Funkcija `pow` (i ostale funkcije) iz `math.h` ispravno se realiziraju otprilike ovako

```
/* deklaracija funkcije pow */  
double pow (double x, double y);
```

math.h

```
/* definicija funkcije pow */  
double pow (double x, double y) {  
    /* programski kod za izracunavanje "x na y" */  
    ...  
    return konacniRezultat;  
}
```

math.c

Kako pomoću ispravne funkcije `pow` izračunati $x^{(y^z)}$

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main (void) {
```

```
    double x, y, z;
```

```
    scanf("%lf %lf %lf", &x, &y, &z);
```

```
    printf("%f na (%f na %f) = %f\n", x, y, z,  
           pow(x, pow(y, z)));
```

```
    return 0;
```

```
}
```

glavni.c

Potpuno neprihvatljiv način realizacije funkcije pow iz math.h

math.h

```
/* deklaracije eksternih varijabli potrebnih za prijenos  
   argumenata i rezultata */  
extern double powResult, xPow, yPow;  
  
/* deklaracija funkcije pow */  
void pow (void);
```

math.c

```
/* definicije eksternih varijabli potrebnih za prijenos  
   argumenata i rezultata */  
double powResult, xPow, yPow;  
  
/* definicija funkcije pow */  
void pow (void) {  
    /* programski kod za izracunavanje "xPow na yPow" */  
    ...  
    powResult = ...; /* konacni rezultat spremi u powResult */  
    return;  
}
```

Kako pomoću takve funkcije izračunati $x^{(y^z)}$

```
#include <stdio.h>
#include <math.h>

int main (void) {
    double x, y, z;
    scanf("%lf %lf %lf", &x, &y, &z);
    xPow = y;
    yPow = z;
    pow();
    yPow = powResult;
    xPow = x;
    pow();
    printf("%f na (%f na %f) = %f\n", x, y, z, powResult);
    return 0;
}
```

glavni.c

Mora se paziti čak i na to da se niti jedna lokalna ili statička varijable ne nazove xPow, yPow ili powResult, jer su ta imena "rezervirana" isključivo za prijenos argumenata i rezultata funkcije pow.

- upotreba statičkih umjesto eksternih varijabli dovela bi do još većih teškoća. Uz sve navedeno, funkciju *pow* bi trebalo definirati u svakom modulu u kojem se koristi i pri tom ju u svakom modulu drugačije nazvati jer imena funkcija na razini cijelog programa moraju biti jedinstvena.

Zašto se podatak o dimenziji polja u funkciju **ne smije** "prenijeti" preko simboličke konstante

Napisati funkciju za ispis cjelobrojne matrice na zaslon - ispravno rješenje

matr.h

```
void ispisMatrice(int *mat, int m, int n, int maxStup);
```

matr.c

```
#include <stdio.h>
#include "matr.h"

void ispisMatrice (int *mat, int m, int n, int maxStup) {
    int i, j;
    for (i = 0; i < m; ++i) {
        for (j = 0; j < n; ++j)
            printf("%6d", *(mat + i*maxStup + j));
        printf("\n");
    }
    return;
}
```


Kako koristiti funkciju `ispisMatrice`

glavni1.c

```
#include <stdio.h>
#include "matr.h"
#define MAXBRRED 10
#define MAXBRSTUP 20
int main (void) {
    int nekaMatrica[MAXBRRED][MAXBRSTUP];
    int brred, brstup;
    scanf("%d %d", &brred, &brstup);
    /* ovdje treba procitati elemente matrice */
    ...
    ispisMatrice(&nekaMatrica[0][0],
                 brred,
                 brstup,
                 MAXBRSTUP);

    return 0;
}
```

Istu funkciju moći će koristiti i neki drugi glavni program

glavni2.c

```
#include <stdio.h>
#include "matr.h"
#define NAJVICE_REDAKA 512
#define NAJVICE_STUPACA 853
int main (void) {
    int nekaDrugaMatrica[NAJVICE_REDAKA][NAJVICE_STUPACA];
    int stvarnoRedaka, stvarnoStupaca;
    scanf("%d %d", &stvarnoRedaka, &stvarnoStupaca);
    /* ovdje treba procitati elemente matrice */
    ...
    ispisMatrice(&nekaDrugaMatrica[0][0],
                stvarnoRedaka,
                stvarnoStupaca,
                NAJVICE_STUPACA);

    return 0;
}
```

Potpuno neprihvatljiv način realizacije funkcije za ispis cjelobrojne matrice na zaslon

matr.h

```
void ispisMatrice(int *mat, int m, int n);
```

matr.c

```
#include <stdio.h>
#include "matr.h"
#define MAXBRSTUP 20

void ispisMatrice (int *mat, int m, int n) {
    int i, j;
    for (i = 0; i < m; ++i) {
        for (j = 0; j < n; ++j)
            printf("%6d", *(mat + i*MAXBRSTUP + j));
        printf("\n");
    }
    return;
}
```

Funkcija se može koristiti samo za ispis matrica od točno 20 stupaca

glavni1.c

```
#include <stdio.h>
#include "matr.h"
#define MAXBRRED 10
#define MAXBRSTUP 20
int main (void) {
    int nekaMatrica[MAXBRRED][MAXBRSTUP];
    int brred, brstup;
    scanf("%d %d", &brred, &brstup);
    /* ovdje treba procitati elemente matrice */
    ...
    ispisMatrice(&nekaMatrica[0][0], brred, brstup);
    /* ispis ce u ovom primjeru dobro raditi, ali samo zato
       jer saljemo matricu koja ima točno 20 stupaca */

    return 0;
}
```

Istu funkciju ne može koristiti glavni program koji koristi matricu drugačijih dimenzija

glavni2.c

```
#include <stdio.h>
#include "matr.h"
#define NAJVICE_REDAKA 512
#define NAJVICE_STUPACA 853
int main (void) {
    int nekaDrugaMatrica[NAJVICE_REDAKA][NAJVICE_STUPACA];
    int stvarnoRedaka, stvarnoStupaca;
    scanf("%d %d", &stvarnoRedaka, &stvarnoStupaca);
    /* ovdje treba procitati elemente matrice */
    ...
    ispisMatrice(&nekaDrugaMatrica[0][0],
                stvarnoRedaka,
                stvarnoStupaca);
    /* ispis nece dobro raditi jer se u funkciji
       pretpostavlja da matrica ima 20 stupaca */
    ...
    return 0;
}
```