

**Napomene:**

- Savjetuje se navedene zadatke riješiti ubrzo nakon predavanja
- Savjetuje se ne gledati rješenja prije nego se pokuša samostalno riješiti zadatke

## 21. 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 svom računalu testirajte primjere s predavanja vezane uz funkcije `fabs` i `abs`, te primjer kojim je prikazana razlika između funkcije `exit` i naredbe `return`.
2. 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.
3. 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
```

4. Napisati funkciju koja u zadano dvodimenzijsko 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 `h` koje u polje dimenzija `mxn` funkcija treba "ispaliti". Pozvati funkciju te u obliku tablice ispisati sadržaj polja nakon što je "pogođeno" s `h` hitaca, 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 tada (u najvećem broju slučajeva) dobiju dva jednaka polja?

5. 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.
6. 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>`.
7. Napisati funkciju `umetniZnakove` koja ispred svakog znaka zadanog niza umeće zadani znak (npr. ako se funkciji zada niz "Studen" i znak 'X', funkcija mijenja niz u "XSXtXuDxXnXi"). 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>`.
8. 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>`.
9. 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>`.

10. 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>`.
11. Napisati funkciju koja kao rezultat vraća koliko puta se niz `s2` pojavljuje unutar niza `s1`. U glavnom programu pomoću funkcije `gets` učitati dva niza, pozvati funkciju i ispisati rezultat. U funkciji je dopušteno koristiti funkcije iz biblioteke `<string.h>`.
12. Što će biti sadržaj nizova znakova `s1`, `s2` i `s3` nakon obavljanja odsječka programa:  

```
char s1[20] = "ef123f156xyz", s2[20]="ABCD", s3[20];  
strcat(strcat(s2, "abc"), strcpy(s3, strstr(strchr(s1, '2'), "f1")));
```
13. Napisati macro `MYTOUPPER` koja obavlja isto što i funkcija `toupper` definirana u `<ctype.h>`. Napisati glavni program koji će pomoću funkcije `gets` učitati niz znakova (sigurno ne dulji od 40 znakova), te na zaslon ispisati učitani niz, ali tako da su sva mala slova pretvorena u velika. Za pretvaranje malih slova u velika koristiti macro `MYTOUPPER`.
14. Napisati vlastitu funkciju `myStrchr` koja obavlja isto što i funkcija `strchr` definirana u `<string.h>`. U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
15. Napisati vlastitu funkciju `myStrstr` koja obavlja isto što i funkcija `strstr` definirana u `<string.h>`. U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
16. Napisati program kojim će se generirati 1000 slučajnih znakova čije su ASCII vrijednosti u intervalu `[32,126]`. Na zaslon ispisati koliko dobivenih znakova je slovo, koliko dobivenih znakova je znamenka, a koliko dobivenih znakova nije niti znamenka niti slovo. Za ispitivanje znakova koristite ugrađene macro definicije iz `<ctype.h>`.

## Rješenja

### Rješenje 2. zadatka

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

int main (void) {
    int n, i, slucajni;
    char ascii;
    do {
        printf("Upisite 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 3. zadatka

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

#define DG 50
#define GG 60

int main (void) {
    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 4. 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 (void) {
    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 (i neželjeno) 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 tada dogoditi vrlo vjerojatno s istim početnim uvjetima-*seed* (jer će se oba poziva funkcije vrlo vjerojatno obaviti unutar iste sekunde), stoga će i nizovi generiranih pseudoslučajnih brojeva biti jednaki.

## Rješenje 5. zadatka

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

void obrniNiz(char *niz);

int main (void) {
    char niz[MAX+1];
    printf ("Upisite 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 6. zadatka

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

void umetniZnak(char *niz, char c);

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

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

void umetniZnakove(char *niz, char c);

int main (void) {
    char niz[7+1+7];
    strcpy(niz, "Studen");
    umetniZnakove(niz, 'X');
    printf("%s\n", niz);
    /* VAZNO PITANJE: */
    /* zasto funkciju nije moguće pozvati ovako: umetniZnakove("Studen", '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 8. zadatka

```
#include <stdio.h>

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

int main (void) {
    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 9. zadatka

```
#include <stdio.h>

int myStrlen(char *niz);

int main (void) {
    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 10. zadatka

```
#include <stdio.h>

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

int main (void) {
    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") */
    /* proucrite sljedecu naredbu */
    myStrcat(myStrcat(ciljniNiz2, "Sredina"), "Kraj");
    printf("%s\n", ciljniNiz2);
    return 0;
}

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

## Rješenje 11. zadatka

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

int brojPodnizova(char *s1, char *s2) {
    char *pocetakPotrage = s1;
    int brojac = 0;
    while ((pocetakPotrage = strstr(pocetakPotrage, s2)) != NULL) {
        ++brojac;
        ++pocetakPotrage;
    }
    return brojac;
}

int main (void) {
    char s1[40+1];
    char s2[40+1];
    printf("Upisite nizove s1 i s2\n");
    gets(s1);
    gets(s2);
    printf("Niz %s se unutar niza %s pojavljuje %d puta\n",
        s2,
        s1,
        brojPodnizova(s1, s2));
    return 0;
}
```



### Rješenje 13. zadatka

```
#include <stdio.h>
#define MYTOUPPER(c) ((c) >= 'a' && (c) <= 'z' ? \
    (c) - 'a' + 'A' : (c))

int main (void) {
    char niz[40];
    int i = 0;
    gets(niz);
    while (niz[i] != 0) {
        putchar(MYTOUPPER(niz[i])); /* ili printf("%c", MYTOUPPER(niz[i])); */
        ++i;
    }
    return 0;
}
```

### Rješenje 14. zadatka

```
#include <stdio.h>

char *myStrchr(char *niz, char c) {
    while (*niz != '\0') {
        if (*niz == c)
            return niz;
        ++niz;
    }
    return NULL;
}

int main (void) {
    char *nadjen;
    nadjen = myStrchr("Neki niz", 'k');
    if (nadjen != NULL)
        printf("Nasao sam:%c\n", *nadjen);
    else
        printf("Nisam nasao\n");

    nadjen = myStrchr("Neki niz", 'K');
    if (nadjen != NULL)
        printf("Nasao sam:%c\n", *nadjen);
    else
        printf("Nisam nasao\n");

    return 0;
}
```

## Rješenje 15. zadatka

```
#include <stdio.h>

char *myStrstr(char *niz, char *podniz) {
    char *pocetakTraganja = niz;
    while (*pocetakTraganja != '\0') {
        int i = 0;
        while (pocetakTraganja[i] == podniz[i] && podniz[i] != '\0')
            ++i;
        if (podniz[i] == '\0')
            return pocetakTraganja;
        ++pocetakTraganja;
    }
    return NULL;
}

int main (void) {
    char *nadjen;
    nadjen = myStrstr("Neki tekst i drugi tekst", "tek");
    if (nadjen != NULL)
        printf("Nasao sam:%s\n", nadjen);
    else
        printf("Nisam nasao\n");

    nadjen = myStrstr("Neki tekst i drugi tekst", "Tekst");
    if (nadjen != NULL)
        printf("Nasao sam:%s\n", nadjen);
    else
        printf("Nisam nasao\n");

    return 0;
}
```

## Rješenje 16. zadatka

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

int main (void) {
    char slucajniZnak;
    int i, slova = 0, znamenki = 0, ostalih = 0;

    srand ((unsigned) time(NULL));
    for (i = 0; i < 1000; ++i) {
        slucajniZnak = rand() % (126 - 32 + 1) + 32;
        if (isalpha(slucajniZnak))
            ++slova;
        else if (isdigit(slucajniZnak))
            ++znamenki;
        else
            ++ostalih;
    }
    printf("%d slova, %d znamenki, %d ostalih znakova\n",
        slova, znamenki, ostalih);
    return 0;
}
```