**Računalstvo III1 1 i 7h 30.03**

 **Formalni parametri funkcije**

Prema samoj definiciji funkcije, njihova je namjena različita. Neke ne zahtijevaju ulazne podatke, pa **lista\_argumenata** može ostati i prazna (kao kod navedenih funkcija *proba i zamjena*). Ako ulazni podaci postoje, za svaki od njih navodi se **tip podatka** i **indefikator.** Međusobno se odvajaju zarezom (kao kod funkcije proba1). U funkciji proba1 idenfikator x je formalni parametar koji služi za rezerviranje memorijskog prostora. Tek pozivom funkcije ne mjesto formalnog parametra, učitava se u memoriju stvarna vrijednost navedena u pozivu.

Kao što je navedeno na početku, argument funkcije se, osim unutar zagrada, može deklarirati i ovako:

int proba2(x)

int x;

{

 Return(x\*x);

}

Iako je češće u upotrebi deklaracija kod koje se i tip i ime svakog formalnog parametra navode unutar zagrade.

**Poziv funkcije**

Funkcija se izvršava onog trenutka kad je pozvana iz glavnog programa ili neke druge funkcije

Za poziv funkcije dovoljno je navesti njezino ime i listu stvarnih arumenata onih redoslijedom koja su navedeni u najavi funkcije: ime\_funkcije (stvarna lista\_argumenata)

Lista argumenata može biti i prazna. Ako je lista prazna, onda funkcija rabi globalne varijable (kao u primjeru funkcija zamjena). No obvezatno je pisati zagrade(), jer one označavaju da se kontrola tijeka programa prenosi iz nadređene funkcije.

**Funkcije se mogu pozivati u izrazima, petljama ili kao argumenti u pozivu drugih funkcija.**

**Deklaracija funkcije**

Svaka funkcija se prije poziva treba najaviti.

Najava omogućuje pravilnu pretvorbu različitih tipova podataka te provjeru tipova podataka u listi argumenata i podataka iz glavnog programa koji se prenose u funkciju.

Ako se funkcija piše na kraju glavnog programa, najavljuje se navođenjem **tipa, imena i tipa podataka ulaznih argumenata (prototip funkcije),** kao što je to učinjeno u sljedećem primjeru **funkcije za računanje udaljenosti između dvije točke u pravokutnom koordinatnom sustavu:**

#include <stdio.h>

#include<math.h>

/\*najava funkcije za računanje udaljenosti\*/

float udaljenost(float, float, float, float); /\*prototip funkcije\*/

/\*početak glavnog programa\*/

void main()

{

 float x1,x2,y1,y2,d1;

 printf(„Upišite koordinatne točke T1“);

 scanf(„%f,%f“,&x1,&y1);

 printf(„Upišite koordinatne točke T2“);

 scanf(„%f,%f“,&x2,&y2);

 /\*računanje udaljenosti medju točkama T1 i T2 pozivom fukcije\*/

 d1=udaljenost(x1,y1,x2,y2);

 printf(„\n Udaljenost medju zadanim točkama je %.2f\n“,d1);

 /\*kraj glavnog programa dijela programa\*/

}

/\*tijelo funkcije za račuananje udaljenosti.\*/

 float udaljenost(float a, float b, float c, float d)

{

 float dulj;

 dulj=sqrt(pow(c-a,2)+pow(d-b,2));

 return dulj;

}

Tesni primjer:

 0,0

 3,0

Udaljenost među zadanim točkama je 3.00

Međutim, u C-u je uobičajeno pisati funkciju na početku, nakon svih pretprocesorskih naredbi i najave globalnih varijabli, ali prije funkcije **main()**. Tada je ne treba posebno najavljivati, jer sama funkcija ujedno i najava.

**Primjer:**

#include <stdio.h>

#include<math.h>

/\*funkcije za računanje udaljenosti\*/

float udaljenost(float, float, float, float); /\*prototip funkcije\*/

 float udaljenost(float a, float b, float c, float d)

{

 float dulj;

 dulj=sqrt(pow(c-a,2)+pow(d-b,2));

 return dulj;

}

 /\*početak glavnog programa\*/

void main()

{

 float x1,x2,y1,y2,d1;

 printf(„Upišite koordinatne točke T1“);

 scanf(„%f,%f“,&x1,&y1);

 printf(„Upišite koordinatne točke T2“);

 scanf(„%f,%f“,&x2,&y2);

 /\*računanje udaljenosti medju točkama T1 i T2 pozivom fukcije\*/

 d1=udaljenost(x1,y1,x2,y2);

 printf(„\n Udaljenost medju zadanim točkama je %.2f\n“,d1);

 /\*kraj glavnog programa dijela programa\*/

}

Osim navedenih načina, definicija funkcija može biti i odvojenoj datoteci ili unutar neke druge fukcije. No takve se najave rjeđe rabe. U tim slučajevima obveznatno je najaviti funkciju.

Funkcija se može pojaviti kao argument druge funkcije i broj ugniježdenih funkcija nije ograničen.

**Primjer:**

Nakon izvođenja programa:

#include <stdio.h>

void funkcija\_3()

{

printf(„\n Funkcija\_3“);

}

void funkcija\_2()

{

printf(„\n Funkcija\_2“);

funkcija\_3();

printf(„\n Opet funkcija\_2“);

}

void funkcija\_1()

{

printf(„\n Funkcija\_1“);

funkcija\_2();

printf(„\n Opet funkcija\_1“);

}

void main()

{

printf(„Glavni program“);

funkcija\_1();

printf(„\nOpet glavni program“);

}

**Zapamtite:**

Funkcija ne može biti pozvana ako prije poziva nije najavljena!

Stoga Funkcija\_3 mora biti najavljena prije nego što je pozvana. S obzirom da se poziva u funkciji\_2 mora biti najavljen prije nje. Isto tako funkcija\_2 je najavljena prije funkcije\_1 jer se u njoj poziva!

Ispod objavljene lekcije dužni ste napisati što će napisati ovaj program?

Lekcija je na 84.stranici udžbenika. Za sve ostale nejasnoće obratiti se na e-mail: ivicatolo@hotmail.com

Zadatak:

Napraviti program koji će, koristeći se funkcijom za udaljenost dviju točaka, izračunati i ispisati opseg i površinu trokuta zadanog koordinatama svojih točaka u pravokutnom kordinatnom sustavu.

Kada uradite zadatak uslikajte i postavite ispod lekcije. LP