

## 20. 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. Napisati funkciju `obrniNiz` koja obrće niz znakova. Npr. niz znakova "ABCDE" mijenja u niz znakova "EDCBA". Uputa: 1. znak zamijeniti s n-tim znakom, 2. znak zamijeniti s n-1-vim znakom, itd.
2. Napisati funkciju `umetniZnak` koja na početak zadanog niza znakova umeće zadani znak. Funkcija može računati na to da je za niz u pozivajućem programu rezervirano dovoljno memorije. U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
3. Napisati funkciju `umetniZnakove` koja ispred svakog znaka zadanog niza umeće zadani znak (npr, ako se funkciji zada niz "Studeni" i znak 'X', funkcija mijenja niz u "XSXtXuXdXeXnXi"). Funkcija može računati na to da je za niz u pozivajućem programu rezervirano dovoljno memorije. U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
4. Napisati vlastitu funkciju `myStrcpy` koja obavlja isto što i funkcija `strcpy` definirana u `string.h`. U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
5. Napisati vlastitu funkciju `myStrlen` koja obavlja isto što i funkcija `strlen` definirana u `string.h` (jedina razlika neka bude u tome što `myStrlen` vraća `int`, umjesto `size_t`). U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
6. Napisati vlastitu funkciju `myStrcat` koja obavlja isto što i funkcija `strcat` definirana u `string.h`. U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
7. 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.
8. Generirati 10 000 cijelih brojeva iz zatvorenog intervala  $[50,60]$ . Na zaslon ispisati frekvenciju pojavljivanja svakog od brojeva.
9. 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  $10 \times 10$ , 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 1. zadatka

```
#include <stdio.h>
#define MAX 20

void obrniNiz(char *niz);

int main () {
    char niz[MAX+1];
    printf ("Upisi niz znakova (ne dulji od %d znakova):", MAX);
    gets(niz);
    printf("%s\n", niz);
    obrniNiz(niz);
    printf("%s\n", niz);
    /* VAZNO PITANJE: */
    /* zasto funkciju nije moguće pozvati ovako: obrniNiz("Prosinać"); */
    return 0;
}

void obrniNiz(char *niz) {
    int i;
    char pom;
    int duljina = 0;
    while (*(niz + duljina))
        duljina++;
    for (i = 0; i < duljina/2; i++) {
        pom = niz[i];
        niz[i] = niz[duljina-1-i];
        niz[duljina-1-i] = pom;
    }
}
```

### Rješenje 2. zadatka

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

void umetniZnak(char *niz, char c);

int main () {
    char niz[7+1+1];
    strcpy(niz, "Studeni");
    umetniZnak(niz, 'A');
    printf("%s\n", niz);
    /* VAZNO PITANJE: */
    /* zasto funkciju nije moguće pozvati ovako: umetniZnak("Studeni", 'A'); */
    return 0;
}

void umetniZnak(char *niz, char c) {
    int i;
    int duljina = 0;
    while (*(niz + duljina))
        duljina++;
    niz[duljina+1] = '\\0';
    for (i = duljina; i > 0; i--)
        niz[i] = niz[i-1];
    niz[0] = c;
}
```

### Rješenje 3. zadatka

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

void umetniZnakove(char *niz, char c);

int main () {
    char niz[7+1+7];
    strcpy(niz, "Studeni");
    umetniZnakove(niz, 'X');
    printf("%s\n", niz);
    /* VAZNO PITANJE: */
    /* zasto funkciju nije moguće pozvati ovako: umetniZnakove("Studeni", 'A'); */
    return 0;
}

void umetniZnakove(char *niz, char c) {
    int i;
    int duljina = 0;
    while (*(niz + duljina))
        duljina++;
    *(niz + 2*duljina) = '\\0';
    for (i = duljina - 1; i >= 0; i--) {
        *(niz + 2*i + 1) = *(niz + i);
        *(niz + 2*i) = c;
    }
}
```

### Rješenje 4. zadatka

```
#include <stdio.h>

char *myStrcpy(char *cilj, char *izvor);

int main () {
    char ciljniNiz[20+1];
    char *izvorniNiz = "Niz znakova";
    myStrcpy(ciljniNiz, izvorniNiz);
    printf("%s\n", ciljniNiz);
    myStrcpy(ciljniNiz, "Ana");
    printf("%s\n", ciljniNiz);
    /* moze i ovako: objasnite zasto! */
    printf("%s\n", myStrcpy(ciljniNiz, "Iva"));
    /* VAZNO PITANJE: */
    /* zasto funkciju nije moguće pozvati ovako: myStrcpy("        ", "Ana"); */
    return 0;
}

char *myStrcpy(char *cilj, char *izvor) {
    int i = 0;
    while (*(izvor+i)) {
        *(cilj+i) = *(izvor+i);
        i++;
    }
    *(cilj+i) = '\\0';
    return cilj;
}
```

## Rješenje 5. zadatka

```
#include <stdio.h>

int myStrlen(char *niz);

int main () {
    char niz[20+1] = "Ovo je niz";
    char *konst = "Ovo je konstantni niz";
    printf("%d\n", myStrlen(niz));
    printf("%d\n", myStrlen(konst));
    printf("%d\n", myStrlen("Ovo je jos jedan konstatni niz"));
    return 0;
}

int myStrlen(char niz[]) {
    int i = 0;
    while (niz[i]) i++;
    return i;
}
```

## Rješenje 6. zadatka

```
#include <stdio.h>

char *myStrcat(char *cilj, char *izvor);

int main () {
    char ciljniNiz[30+1] = "Prvi";
    char ciljniNiz2[30+1] = "Pocetak";
    char *izvorniNiz = "Drugi";
    myStrcat(ciljniNiz, izvorniNiz);
    printf("%s\n", ciljniNiz);
    myStrcat(ciljniNiz, "Treci");
    printf("%s\n", ciljniNiz);
    /* moze i ovako: objasnite zasto! */
    printf("%s\n", myStrcat(ciljniNiz, "Cetvrti"));
    /* VAZNO PITANJE: */
    /* zasto funkciju nije moguće pozvati ovako: myStrcat("ABC", "EFG") */
    /* proucite sljedecu naredbu */
    myStrcat(myStrcat(ciljniNiz2, "Sredina"), "Kraj");
    printf("%s\n", ciljniNiz2);
    return 0;
}

char *myStrcat(char cilj[], char *izvor) {
    int i = 0, j = 0;
    while (cilj[i])
        i++;
    while (izvor[j])
        cilj[i++] = izvor[j++];
    cilj[i] = '\0';
    return cilj;
}
```

## Rješenje 7. 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 8. 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 9. 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.