

Završni ispit iz Programiranja i programskog inženjerstva
31.1.2005.

Napomene uz sve zadatke:

- nije dopušteno korištenje globalnih varijabli
- nije dopušteno korištenje goto naredbi
- u svakom zadatku navesti neophodne naredbe za uključivanje prototipova funkcija iz standardne biblioteke potprograma

1. Napisati funkciju čiji je prototip:

```
void napraviLozinku(char *lozinka, char *znakovi, int duljinaLozinke);
```

Funkcija treba uzastopnim slučajnim odabirom znakova, koji se nalaze u nizu znakova znakovi, formirati niz znakova lozinka zadane duljine. Dopušteno je više puta odabrati isti znak. Nije potrebno provjeravati je li za niz lozinka rezerviran dovoljno velik prostor u memoriji.

Na primjer, rezultat izvođenja naredbi:

```
napraviLozinku(loz, "ABC1234", 5);  
printf("%s", loz);
```

bi mogao biti:

ABBAB ili A14BC ili 2CCC2 itd.

(700 bodova)

2. Napisati funkciju maxmin koja u zadanom dvodimenzionalnom cjelobrojnom polju pronalazi najveću i najmanju vrijednost članova polja, te u pozivajući program (funkciju) vraća **obje** pronađene vrijednosti. Funkcija mora biti napisana tako da ispravno radi s dvodimenzionalnim cjelobrojnim poljima s bilo kojim brojem redaka i stupaca. Npr:

za polje

1	-1	2
2	3	-2
5	-2	3

 funkcija vraća 5 i -2. Za polje

1	-1
-2	3
8	-4
4	8

 funkcija vraća 8 i -4.

(800 bodova)

3. Postojeća neformatirana (binarna) datoteka "radnici.dat" sadrži zapise o radnicima:

imeRadnika	20+1 znak
prezRadnika	30+1 znak
placa	realan broj standardne preciznosti

Napisati program koji će svakom radniku plaću uvećati za 10% (za svakog radnika treba **izmijeniti** podatak o plaći u datoteci "radnici.dat"). Ukoliko se datoteka ne može otvoriti, na zaslon ispisati poruku "ne mogu otvoriti datoteku" i završiti program. **(800 bodova)**

4. Napisati funkciju koja vraća **logičku** vrijednost *istina* ako je u zadanom znakovnom nizu pronađen zadani podniz, a **logičku** vrijednost *laž* inače. Pored toga, funkcija mora modificirati zadani znakovni niz tako da samo prvo pojavljivanje podniza u zadanom nizu pretvori u velika slova. Za traženje podniza **obavezno** koristiti odgovarajuću funkciju iz programske biblioteke, te pokazivač dobiven obavljanjem te funkcije **obavezno** koristiti pri izmjeni znakova u nizu.

Primjeri: za zadani niz "sferakon je na feru" i zadani podniz "fer", funkcija vraća *istina*, a zadani niz mijenja u "sFERakon je na feru". Za zadani niz "danas nije sferakon" i zadani podniz "Fer", funkcija vraća *laž*, i ne mijenja zadani niz. **(700 bodova)**

1.

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

void napraviLozinku(char *lozinka, char *znakovi, int duljinaLozinke){
    int i, idx;
    int d = strlen(znakovi) - 1;
    /* sa srand() ili bez srand(), priznaje se jednako */
    for(i = 0; i < duljinaLozinke; i++){
        idx = (float)rand() / (RAND_MAX+1) * (d+1);
        /* ili idx = rand() % strlen(znakovi); */
        lozinka[i] = znakovi[idx];
    }
    lozinka[duljinaLozinke] = '\0';
}
```

2.

```
void maxmin(int *polje, int maxred, int m, int n,
            int *maxclan, int *minclan) {
    int i, j;
    *maxclan = *polje;
    *minclan = *polje;
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++) {
            if (*(polje + i*maxred + j) > *maxclan)
                *maxclan = *(polje + i*maxred + j);
            if (*(polje + i*maxred + j) < *minclan)
                *minclan = *(polje + i*maxred + j);
        }
}
```

3.

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

int main () {
    FILE *dat;
    struct {
        char imeRadnika[20+1];
        char prezRadnika[30+1];
        float placa;
    } zapRadnik;

    if ((dat = fopen("radnici.dat", "r+b")) == NULL) {
        printf("Ne mogu otvoriti datoteku\n");
        exit(1);
    }
    /* povecaj placu svima */
    while (fread(&zapRadnik, sizeof(zapRadnik), 1, dat) == 1) {
        zapRadnik.placa *= 1.1f;
        /* vrati se na pocetak zapisa */
        fseek(dat, -1L*sizeof(zapRadnik), SEEK_CUR);
        fwrite(&zapRadnik, sizeof(zapRadnik), 1, dat);

        fseek(dat, 0L, SEEK_CUR);      /* ili fflush(dat) */
    }
    fclose(dat);
    return 0;
}
```

4.

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

int fun(char *niz, char*podniz) {
    int i;
    char *poc;
    if ((poc = strstr(niz, podniz)) == NULL) {
        return 0;
    }
    else {
        for (i = 0; i < strlen(podniz); i++)
            *(poc+i) = toupper(*(poc+i));
        return 1;
    }
}
```