

1. Napisati funkciju `izbaciNR` koja zadani niz znakova mijenja tako da iz njega izbaci znak novog retka ('\n') ako se radi o posljednjem znaku u nizu (dakle, ako se znak '\n' nalazi neposredno prije terminatora niza). U glavnom programu (funkciji `main`) učitati niz znakova koji zajedno s eventualno učitanoj oznakom novog retka sigurno neće biti dulji od 20 znakova, pomoću funkcije `izbaciNR` izbaciti iz niza eventualno učitanoj oznaku novog retka, ispisati niz i neposredno iza njega znak uskličnik.

Primjeri izvršavanja programa.

```
Upisite niz > Niz dugacak točno 20.↵
Niz dugacak točno 20!
```

```
Upisite niz > Jedan kraci niz.↵
Jedan kraci niz!
```

```
Upisite niz > ↵
!
```

2. Napisati funkciju `izbaciSamoglasne` koja zadani niz znakova mijenja tako da iz njega izbaci sve samoglasnike. U glavnom programu (funkciji `main`) učitati niz znakova koji zajedno s eventualno učitanoj oznakom novog retka sigurno neće biti dulji od 20 znakova, pomoću funkcije `izbaciNR` izbaciti iz niza eventualno učitanoj oznaku novog retka, pomoću funkcije `izbaciSamoglasne` iz niza izbaciti sve samoglasnike i ispisati tako promijenjeni niz.

Primjer izvršavanja programa.

```
Upisite niz > Sofoklova Antigona.↵
Sfklv ntgn
```

3. Napisati funkciju `traziZadnjiZnak` koja u zadanom nizu pronalazi zadnju pojavu zadanog znaka i vraća pokazivač na taj znak. Ako u zadanom nizu ne postoji zadan znak, funkcija treba vratiti prikladan rezultat na temelju kojeg će se moći prepoznati da zadanog znaka u nizu nema. U glavnom programu (funkciji `main`) učitati niz znakova koji zajedno s eventualno učitanoj oznakom novog retka sigurno neće biti dulji od 20 znakova, učitati znak, pomoću funkcije `traziZadnjiZnak` pronaći znak i na zaslon ispisati koliko je pronađeni znak udaljen od početka niza (izraženo u broju bajtova) ili ispisati poruku "U nizu nema zadanog znaka".

Primjeri izvršavanja programa.

```
Upisite niz > Sofoklova Antigona.↵
Upisite znak > o.↵
15
```

```
Upisite niz > Sofoklova Antigona.↵
Upisite znak > B.↵
U nizu nema zadanog znaka
```

4. Napisati funkciju `umetniZnak` koja u zadani niz znakova neposredno ispred zadane pozicije ubacuje zadani znak. Pozicija znaka odgovara indeksu znaka. Funkcija vraća logičku vrijednost *istina* ako je pozicija bila ispravno zadana, inače vraća *laž*. U glavnom programu (funkciji `main`) učitati niz znakova koji zajedno s eventualno učitanim oznakom novog retka sigurno neće biti dulji od 20 znakova, poziciju ispred koje treba umetnuti znak i znak. Pomoću funkcije `izbaciNR` izbaciti iz niza eventualno učitane oznake novog retka, pomoću funkcije `umetniZnak` umetnuti znak ispred zadane pozicije. Ako je znak uspješno umetnut, ispisati novi sadržaj niza, inače, ispisati poruku "Neispravna pozicija".

Primjeri izvršavanja programa.

```
Upisite niz > Sofoklova Antigona↵
Upisite znak > W↵
Upisite poziciju > 2↵
Sofoklova Antigona↵
```

```
Upisite niz > ↵
Upisite znak > W↵
Upisite poziciju > 0↵
W↵
```

```
Upisite niz > Sofoklova Antigona↵
Upisite znak > W↵
Upisite poziciju > 19↵
Neispravna pozicija
```

5. Napisati funkciju `traziPrviSamoglas` koja vraća pokazivač na prvi samoglasnik u zadanom nizu. Ako u zadanom nizu ne postoji niti jedan samoglasnik, funkcija treba vratiti prikladan rezultat na temelju kojeg će se moći prepoznati da samoglasnika u nizu nema. U glavnom programu (funkciji `main`) učitati niz znakova koji zajedno s eventualno učitanim oznakom novog retka sigurno neće biti dulji od 20 znakova, pomoću funkcije `traziPrviSamoglas` pronaći i ispisati na zaslon samoglasnik ili poruku "U nizu nema samoglasnika".

Primjeri izvršavanja programa.

```
Upisite niz > Sofoklova Antigona↵
o
```

```
Upisite niz > Nvmbrr 12th↵
U nizu nema samoglasnika
```

6. Napisati glavni program (funkciji `main`) kojim će se učitati niz znakova koji zajedno s eventualno učitanoj oznakom novog retka sigurno neće biti dulji od 20 znakova. Uzastopnim pozivanjem funkcije iz prethodnog zadatka, `traziPrviSamoglas`, na zaslon ispisati sve samoglasnike pronađene u nizu. Uputa: funkciju pozivati uzastopno, ali ne tako da samoglasnik traži uvijek od samog početka niza.

Primjeri izvršavanja programa.

```
Upisite niz > Sofoklova Antigona↵
Svi samoglasnici: oooaAioa
```

```
Upisite niz > Nvmbr 12th↵
Svi samoglasnici:
```

7. Napisati funkciju `brojiVelikaMala` koja vraća broj velikih i broj malih slova u zadanom nizu. U glavnom programu (funkciji `main`) učitati niz znakova koji zajedno s eventualno učitanoj oznakom novog retka sigurno neće biti dulji od 20 znakova, pomoću funkcije izračunati, a zatim na zaslon ispisati dobivene rezultate.

Primjer izvršavanja programa.

```
Upisite niz > Sofoklova Antigona↵
Velikih: 2
Malih: 15
```

8. Napisati funkciju `stvariObrnutiNiz` koja za zadani niz `niz1` stvara novi niz znakova `niz2` u kojem su znakovi iz niza `niz1` upisani obrnutim redoslijedom. U glavnom programu (funkciji `main`) učitati niz znakova koji zajedno s eventualno učitanoj oznakom novog retka sigurno neće biti dulji od 20 znakova, pomoću funkcije `izbaciNR` izbaciti iz tog niza eventualno učitanoj oznaku novog retka, pomoću funkcije `stvariObrnutiNiz` načiniti novi niz, a zatim ispisati učitani i "obrnuti" niz.

Primjer izvršavanja programa.

```
Upisite niz > Sofoklova Antigona↵
Originalni: Sofoklova Antigona
Obrnuti    : anogitnA avolkofoS
```

9. Definirati tip podatka `tTocka` kojim se opisuje jedna točka u pravokutnom koordinatnom sustavu (x i y su vrijednosti tipa `double`). Napisati funkciju `uda1jToc` koja kao parametre prima dva podatka tipa `tTocka` i izračunava udaljenost među tim točkama (rezultat je tipa `double`). Napisati glavni program koji će u varijable `t1` i `t2` tipa `tTocka` učitati koordinate dviju točaka, zatim pomoću funkcije `uda1jToc` izračunati njihovu udaljenost, te izračunatu vrijednost ispisati na zaslon.

Primjer izvršavanja programa.

```
Upisite koordinate 1. tocke > 1 2↵
Upisite koordinate 2. tocke > 3 4↵
2.828427
```

10. Jednako kao prethodni zadatak, ali funkcija `uda1jToc` kao parametre prima pokazivače na dva podatka tipa `tTocka`. U kojoj od verzija funkcije će biti korišten manji prostor na stogu i zašto?

11. Deklarirana je struktura

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

Napisati funkciju s prototipom

```
void delCharFromName(struct person_s *s, char c);
```

koja iz polja **name** u strukturi na koju pokazuje **s** izbacuje sve pojave znaka **c**.

Primjeri izvršavanja:

Niz "mali Perica" uz znak 'a' mijenja se u "mli Peric".

Niz "ccc" uz znak 'c' mijenja se u "" (prazan niz).

Niz "FER" uz znak 'e' ostaje nepromijenjen.

Napisati glavni program kojim će se

- u varijablu tipa `struct person_s` učitati vrijednost člana **name**,
- u varijablu tipa `char` učitati jedan znak,
- pozivom funkcije `delCharFromName` izbaciti sva pojavljivanja znaka **c** iz člana **name** varijable tipa `struct person_s`
- ispisati izmijenjena vrijednost člana **name**.

Rješenja:

```

1. #include <stdio.h>
   #define MAX_NIZ 20

   // funkcija izbacuje \n neposredno prije terminatora
   void izbaciNR(char *niz) {
       while (*niz != '\0') {
           if (*niz == '\n' && *(niz + 1) == '\0') {
               *niz = '\0';
           }
           ++niz;
       }
       return;
   }

   int main(void) {
       char niz[MAX_NIZ + 1];

       printf("Upisite niz > ");
       fgets(niz, MAX_NIZ + 1, stdin);

       izbaciNR(niz);
       printf("%s!", niz);

       return 0;
   }

2. #include <stdio.h>
   #define MAX_NIZ 20

   // ovdje copy-paste kod za izbaciNR iz 1. zadatka
   void izbaciSamoglase(char *niz) {
       int i = 0, potroseno = 0;
       while (*(niz + i) != '\0') {
           if (*(niz + i) != 'a' && *(niz + i) != 'A' &&
               *(niz + i) != 'e' && *(niz + i) != 'E' &&
               *(niz + i) != 'i' && *(niz + i) != 'I' &&
               *(niz + i) != 'o' && *(niz + i) != 'O' &&
               *(niz + i) != 'u' && *(niz + i) != 'U')
               *(niz + potroseno++) = *(niz + i);
           ++i;
       }
       *(niz + potroseno) = '\0';    // <-- OVO JE VAZNO!
       return;
   }

   int main(void) {
       char niz[MAX_NIZ + 1];
       printf("Upisite niz > ");
       fgets(niz, MAX_NIZ + 1, stdin);
       izbaciNR(niz);
       izbaciSamoglase(niz);
       printf("%s", niz);
       return 0;
   }

```

```
3. #include <stdio.h>
#define MAX_NIZ 20

char *traziZadnjiZnak(char *niz, char z) {
    char *zadnjaPojava = NULL;
    while (*niz != '\0') {
        if (*niz == z) {
            zadnjaPojava = niz;
        }
        ++niz;
    }
    return zadnjaPojava;
}

int main(void) {
    char niz[MAX_NIZ + 1];
    char z;
    char *pokNaZadnjeg = NULL;

    printf("Upisite niz > ");
    fgets(niz, MAX_NIZ + 1, stdin);
    printf("Upisite znak > ");
    scanf("%c", &z);

    pokNaZadnjeg = traziZadnjiZnak(niz, z);
    if (pokNaZadnjeg == NULL) {
        printf("U nizu nema zadanog znaka");
    } else {
        printf("%d", pokNaZadnjeg - niz);
    }

    return 0;
}
```

```
4. #include <stdio.h>
#include <stdbool.h>
#define MAX_NIZ 20

// ovdje copy-paste kod za izbaciNR iz 1. zadatka

bool umetniZnak(char *niz, int poz, char z) {
    int i, duljina = 0;
    bool uspjesno = 0;
    while (*(niz + duljina) != '\0') {
        ++duljina;
    }
    if (poz >= 0 && poz <= duljina) {
        for (i = duljina; i >= poz; --i)
            *(niz + i + 1) = *(niz + i);
        *(niz + poz) = z;
        uspjesno = 1;
    }
    return uspjesno;
}

int main(void) {
    char niz[MAX_NIZ + 1];
    int poz;
    char z;

    printf("Upisite niz > ");
    fgets(niz, MAX_NIZ + 1, stdin);
    printf("Upisite znak > ");
    scanf("%c", &z);
    printf("Upisite poziciju > ");
    scanf("%d", &poz);

    izbaciNR(niz);
    if (umetniZnak(niz, poz, z)) {
        printf("%s", niz);
    } else {
        printf("Neispravna pozicija");
    }

    return 0;
}
```

```
5. #include <stdio.h>
#define MAX_NIZ 20

char *traziPrviSamoglas(char *niz) {
    while (*niz != '\0') {
        if (*niz == 'a' || *niz == 'A' ||
            *niz == 'e' || *niz == 'E' ||
            *niz == 'i' || *niz == 'I' ||
            *niz == 'o' || *niz == 'O' ||
            *niz == 'u' || *niz == 'U') {
            return niz;
        }
        ++niz;
    }
    return NULL;
}

int main(void) {
    char niz[MAX_NIZ + 1];

    printf("Upisite niz > ");
    fgets(niz, MAX_NIZ + 1, stdin);

    char *pokNaPrviSamoglas = traziPrviSamoglas(niz);
    if (pokNaPrviSamoglas == NULL) {
        printf("U nizu nema samoglasnika");
    } else {
        printf("%c", *pokNaPrviSamoglas);
    }
    return 0;
}
```



```
6. #include <stdio.h>
#define MAX_NIZ 20

// ovdje copy-paste kod za funkciju traziPrviSamoglas

int main(void) {
    char niz[MAX_NIZ + 1];

    printf("Upisite niz > ");
    fgets(niz, MAX_NIZ + 1, stdin);

    printf("Svi samoglasnici: ");
    char *traziOdOvogMjestaNadalje = niz;
    char *pokNaPrviSamoglas = NULL;

    do {
        pokNaPrviSamoglas = traziPrviSamoglas(traziOdOvogMjestaNadalje);
        if (pokNaPrviSamoglas != NULL) {
            printf("%c", *pokNaPrviSamoglas);
            traziOdOvogMjestaNadalje = pokNaPrviSamoglas + 1;
        }
    } while (pokNaPrviSamoglas != NULL);

    return 0;
}
```

```
7. #include <stdio.h>
#define MAX_NIZ 20

void brojiVelikaMala(char *niz, int *brV, int *brM) {
    *brV = *brM = 0;
    while (*niz != '\0') {
        if (*niz >= 'A' && *niz <= 'Z') {
            ++*brV;
        } else if (*niz >= 'a' && *niz <= 'z') {
            ++*brM;
        }
        ++niz;
    }
    return;
}

int main(void) {
    char niz[MAX_NIZ + 1];

    printf("Upisite niz > ");
    fgets(niz, MAX_NIZ + 1, stdin);

    int velika, mala;
    brojiVelikaMala(niz, &velika, &mala);

    printf("Velikih: %d\n", velika);
    printf("Malih: %d\n", mala);

    return 0;
}
```

```
8. #include <stdio.h>
#define MAX_NIZ 20

// ovdje copy-paste kod za izbaciNR iz 1. zadatka

void stvoriObrnutiNiz(char *niz, char *obrnutiNiz) {
    int duljina = 0, i = 0;
    while (*(niz + i) != '\0') {
        ++duljina;
        ++i;
    }
    *(obrnutiNiz + duljina) = '\0';
    for (i = 0; i < duljina; ++i) {
        *(obrnutiNiz + duljina - i - 1) = *(niz + i);
    }
    return;
}

int main(void) {
    char niz[MAX_NIZ + 1];
    char obrnutiNiz[MAX_NIZ + 1];
    printf("Upisite niz > ");
    fgets(niz, MAX_NIZ + 1, stdin);
    izbaciNR(niz);

    stvoriObrnutiNiz(niz, obrnutiNiz);
    printf("Originalni: %s\n", niz);
    printf("Obrnuti   : %s", obrnutiNiz);

    return 0;
}

9. #include <stdio.h>
#include <math.h>

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

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

int main(void) {
    tTocka t1, t2;
    printf("Upisite koordinate 1. tocke > ");
    scanf("%lf %lf", &t1.x, &t1.y);
    printf("Upisite koordinate 2. tocke > ");
    scanf("%lf %lf", &t2.x, &t2.y);

    printf("%lf", udaljToc(t1, t2));

    return 0;
}
```

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

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

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

    int main(void) {
        tTocka t1, t2;
        printf("Upisite koordinate 1. tocke > ");
        scanf("%lf %lf", &t1.x, &t1.y);
        printf("Upisite koordinate 2. tocke > ");
        scanf("%lf %lf", &t2.x, &t2.y);

        printf("%lf", udaljToc(&t1, &t2));

        return 0;
    }
```

11.

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

void delCharFromName(struct person_s *p, char c) {
    char *s = p->name;
    for (; *s; s++) {
        while (*s == c) {
            for (char *t = s; *t; t++) {
                *t = *(t + 1);
            }
        }
    }
}

int main(void) {
    struct person_s s;
    char c;
    printf("Unesite niz > ");
    fgets(s.name, 128, stdin);
    printf("Unesite znak > ");
    scanf("%c", &c);

    delCharFromName(&s, c);
    printf("%s", s.name);

    return 0;
}
```