

Završni ispit iz Programiranja i programskog inženjerstva 23. siječnja 2007.

Napomene za sve zadatke:

- Nije dopušteno korištenje goto naredbe, te statičkih i globalnih varijabli.

1. (8 bodova) Napisati vlastitu *macro* definiciju `mojIsUpper` koja kao argument prihvaća znak. Za veliko slovo abecede *macro* vraća logičku vrijednost *istina*, a za sve ostale znakove vraća logičku vrijednost *laž* (**nije dopušteno** korištenje ugrađene *macro* definicije `isupper` iz `ctype.h`).

Napisati **funkciju** koja kao ulazni argument prima niz znakova. Funkcija mijenja zadani niz tako da svako veliko slovo koje pronađe u nizu redom nadopiše na kraj niza. Funkcija u pozivajući program vraća pokazivač na prvi dodani znak ili NULL pokazivač ukoliko nije dodan niti jedan znak. Za ispitivanje je li neki znak veliko slovo, u funkciji obavezno koristiti *macro* `mojIsUpper`.

Primjer: Zadani niz znakova "Prvo I Drugo" funkcija mijenja u niz "Prvo I DrugoPID", te u pozivajući program vraća pokazivač na 13. znak niza

Napisati **glavni program** koji će funkciju pozvati za niz "Prvo I Drugo", ispisati izmijenjeni niz te, uz pomoću pokazivača dobivenog kao rezultat funkcije, ispisati znakove koji su nadopisani na kraj niza.

Izgled ispisa na zaslonu:
Prvo I DrugoPID
PID

2. (8 bodova) Napisati **funkciju** koja unutar zadanog dvodimenzionalnog cjelobrojnog polja definiranog **s proizvoljnim dimenzijama**, generira matricu od zadanih **m** redaka i zadanih **n** stupaca. Elemente prvog retka i prvog stupca funkcija postavlja na **slučajne vrijednosti** iz zatvorenog intervala [5,15], a ostale elemente matrice na nulu. Npr, ako je u pozivajućem programu definirano polje od 20 redaka i 30 stupaca, a za **m** i **n** se funkciji zadaju vrijednosti 3 i 5, funkcija treba unutar definiranog polja generirati matricu kao na slici. Napisati **glavni program** koji će definirati polje dimenzija 20×30 i pomoću funkcije generirati matricu dimenzija 3×5 (jednako kao u navedenom primjeru). Generiranu matricu **nije potrebno** ispisivati.

20 {	15	9	11	5	7	?	?	...
	8	0	0	0	0	?	?	...
	5	0	0	0	0	?	?	...
	?	?	?	?	?	?	?	...
	...							
30								

3. (7 bodova) Svaki zapis postojeće slijedne formatirane (tekstualne) datoteke "polozio.txt" sadrži šifru studenta (cijeli broj od najviše 5 znamenaka), prezime studenta (niz znakova bez praznina duljine najviše 20 znakova) i broj položenih ispita (cijeli broj iz zatvorenog intervala [0, 10]). Napisati **glavni program** koji će za svaki cijeli broj iz intervala [0, 10], na zaslon ispisati broj studenata koji su položili taj broj ispita. U slučaju pogreške kod otvaranja toka podataka (datoteke) prekinuti izvođenje programa. U primjeru na slici prikazan je mogući izgled datoteke (Horvat je položio 2 ispita, Novak je položio 7 ispita, itd.) i mogući izgled ispisa na zaslonu (niti jedan ispit nije položilo 47 studenata, jedan ispit položilo je 63 studenata, itd.)

polozio.txt

```
12362 Horvat 2
162 Novak 7
52351 Kolar 0
1564 Kos 3
...
```

Izgled ispisa na zaslonu:

```
0 47
1 63
2 78
...
8 0
9 12
10 2
```

4. (7 bodova) Svaki zapis postojeće direktne neformatirane (binarne) datoteke "stud.bin" sadrži podatke o jednom studentu: matični broj studenta (int), ime i prezime studenta (niz znakova duljine 40+1) i ukupni broj bodova koje je stekao (float). Redni broj zapisa datoteke "stud.bin" odgovara matičnom broju studenta.

Napisati **glavni program** koji će s tipkovnice učitati jedan matični broj studenta, te ako zapis s učitanim matičnim brojem postoji u datoteci, uvećati broj bodova u tom zapisu datoteke za 20%. Zapis datoteke se smatra "praznim" ukoliko je kao vrijednost matičnog broja u zapisu upisana nula. Nije potrebno provjeravati uspješnost obavljanja operacija nad datotekom (otvaranje, pozicioniranje, čitanje, zatvaranje).

RJEŠENJA:

1. Zadatak

```
#include <stdio.h>
#include <string.h>
#define mojIsUpper(c) ((c) >= 'A' && (c) <= 'Z' ? 1 : 0)

char *fun(char *niz) {
    int duljina, slobodni, i;
    duljina = slobodni = strlen(niz);
    for (i = 0; i < duljina; i++)
        if (mojIsUpper(niz[i]))
            niz[slobodni++] = niz[i];
    if (slobodni > duljina) {
        niz[slobodni] = '\0';
        return &niz[duljina];
    }
    else
        return NULL;
}

int main () {
    char *rez;
    char niz[12+1+3] = "Prvi I Drugi";
    rez = fun(niz);
    printf("%s\n%s", niz, rez);
    return 0;
}
```

2. Zadatak

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

void generiraj(int *polje, int maxstup, int m, int n) {
    int i, j;
    srand((unsigned)time(NULL));
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            if (i == 0 || j == 0)
                *(polje + i* maxstup + j) = rand() % (15-5+1) + 5;
            else
                *(polje + i* maxstup + j) = 0;
}

int main() {
    int polje[20][30];
    generiraj(&polje[0][0], 30, 3, 5);
    return 0;
}
```

3. Zadatak

```
#include <stdio.h>

int main () {
    FILE *du;
    int mbr, brojIspita, i;
    int brojStud[11] = {0};
    char prez[20+1];
    du = fopen("polozio.txt", "r");

    if (du == NULL)
        exit(-1);

    while (fscanf(du, "%d%s%d", &mbr, prez, &brojIspita) == 3)
        brojStud[brojIspita] += 1;

    for (i = 0; i <= 10; i++)
        printf("%d %d\n", i, brojStud[i]);

    fclose(du);
    return 0;
}
```

4. Zadatak

```
#include <stdio.h>

int main () {
    FILE *dat;
    int mbr;
    struct {
        int mbr;
        char imeprez[40+1];
        float brBod;
    } stud;
    dat = fopen ("stud.bin", "r+b");
    scanf ("%d", &mbr);
    fseek(dat, (long)(mbr-1)*sizeof(stud), SEEK_SET);
    fread(&stud, sizeof (stud), 1, dat);
    if (mbr == stud.mbr) { /* ili if (stud.mbr != 0) */
        stud.brBod = stud.brBod * 1.2f;
        fseek(dat, -1L*sizeof(stud), SEEK_CUR);
        fwrite(&stud, sizeof (stud), 1, dat);
    }
    fclose (dat);
    return 0;
}
```

