

19. Dodatne vježbe

Na kraju ovog dokumenta možete pronaći rješenja nekih od zadataka. **Preporučam ne gledati rješenja zadataka prije nego ih sami riješite i testirate!**

1. Napisati vlastitu funkciju koja obavlja isto što i funkcija `strchr` definirana u `<string.h>`.
2. Napisati vlastitu funkciju koja obavlja isto što i funkcija `strstr` definirana u `<string.h>`.
3. Generirati 1000 slučajnih znakova čije su ASCII vrijednosti u intervalu [32,127]. 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>`.
4. Na svom računalu testirajte sve primjere s predavanja vezane uz funkcije `getchar` i `gets`.
5. Napisati funkciju `ucitaj` koja ima sljedeći prototip:

```
int ucitaj(char *polje, int brred, int maxstup);
```

Funkcija `ucitaj` u zadano dvodimenzionalno znakovno polje učitava tekst s tipkovnice. Funkcija svaki redak teksta treba učitati u zasebni redak polja. Funkcija prekida učitavanje redaka teksta kad se učitava prazan redak (redak koji ima duljinu 0) ili nakon što se učitava `brred` redaka. Pri učitavanju retka nije potrebno paziti hoće li učitani redak teksta biti dulji od `maxstup`. Funkcija vraća broj nepraznih redaka teksta koje je pročitala. Glavni program treba pozvati funkciju za učitavanje teksta, te ispisati sve učitane retke teksta, ali tako da tekst ispisuje od zadnjeg retka prema prvom. Za čitanje redaka teksta koristiti funkciju `gets` iz `<stdio.h>`.

6. Napisati program kojim će se u dvodimenzionalno znakovno polje učitati tekst s tipkovnice. Svaki redak teksta treba učitati u zasebni redak polja. Oznake kraja retka (znak `'\n'`) ne upisivati u polje, ali svaki redak u polju treba biti terminiran znakom `'\0'`. Učitavanje redaka teksta treba prekinuti kad se učitava oznaka kraja datoteke. Može se računati da niti jedan redak teksta neće biti dulji od 80 znakova, te da neće biti uneseno više od 10 redaka teksta. Za čitanje teksta koristiti funkciju `getchar` iz `<stdio.h>`. Na kraju, ispisati tekst, svaki redak u svom retku na ekranu.
7. Na svom računalu izvedite sljedeći programski odsječak, uz ulazne podatke `1 2 3.0 4.0`

```
int i;  
short j;  
float x;  
double y;  
scanf("%d %d %f %f", &i, &j, &x, &y);  
printf("%d %d %f %f\n", i, j, x, y);
```

Jesu li se `printf` naredbom ispisale ispravne vrijednosti? **Objasnite zašto.** Što treba promijeniti u formatu `scanf` naredbe da bi odsječak radio ispravno?

8. Što će se ispisati sljedećim programskim odsječkom. Izvođenjem programa na svom računalu provjerite svoj odgovor. Ulazni podaci su `11 12 13 1e`

```
int i, j, k, m;  
scanf("%d %o %x %x", &i, &j, &k, &m);  
printf("%d %d %d %d\n", i, j, k, m);
```

5.

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

#define MAXRED 5
#define MAXSTUP 20

int ucitaj(char *polje, int brred, int maxstup);

int main () {
    char polje[MAXRED][MAXSTUP];
    int ucitanoRedaka, i;
    ucitanoRedaka = ucitaj(&polje[0][0], MAXRED, MAXSTUP);

    for (i = ucitanoRedaka - 1; i >= 0; i--)
        /* puts kao argument dobija adresu prvog clana odgovarajuceg retka 2D polja */
        puts(&polje[i][0]);
}

int ucitaj(char *polje, int brred, int maxstup) {
    int brUcitanihRedaka = 0;
    while (1) {
        /* gets kao argument dobija adresu prvog clana odgovarajuceg retka 2D polja */
        gets(polje + brUcitanihRedaka * maxstup + 0);
        if (strlen(polje + brUcitanihRedaka * maxstup + 0) == 0)
            break;
        brUcitanihRedaka++;
        if (brUcitanihRedaka == brred)
            break;
    }
    return brUcitanihRedaka;
}
```

6.

```
#include <stdio.h>

#define MAXRED 10
#define MAXSTUP 80

int main () {
    char polje[MAXRED][MAXSTUP+1];
    int i = 0, j = 0;
    int c;
    int ucitanoRedaka;

    while (1) {
        c = getchar();
        if (c == '\n') {
            /* oznaci kraj niza, povecaj brojac retka, postavi brojac stupaca na nulu */
            polje[i][j] = '\0';
            i++;
            j = 0;
        } else if (c == EOF) {
            /* oznaci kraj niza i prekini citanje */
            polje[i][j] = '\0';
            break;
        } else {
            polje[i][j] = c;
            j++;
        }
    }
    ucitanoRedaka = i;
    for (i = 0; i < ucitanoRedaka; i++)
        /* puts kao argument dobija adresu prvog clana odgovarajuceg retka 2D polja */
        puts(&polje[i][0]);
}
```