

Napomena

U svim zadacima, na svim provjerama znanja i vježbama, vrijedi sljedeće: u rješenjima zadataka je upotreba funkcija iz standardne biblioteke dopuštena u svim slučajevima osim kada je to tekstom zadatka eksplicitno zabranjeno.

1. Napisati i testirati vlastitu funkciju `myStrlen` koja obavlja isto što i funkcija `strlen` definirana u `<string.h>`. Jedina razlika neka bude u tome što `myStrlen` vraća rezultat tipa `int`, umjesto tipa `size_t`. U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
2. Napisati i testirati vlastitu funkciju `myStrcat` koja obavlja isto što i funkcija `strcat` definirana u `<string.h>`. U funkciji se ne smiju koristiti funkcije iz `<string.h>`.
3. Napisati funkciju `brojPodnizova` tipa `int` koja kao rezultat vraća koliko puta se zadani niz znakova `s2` pojavljuje unutar zadanog niza znakova `s1`. U glavnom programu (funkciji `main`) učitati dva niza znakova koji zajedno s eventualno učitanoj oznakom novog retka sigurno neće biti dulji od 60 znakova, iz njih izbaciti eventualno učitanoj oznaku novog retka i zatim pomoću funkcije izračunati i na zaslon ispisati rezultat.

Primjeri izvršavanja programa

```
Upisite 1. niz > Koliko u ananasu ima ana?↵
Upisite 2. niz > ana↵
2. niz se u 1. nizu pojavljuje 3 puta
```

```
Upisite 1. niz > Ima li ovdje koji podniz↵
Upisite 2. niz > Potraga za podnizom↵
2. niz se u 1. nizu pojavljuje 0 puta
```

4. 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.

Primjer izvršavanja programa

```
Slova   : 554↵
Znamenke: 103↵
Ostali  : 343
```

5. Napisati program kojim će se učitati tri niza znakova koji zajedno s eventualno učitanoj oznakom novog retka sigurno neće biti dulji od 60 znakova, iz njih izbaciti eventualno učitanoj oznaku novog retka, te ih ispisati na zaslon leksikografski poredane, svaki su svom retku.

```
Upisite 1. niz > kojim se↵
Upisite 2. niz > na zaslon ispisuje↵
Upisite 3. niz > Napisati program↵
Napisati program↵
kojim se↵
na zaslon ispisuje↵
```

6. Što će se ispisati izvršavanjem sljedećeg programa? Riješiti "na papiru" i rezultat provjeriti izvršavanjem na računalu.

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

int main(void) {
    char s1[20] = "ef123f156xyz";
    char s2[20]="ABCD";
    char s3[20];
    strcat(strcat(s2, "abc"), strcpy(s3, strstr(strchr(s1, '2'), "f1")));
    printf("|%s|\n", s1);
    printf("|%s|\n", s2);
    printf("|%s|", s3);

    return 0;
}
```

7. Deklarirana je struktura

```
struct person_s {
    int code;
    char name[128];
    float salary;
};
```

- Napisati funkciju s prototipom

```
bool capitalizeName(struct person_s *person)
```

koja sve riječi unesene u član **name** promijeni tako da počinju velikim slovom, a ostatak riječi pretvara u mala slova. Ako je **name** prazan niz, funkcija vraća *false*, inače *true*.

Osim slova, u **name** se mogu pojaviti samo još znakovi ' ' (space) i '-' (minus).

Primjeri:

Niz "pero peric" mijenja se u "Pero Peric", a rezultat funkcije je *true*

Niz "ana ANIC ivic" mijenja se u "Ana Anic Ivic", a rezultat funkcije je *true*

Niz "ana ivic-anic" mijenja se u "Ana Ivic-Anic", a rezultat funkcije je *true*

Prazan niz "" ostaje "", a rezultat funkcije je *false*

Napisati glavni program kojim će se

- u varijablu tipa struct person_s učitati vrijednost člana **name**,
- ukloniti eventualno učitani znak za novi red (enter),
- pozivom funkcije capitalizeName promijeniti vrijednost člana **name** varijable tipa struct person_s
- ispisati izmijenjena vrijednost člana **name**, odnosno poruka „Prazan niz.“, ako u član **name** varijable tipa struct person_s nije učitani nijedan znak.

Rješenja:

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

```
int main(void) {  
    double x = -12345.1234567;  
    double rez1 = fabsf(x);  
    double rez2 = fabs(x);  
    printf("%14.7lf\n", rez1);  
    printf("%14.7lf\n", rez2);  
  
    return 0;  
}
```

2. `#include <stdio.h>`

```
int myStrlen(const char *niz) {  
    int i = 0;  
    while (*(niz + i) != '\0') {  
        ++i;  
    }  
  
    return i;  
}  
  
int main(void) {  
    char niz[] = "Ovo je niz";  
    char *konst = "Ovo je konstantni niz";  
    printf("%d\n", myStrlen(niz));  
    printf("%d\n", myStrlen(konst));  
  
    return 0;  
}
```

3. #include <stdio.h>

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

int main(void) {
    char ciljniNiz[30+1] = "Prvi";
    char *izvorniNiz = "Drugi";

    myStrcat(ciljniNiz, izvorniNiz);
    printf("%s\n", ciljniNiz);

    myStrcat(ciljniNiz, "Treci");
    printf("%s\n", ciljniNiz);

    return 0;
}
```

```
4. #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;
}

void izbaciNR(char *niz) {
    while (*niz != '\0') {
        if (*niz == '\n' && *(niz + 1) == '\0') {
            *niz = '\0';
        }
        ++niz;
    }

    return;
}

#define MAXNIZ 60

int main(void) {
    char niz1[MAXNIZ + 1], niz2[MAXNIZ + 1];

    printf("Upisite 1. niz > ");
    fgets(niz1, MAXNIZ + 1, stdin);
    izbaciNR(niz1);

    printf("Upisite 2. niz > ");
    fgets(niz2, MAXNIZ + 1, stdin);
    izbaciNR(niz2);

    printf("2. niz se u 1. nizu pojavljuje %d puta", brojPodnizova(niz1, niz2));

    return 0;
}
```

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

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

    srand((unsigned)time(NULL));
    for (int i = 0; i < 1000; ++i) {
        slucajniZnak = rand() % (126 - 32 + 1) + 32;
        if (isalpha(slucajniZnak))
            ++slova;
        else if (isdigit(slucajniZnak))
            ++znamenki;
        else
            ++ostalih;
    }

    printf("Slova   : %3d\n", slova);
    printf("Znamenke: %3d\n", znamenki);
    printf("Ostali   : %3d", ostalih);

    return 0;
}
```

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

void izbaciNR(char *niz) {
    while (*niz != '\0') {
        if (*niz == '\n' && *(niz + 1) == '\0') {
            *niz = '\0';
        }
        ++niz;
    }

    return;
}

#define MAXNIZ 60

int main(void) {
    char niz1[MAXNIZ + 1], niz2[MAXNIZ + 1], niz3[MAXNIZ + 1];

    printf("Upisite 1. niz > ");
    fgets(niz1, MAXNIZ + 1, stdin);
    izbaciNR(niz1);

    printf("Upisite 2. niz > ");
    fgets(niz2, MAXNIZ + 1, stdin);
    izbaciNR(niz2);

    printf("Upisite 3. niz > ");
    fgets(niz3, MAXNIZ + 1, stdin);
    izbaciNR(niz3);

    char *p1 = niz1, *p2 = niz2, *p3 = niz3, *pom = NULL;
    if (strcmp(p1, p2) > 0) {
        pom = p1;
        p1 = p2;
        p2 = pom;
    }
    if (strcmp(p1, p3) > 0) {
        pom = p1;
        p1 = p3;
        p3 = pom;
    }
    if (strcmp(p2, p3) > 0) {
        pom = p2;
        p2 = p3;
        p3 = pom;
    }
    printf("%s\n%s\n%s", p1, p2, p3);

    return 0;
}
```

7. -

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <ctype.h>
struct person_s {
    int code;
    char name[128];
    float salary;
};

bool capitalizeName(struct person_s *person) {

    if(person->name[0] == '\0')
        return false;
    bool first_letter = true;
    for(int i = 0; person->name[i]; i++) {
        if(first_letter) {
            person->name[i] = toupper(person->name[i]);
            first_letter = false;
        }
        else if(person->name[i-1] == ' ') {
            person->name[i] = toupper(person->name[i]);
            first_letter = false;
        }
        else if(person->name[i-1] == '-') {
            person->name[i] = toupper(person->name[i]);
            first_letter = false;
        }
        else
            person->name[i] = tolower(person->name[i]);
    }
    return true;
}

int main(void) {
    struct person_s s;
    printf("Unesite ime > ");
    fgets(s.name, 128, stdin);
    char *p;
    if ((p = strchr(s.name, '\n'))) *p = '\0';
    if(capitalizeName(&s))
        printf("%s", s.name);
    else
        printf("Prazan niz.");

    return 0;
}
```