

Uvod

Alfabet

Alfabet programskog jezika C se sastoji iz malih i velikih slova engleskog alfabeta, cifara dekadnog brojnog sistema i znakova: () + - * / | ~ # ^ % & _ ! = ' " : ; < > , . ?

Konstante

Konstante u C programskom jeziku mogu biti celobrojne konstante (zadate u dekadnom, oktalnom i heksadecimalnom brojnom sistemu), realne ili razlomljene konstante (bez eksponenta ili sa eksponentom) i znakovne konstante (vrednosti znakova koje je moguće učitati ili ispisati na posmatranom sistemu).

Identifikatori

Identifikatori su namenjeni imenovanju: varijabli, funkcija, makroa, struktura, konstanti i drugih objekata. Imena identifikatora moraju biti u skladu sa pravilima imenovanja u programskom jeziku C, ona obuhvataju: slova iz alfabeta C jezika (mala i velika slova se razlikuju), donju crtu, dekadne cifre. Pri čemu prvi znak identifikatora ne sme biti cifra.

Oblast važenja identifikatora

Oblast važenja identifikatora je deo izvornog koda u kojem je on prepoznat, počinje definicijom i važi do kraja izvornog koda, bloka, naredbe ili do kraja funkcije ako je identifikator lokalna za funkciju. Oblasti važenja identifikatora su:

- oblast važenja fajl (file scope),
- oblast važenja blok (block scope),
- oblast važenja prototip (function prototype scope),
- oblast važenja funkcija (function scope).

Oblast važenja se može proširiti.

Tipovi podataka

Tipologija C programskog jezika prepoznaje bazne i izvedene tipove. Bazni tipovi mogu biti osnovni (celobrojni, realni i kompleksni kao i tip void) i nabrojivi tipovi (enumeracija, definiše se službenom rečju enum). Izvedeni tipovi su: pokazivači, nizovi, strukture, unije, polja bita, funkcije. Za tri osnovne familije tipova znakovnih tipova char, celobrojnih tipova int i realnih tipova double i float, važe modifikatori dužine (short, long) i predznaka signed, unsigned. Prirodni tipovi su int za ceo broj double za realni.

Iskaz typedef

Iskaz `typedef` je namenjen za formiranje tipova definisanih od strane programera. Ova deklaracija ne formira novi tip nego uvodi novo ime za već postojeći tip.

Promenljive

Promenljive ili varijable predstavljaju imenovani deo memorije namenjen za čuvanje vrednosti tokom izvršavanja programa. Promenljive se moraju definisati tačno jedanput, definicijom se zadaju tip i ime (identifikator), a kompajler na osnovu tipa rezerviše memorijsku lokaciju. Opseg važenja promenljive počinje njenom definicijom i može se inicijalizovati samo konstantnim izrazom.

Standardni ulaz/izlaz

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Zadatak 1.1

Primer poziva:

Primer rešenja:

```
/* Ispisuje na standardni izlaz Zdravo svete! */
#include<stdio.h>

main()
{
    printf("Zdravo svete!\n");
    return 0;
}
```

Zadatak 1.2

Primer poziva:

Primer rešenja:

```
/* Novi red \n, tab \t, obrnuta kosa crta \\", */
#include<stdio.h>

main()
{
    printf("\nZdravo svete!\tZdravo svete\n");
    printf("\n\\n\\tnovi red\\n\\t\\ttab\\n\\\\\\tobrnuta kosa crta\\n");
    return 0;
}
```

Zadatak 1.3

Primer poziva:

Primer rešenja:

```
// Ispis konstanti na standardni izlaz
```

Zadatak 1.4

Primer poziva:

Primer rešenja:

```
/* Ispis vrednosti promenljive */
#include <stdio.h>

main()
{
    char aa='A';
    short bb=-1234;
    unsigned cc=76543;
    int dd=1;
    float ee=-432.789;
    double ff=123456.76543;
    char str[]="Zdravo svete!";

    printf("Vrednost promenljive aa je %c\n",i);

    return 0;
}
```

Zadatak 1.5

Primer poziva:

Primer rešenja:

```
/* Unos celobrojne vrednosti sa standardnog ulaza, nakon ulazne poruke
i ispis poruke i vrednosti promenljive na standardni izlaz */
#include<stdio.h>

main()
{
    int aa;

    //Tekst namenjen korisniku za identifikaciju unosa
    printf("\nUnesi ceo broj:");

    scanf ("%d",&aa);

    //Tekst izlaza koji kombinuje poruku i vrednost promenljive
    printf("Uneli ste broj %d\n", aa);

    return 0;
}
```

Zadatak 1.6

Primer poziva:

Primer rešenja:

```
/*Unos i ispis realnih brojeva*/
#include<stdio.h>

main()
{
    double xx;
    printf("\nUnesite realan broj (npr: 3456.9876): ");

    scanf("%f",&xx);
    printf("\nUneti broj u decimalnom zapisu: %lf", xx);
    printf("\nUneti broj u eksponencijalnom zapisu: %lg", xx);

    return 0;
}
```

Zadatak 1.7

Primer poziva:

Primer rešenja:

```
#include<stdio.h>

main() {

    char ime[20];
    char prezime[20];

    printf("\nKako se zoves? ");
    scanf("%s", ime);

    printf("\nKako se prezivas? ");
    scanf("%s", prezime);

    printf("\n Tvoje puno ime je: %s %s", ime, prezime);
    return 0;
}
```

Zadatak 1.8

Primer poziva:

Primer rešenja:

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

int main() {
    double a = 2.0;
    double b = 4.0;
    double c = 5.0;

    // poluobim trougla
    double s = (a + b + c) / 2.0;

    // površina trougla, putem heronovog obrasca
    double p = sqrt(s * (s - a) * (s - b) * (s - c));
    printf("p = %lf\n", p);

    return 0;
}
```

Zadatak 1.9

Primer poziva:

Primer rešenja:

```
// metri u fite

#include <stdio.h>

int main() {
    double celzijusi, kelvini;

    printf("Unesite temperaturu u celzijusima: ");
    scanf("%lf", &celzijusi);

    kelvini = celzijusi + 273.15;
    printf("%.2lf stepeni celzijusa je %.2lf stepeni kelvina\n", celzijusi, kelvini);

    return 0;
}
```

Zadatak 1.10

Primer poziva:

Primer rešenja:

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

int main() {
    double a, b, c;

    printf("a = ");
    scanf("%lf", &a);

    printf("b = ");
    scanf("%lf", &b);

    printf("c = ");
    scanf("%lf", &c);

    // poluobim trougla
    double s = (a + b + c) / 2.0;

    // površina trougla, putem heronovog obrasca
    double p = sqrt(s * (s - a) * (s - b) * (s - c));
    printf("p = %lf\n", p);

    return 0;
}
```