

17. Dodatne vježbe

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. S tipkovnice učitati cijeli broj n uz kontrolu da je $3 \leq n \leq 20$. Na slučajan način odabrati n velikih slova (među znakovima A-Z) i ispisati ih na zaslon.
4. Generirati 10 000 cijelih brojeva iz zatvorenog intervala [50,60]. Na zaslon ispisati frekvenciju pojavljivanja svakog od brojeva.

```
50 se pojavio 1100 puta
51 se pojavio 987 puta
...
60 se pojavio 1083 puta
```

5. Napisati funkciju koja u zadano dvodimenzionalno cjelobrojno polje "ispaljuje zadani broj hitaca". Element polja u kojeg pojedini hitac pogađa odabire se na slučajan način (funkcija na slučajan način odabere redak elementa, a zatim na slučajan način odabere stupac elementa). Vrijednost elementa polja koji je "pogođen", funkcija uvećava se za jedan. Funkcija treba inicijalizirati generator pseudoslučajnih brojeva samo pri prvom pozivu, a prije nego počne "gađati" polje, mora elemente polja inicijalizirati na vrijednost 0.

U glavnom programu definirati polje maksimalnih dimenzija 10x10, s tipkovnice učitati m i n , te broj "hitaca" koje u polje dimenzija $m \times n$ funkcija treba "ispaliti". Pozvati funkciju i ispisati sadržaj "izrešetanog" polja u obliku tablice, a zatim u nastavku programa to ponoviti (s istim argumentima).

Što će se dogoditi ako funkciju prepravite tako da se generator pseudoslučajnih brojeva inicijalizira **pri svakom** pozivu funkcije? Zašto se sada (u najvećem broju slučajeva) dobiju dva jednaka polja?

Rješenja svih zadataka provjerite prevođenjem i testiranjem vlastitih programa!

Rješenja: NE GLEDATI prije nego sami pokušate riješiti zadatke

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 PODIJELI_B(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 / PODIJELI_B(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 / PODIJELI_B(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 PODIJELI_B(x, y) ((x)/(y))
```

Rješenje 3. zadatka

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

int main () {
    int n, i, slucajni;
    char ascii;
    do {
        printf("Upisi n:");
        scanf("%d", &n);
    }
    while (n < 3 || n > 20);

    srand ((unsigned) time(NULL));

    for (i = 0; i < n; i++) {
        slucajni = rand();
        /* interval [a=0, b=RAND_MAX] preslikati u interval [c='A', d='Z'] */
        ascii = (float) slucajni / (RAND_MAX+1) * ('Z'-'A'+1) + 'A';
        /* moze i ovako: ascii = slucajni % ('Z' - 'A' + 1) + 'A'; */
        printf("%c", ascii);
    }
    printf("\n");
}
```

Rješenje 4. zadatka

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

#define DG 50
#define GG 60

int main () {
    int i, slucajni, slucajni2;
    int polje[GG-DG+1] = {0};

    srand ((unsigned) time(NULL));
    for (i = 0; i < 1000; i++) {
        slucajni = rand();
        /* interval [a=0, b=RAND_MAX] preslikati u interval [c=DG, d=GG] */
        slucajni2 = (float) slucajni / (RAND_MAX+1) * (GG-DG+1) + DG;
        /* moze i ovako: slucajni2 = slucajni % (GG - DG + 1) + DG; */
        polje[slucajni2-DG]++;
    }
    for (i = DG; i <= GG; i++) {
        printf("%d se pojavio %d puta\n", i, polje[i-DG]);
    }
}
```

Rješenje 5. zadatka

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

#define MAXR 10
#define MAXS 10

void gadjaj(int *polje, int m, int n, int maxstup, int brojHitaca) {
    static int generatorInicijaliziran = 0;
    int i, j, redak, stupac;
    if (!generatorInicijaliziran) {
        srand ((unsigned) time(NULL));
        generatorInicijaliziran = 1;
    }

    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            *(polje + i*maxstup + j) = 0;

    for (i = 0; i < brojHitaca; i++) {
        /* slucajni redak */
        redak = rand() % m;
        /* slucajni stupac */
        stupac = rand() % n;
        *(polje + redak * maxstup + stupac) += 1;
    }
}

void ispisiPolje(int *polje, int m, int n, int maxstup) {
    int i, j;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++)
            printf("%5d", *(polje + i*maxstup + j));
        printf("\n");
    }
}

int main () {
    int m, n, brojHitaca;
    int polje[MAXR][MAXS];
    printf("Upisite m, n i broj hitaca -> ");
    scanf("%d %d %d", &m, &n, &brojHitaca);

    gadjaj(&polje[0][0], m, n, MAXS, brojHitaca);
    ispisiPolje(&polje[0][0], m, n, MAXS);

    printf("\n");

    gadjaj(&polje[0][0], m, n, MAXS, brojHitaca);
    ispisiPolje(&polje[0][0], m, n, MAXS);
    return 0;
}
```

Funkcija će pri svakom pozivu ponovo inicijalizirati generator ako se blok naredbi

```
if (!generatorInicijaliziran) {
    srand ((unsigned) time(NULL));
    generatorInicijaliziran = 1;
}
```

zamijeni sa

```
srand ((unsigned) time(NULL));
```

Druga inicijalizacija generatora će se sada dogoditi vrlo vjerojatno s istim sjemenom (jer će se oba poziva funkcije vrlo vjerojatno obaviti unutar iste sekunde), stoga će i nizovi generiranih pseudoslučajnih brojeva biti jednaki.