

JMBAG	
Ime i prezime	

Programiranje i programsko inženjerstvo

Završni ispit

31. siječnja 2013.

Rješenja 1, 2. i 3. zadatka napišite na vlastitim papirima i predajte u košuljici.

Rješenja zadataka 4, 5, 6. i 7. napišite na za to predviđenom mjestu uz zadatke i također predajte u košuljici.

Nije dopušteno korištenje statičkih, eksternih varijabli te naredbe goto.

Zadatak 1. (11 bodova)

Palindrom je niz znakova koji ima isto značenje čitan od početka i od kraja (npr. "Radar" i "Evo sada sove").

- a) Napisati funkciju *palindrom* koja kao parametar prima znakovni niz, a vraća cjelobrojnu vrijednost 1 ako je niz palindrom, odnosno 0 ako nije. Funkcija ne smije razlikovati mala i velika slova, a u obzir treba uzimati samo znamenke i slova iz ASCII tablice (npr. "Eva, can I see bees in a cave?" JEST palindrom!).

Pretpostavite da u ulaznim podacima nema digrafa (npr. Lj ili Nj).

- b) Napisati glavni program u kojem se s tipkovnice učitava jedan niz znakova (ne dulji od 80 znakova), pozove funkciju *palindrom* i nakon toga na zaslon ispiše poruka "Niz jest palindrom" ili "Niz nije palindrom".

Zadatak 2. (15 bodova)

Šifriranje jednokratnim ključem (eng. One time pad) provodi se tako da se svaki znak poruke zamijeni rezultatom XOR operacije (bitovno „isključivo ili“) tog znaka s odgovarajućim znakom u polju ključa.

- a) Napisati funkciju *sifriraj* koja prima **dvije** znakovne matrice jednake veličine, *poruka* i *otp* i šifrira matricu *poruka* i rezultat se pohranjuje natrag u matricu *poruka* ($PORUKA_{ij} = PORUKA_{ij} \oplus OTP_{ij}$).
- b) Poruka i otp su znakovne matrice jednake veličine (no nisu veće od 30x30 znakova), pohranjene u binarne datoteke *otp.bin* i *poruka.bin*. U *otp.bin* se prvo nalaze dva cijela broja (broj redaka i stupaca) i nakon toga odgovarajući broj znakova matrice, dok *poruka.bin* odmah započinje s podacima.

Napisati glavni program u kojem će se učitati obje datoteke u odgovarajuće matrice, pozvati funkcija iz a) dijela zadatka i ispisati šifrirana matrica *poruka* u tabličnom obliku.

Napomena: Nije potrebno provjeravati uspješnost obavljanja operacija nad datotekama.

Zadatak 3. (15 bodova)

U direktnoj binarnoj datoteci *sati.bin* su zapisi o vremenu boravka na radnom mjestu svih zaposlenika neke tvrtke. Svaki zapis sadrži šifru zaposlenika (int), ime (20+1 znak), prezime (30+1 znak) i broj minuta provedenih na radnom mjestu (int). U tekstualnoj datoteci *dnevnik.txt* se nalaze podaci o ulasku i izlasku zaposlenika iz tvrtke za tekući dan. Svaki redak sadrži šifru zaposlenika (cijeli broj od najviše 5 znamenki), razmak, vrijeme ulaska u obliku *sati:minute*, razmak i vrijeme izlaska u obliku *sati:minute* (Npr. „125 09:05 17:05“).

Napisati program koji broju minuta provedenih na radnom mjestu u binarnoj datoteci pribraja broj minuta provedenih na radnom mjestu za tekući dan dobiven na temelju zapisa iz tekstualne datoteke.

Primjerice, ako u binarnoj datoteci postoji zapis (125, Ana, Horvat, 1000), a u tekstualnoj datoteci se nalazi redak "125 09:05 17:05" zapis u binarnoj datoteci nakon ažuriranja mora sadržavati vrijednosti (125, Ana, Horvat, 1480).

Napomena: Nije potrebno provjeravati uspješnost obavljanja operacija nad datotekama.

Zadatak 4. (2 boda)

Što će se ispisati izvođenjem sljedećeg programskog odsječka?

```
#include <stdio.h>

int main() {
    printf("%f", 3/2);
    return 0;
}
```

Zadatak 5. (2 boda)

Što će se ispisati izvođenjem sljedećeg programskog odsječka?

```
#include <stdio.h>
#define MACRO(A,B,C) A/2 - 2*B + C

int main() {
    int d;
    d = MACRO(1+1,2+2,3-3);
    printf("%d", d);
    return 0;
}
```

Zadatak 6. (3 boda)

Nadopuniti programski odsječak tako da funkcija zamijeni prvo pojavljivanje podniza *rijec* u nizu *recenica* s podnizom *zamjena*, ali samo u slučaju ako su podnizovi *rijec* i *zamjena* jednake duljine.

Nije dopušteno koristiti programske petlje.

```
#include <string.h>
...
int funkcija(char *recenica, char *rijec, char *zamjena){
    char *pom;

    if ( _____ ) {
        pom = _____;
        if(pom != NULL) {
            _____;
        }
    }
    return 0;
}
```

Zadatak 7. (2 boda)

U prostoru označenom kvadratićima napisati vrijednosti pojedinih elemenata polja *niz* nakon izvođenja sljedećeg programskog odsječka.

```
int main() {
    int i, niz[5]={1};
    int *pniz = &niz[1];
    for (i=0; i<3; i++) {
        *pniz = *(pniz+1) + i;
        pniz++;
    }
    return 0;
}
```

Rješenja

Zadatak 1.

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

int palindrom(char *ulaz) {
    int i1, i2;

    i1=0;
    i2 = strlen(ulaz)-1;

    while (i1<i2) {
        if (!isalnum(ulaz[i1])) { i1++; continue; }
        if (!isalnum(ulaz[i2])) { i2--; continue; }
        if (tolower(ulaz[i1]) != tolower(ulaz[i2])) return 0;
        i1++;
        i2--;
    }
    return 1;
}

int main() {
    char test[81];

    printf("\n\nUpisite niz za provjeru: ");
    gets(test);
    printf("Niz \"%s\" %s palindrom!\n", test, palindrom(test)?"JE":"NIJE");

    return 0;
}
```

Zadatok 2.

```
#include <stdio.h>
#define MAX 30

int sifriraj(char *poruka, char *otp, int r, int s, int maxstup){
    int i,j;
    for (i=0; i<r; i++)
        for (j=0; j<s; j++)
            poruka[i*maxstup+j]^=otp[i*maxstup+j];
    return 0;
}

int main() {
    FILE *fotp, *fpor;
    char motp[MAX][MAX], mpor[MAX][MAX];
    int r,s,i,j;
    fotp = fopen("otp.bin", "rb");
    fpor = fopen("poruka.bin", "rb");

    fread(&r,sizeof(int),1,fotp);
    fread(&s,sizeof(int),1,fotp);

    for (i=0; i<r; i++)
        fread(&motp[i], s*sizeof(char), 1, fotp);

    for (i=0; i<r; i++)
        fread(&mpor[i], s*sizeof(char), 1, fpor);

    sifriraj(&mpor[0][0], &motp[0][0], r, s, MAX);

    printf("DESIFRIRANA PORUKA:\n");
    for (i=0; i<r; i++) {
        for (j=0; j<s; j++)
            printf("%c",mat[i][j]);
        printf("\n");
    }

    fclose(fotp);
    fclose(fpor);
    return 0;
}
```

Zadatak 3.

```
#include <stdio.h>

typedef struct {
    int sif;
    char ime[20+1];
    char prezime[30+1];
    int minute;
} zapis;

int main(){
    FILE *fbin, *ftxt;
    zapis zaposlenik;
    int sif, sati_ulaz, minute_ulaz, sati_izlaz, minute_izlaz;

    fbin = fopen("prekovremeni.bin", "r+b");
    ftxt = fopen("log.txt", "r");

    while(fscanf(ftxt, "%5d %2d:%2d %2d:%2d", &sif, &sati_ulaz, &minute_ulaz, &sati_izlaz,
&minute_izlaz) == 5) {

        fseek(fbin, (long) (sif-1) * sizeof(zaposlenik), SEEK_SET);
        fread(&zaposlenik, sizeof(zaposlenik), 1, fbin);

        zaposlenik.minute += (sati_izlaz - sati_ulaz) * 60 + (minute_izlaz - minute_ulaz);

        fseek(fbin, (long) (-1) * sizeof(zaposlenik), SEEK_CUR);
        fwrite(&zaposlenik, sizeof(zaposlenik), 1, fbin);
    }

    fclose(fbin);
    fclose(ftxt);
    return 0;
}
```

Zadatak 4.

0.000000

Priznat će se i rješenje 1.000000

Zadatak 5.

-1

Zadatak 6.

```
#include <string.h>
...
int funkcija(char *recenica, char *rijec, char *zamjena){
    char *pom;
    if (strlen(rijec) == strlen(zamjena)) {
        pom = strstr(recenica, rijec);
        if(pom != NULL) {
            strncpy(pom, zamjena, strlen(zamjena));
        }
    }
    return 0;
}
```

Zadatak 7.

1 0 1 2 0