

19. Vježbe uz predavanja

U svim zadacima u kojima se traži definiranje funkcije, treba napisati odgovarajući glavni program (tj. funkciju `main`) u kojem ćete po potrebi definirati stvarne argumente, pozvati funkciju i ispisati rezultat.

1. Na vlastitom računalu testirajte sve primjere vezane uz makro definicije s predavanja.
2. Napišite program koji će poslužiti kao primjer kojim ćete "dokazati" da sljedeće macro definicije nisu ispravno napisane:

```
#define ZBROJI(a, b) a + b
#define ODUZMI(a, b) (a)-(b)
#define PODIJELI(x, y) (x/y)
#define OPETPODIJELI(x, y) (x)/(y)
```

Koja pravila pisanja ove macro definicije ne zadovoljavaju? Zatim u svom programu ispravite navedene macro definicije i ponovo izvedite program.

3. Napisati funkciju `traziPrviSamoglas` koja u zadanom nizu znakova pronalazi prvi samoglasnik (malo ili veliko slovo) koji se pojavljuje u nizu. Funkcija u pozivajući program vraća pokazivač na pronađeni samoglasnik, a ako u nizu nema niti jedan samoglasnik, vraća `null` pokazivač.

Napisati glavni program koji će pomoću funkcije `gets` učitati niz znakova ne dulji od 80 znakova, te pomoću funkcije `traziPrviSamoglas` pronaći te na zaslon ispisati prvi samoglasnik učitano niza ili poruku "U nizu nema samoglasnika".

4. Definirati tip podatka `tTocka` kojim se opisuje jedna točka u pravokutnom koordinatnom sustavu (x i y su vrijednosti tipa `double`). Napisati prototip funkcije `udaljToc` koja za dvije točke zadane pomoću tipa podatka `tTocka` izračunava udaljenost među točkama (`double`). Definiciju tipa `tTocka` i prototip funkcije `udaljToc` napisati u zaglavnoj datoteci `geom.h`. Definiciju funkcije `udaljToc` napisati u modulu `geom.c`

Napisati glavni program koji će u varijable `t1` i `t2` tipa `tTocka` učitati koordinate dviju točaka, zatim pomoću funkcije `udaljToc` izračunati njihovu udaljenost, te izračunatu vrijednost ispisati na zaslon.

Rješenja svih zadataka provjerite prevođenjem i testiranjem vlastitih programa!

Rješenja:

Rješenje 2. zadatka

```
#include <stdio.h>

#define ZBROJI(a, b) a + b
#define ODUZMI(a, b) (a)-(b)
#define PODIJELI(x, y) (x/y)
#define OPETPODIJELI(x, y) (x)/(y)

int main() {
    int c = ZBROJI(3, 2) * 5;
    float z = ODUZMI(3.f, 2.f) * 5.f;
    int d = PODIJELI(20, 2*5);
    float w = 20.f / OPETPODIJELI(10.f, 2.f);

    printf("ZBROJI(3, 2) * 5 = 25?           dobije se: %d\n", c);
    printf("ODUZMI(3.f, 2.f) * 5.f = 5.0?     dobije se: %f\n", z);
    printf("PODIJELI(20, 2*5) = 2?           dobije se: %d\n", d);
    printf("20.f / OPETPODIJELI(10.f, 2.f) = 4.0? dobije se: %f\n", w);

    return 0;
}
```

Ispravne macro definicije:

```
#define ZBROJI(a, b) ((a) + (b))
#define ODUZMI(a, b) ((a)-(b))
#define PODIJELI(x, y) ((x)/(y))
#define OPETPODIJELI(x, y) ((x)/(y))
```

Rješenje 3. zadatka

```
#include <stdio.h>

char *traziPrviSamoglas(char *niz) {
    char *samoglas = NULL;
    while (*niz) {
        if (*niz == 'a' || *niz == 'A' ||
            *niz == 'e' || *niz == 'E' ||
            *niz == 'i' || *niz == 'I' ||
            *niz == 'o' || *niz == 'O' ||
            *niz == 'u' || *niz == 'U') {
            samoglas = niz;
            break;
        }
        niz++;
    }
    return samoglas;
}
```

```
int main () {
    char niz[80+1];
    char *samoglas;
    gets(niz);
    samoglas = traziPrviSamoglas(niz);
    if (samoglas != NULL) {
        printf("Prvi samoglasnik u nizu \"%s\" je znak '%c'\n", niz, *samoglas);
    }
    else {
        printf("U nizu \"%s\" nema samoglasnika\n", niz);
    }
}
```

Funkcija se može vrlo lako prepraviti tako da pronalazi posljednji samoglasnik u nizu.

Rješenje 4. zadatka

glavni.c

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

int main() {
    tTocka t1, t2;

    printf("Upisite koordinate tocke t1:");
    scanf("%lf%lf", &t1.x, &t1.y);
    printf("Upisite koordinate tocke t2:");
    scanf("%lf%lf", &t2.x, &t2.y);

    printf("Udaljenost izmedju t1 i t2 je %f\n", udaljToc(t1, t2));
    return 0;
}
```

geom.h

```
typedef struct {
    double x;
    double y;
} tTocka;

double udaljToc (tTocka t1, tTocka t2);
```

geom.c

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

double udaljToc (tTocka t1, tTocka t2) {
    return sqrt(pow(t2.x - t1.x, 2.) + pow(t2.y - t1.y, 2.));
}
```