

Programiranje

vježba za ZI

Vježba za završni ispit

- Završni ispit
 - rješavanje problemskih zadataka, 50 bodova
 - Za pozitivnu ocjenu (prolaz) student treba ostvariti ukupno 50,00 ili više bodova pri čemu na završnom ispitu treba postići barem 17,5 od mogućih 50 bodova (prag prolaznosti od 35%).
- Zadaci:
 - Izbor točnog odgovora ili nadopunjavanje (cca 10-15 bodova)
 - Rješavanje zadanog problema u C-u (cca 35-40 bodova)
 - Polja kao argumenti funkcije
 - Baratanje s nizovima znakova
 - Datoteke

Primjena gradiva iz prethodnih cjelina se podrazumijeva!

Nadoknade završnog ispita u slučaju izostanka **NEMA**

1. zadatak

- Napišite funkciju **maskiraj** koja preko parametara prima matrice **mat** i **maska**. Matrice su istih dimenzija. **mat** sadrži bilo koji cijeli broj, dok **maska** sadrži samo brojeve 0 ili 1. Funkcija treba obaviti maskiranje matrice **mat** matricom **maska**. Maskiranje se obavlja tako da na mjestu (i,j) matrice **mat** ostaje originalna vrijednost ako se na istom mjestu matrice **maska** nalazi vrijednost 1, inače se vrijednost u matrici **mat** zamjenjuje s nulom.
- Napišite glavni program u kojem će se učitati dimenzije matrica (max 100x150) i zatim uz pomoć generatora pseudoslučajnih brojeva generirati matrice **mat** (brojevi u intervalu $[100,200]$) i **maska** (interval $[0,1]$) te pozvati funkciju **maskiraj**. Glavni program treba ispisati matricu **mat** nakon maskiranja.

1. zadatak

```
#include <stdio.h>
#include <stdlib.h>
#define MAXSTUP 150
#define MAXRED 100
```

```
void maskiraj(int *mat, int *maska, int r, int s, int maxstup)
{
    int i,j;
    for(i=0;i<r;i++)
    {
        for(j=0;j<s;j++)
        {
            mat[i*maxstup+j] *= maska[i*maxstup+j];
        }
    }
}
```

1. zadatak

```
int main()
{
    int r,s,i,j, mat[MAXRED][MAXSTUP], maska[MAXRED][MAXSTUP];
    srand((unsigned)time(NULL));

    do {
        scanf("%d %d",&r,&s);
    }while(r<1 || r>MAXRED || s<1 || s>MAXSTUP);

    for(i=0;i<r;i++) {
        for(j=0;j<s;j++) {
            mat[i][j]=rand()%(200-100+1)+100;
            maska[i][j]=rand()%2;
        }
    }
    maskiraj(mat[0],maska[0],r,s,MAXSTUP);
    for(i=0;i<r;i++) {
        printf("\n");
        for(j=0;j<s;j++) {
            printf("%3d ",mat[i][j]);
        }
    }
    return 0;
}
```

2. zadatak

- Napisati funkciju prototipa:

```
void kopiraj(char *niz, char *s, char *prvi, char *drugi);
```

koja će iz zadanog znakovnog niza niz prepisati prvi dio u znakovni niz prvi, a drugi dio u znakovni niz drugi. Prvi dio je odijeljen od drugog dijela znakovnim nizom "s" (možete pretpostaviti da zadani znakovni niz ima točno jedan podniz s). Npr. izvršavanjem naredbi:

```
kopiraj("AAA ###ABCD", "###", prvi, drugi);  
printf("%s-%s", prvi, drugi);
```

na zaslone će se ispisati:

```
AAA -ABCD
```

2. zadatak

```
void kopiraj(char *niz, char *s, char *prvi, char *drugi){  
  
    char *pozS;  
    int duljina;  
  
    pozS = strstr(niz, s);  
  
    if (pozS != NULL) {  
        duljina = pozS - niz;  
        strncpy(prvi, niz, duljina);  
        prvi [duljina] = 0;  
        strcpy(drugi, pozS + strlen(s));  
    }  
}
```

strcpy VS strncpy?

Trebamo li nalijepiti '\0' na drugi?

3. zadatak

- Napisati funkciju čiji je prototip:

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

Funkcija treba uzastopnim slučajnim odabirom znakova, koji se nalaze u nizu **znakovi**, formirati lozinku zadane duljine. Dopusšteno je više puta odabrati isti znak.

- Na primjer, rezultat izvođenja naredbi:

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

bi mogao biti:

ABBAB ili A14BC ili 2CCC2 itd.

3. zadatak

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

void napraviLozinku(char *lozinka, char *znakovi, int duljina){

    int i, idx;

    srand( time(NULL) );

    for (i = 0; i < duljina; i++) {

        idx = (float)rand() / (RAND_MAX+1) * strlen(znakovi);
        /* ili idx = rand() % strlen(znakovi); */
        lozinka[i] = znakovi[idx];

    }

    lozinka[duljina] = '\\0';

}
```

4. zadatak

- Što će se ispisati izvođenjem sljedećeg programskog odsječka:

```
#define sum(x,y) x+y  
...  
int a, b=2, c=5;  
a = 2*sum(b,c);  
printf("%d", a);  
...
```

4. zadatak

- Rješenje:

```
sum(x,y) x+y
```

```
a = 2*sum(b,c);
```

```
a = 2 * x+y
```

```
    = 2 * 2 + 5
```

```
    = 9
```

5. zadatak

Koristeći prostor nad crtama, nadopunite dijelove programa (**definiranje novog tipa, definiranje varijable tog tipa**) kako bi programski odsječak bio ispravan.

```
#include <stdio.h>
int main() {
    _____ struct {

        int _____;

    } _____;

    mojtip _____;
    scanf( "%d", &x.a );
    ...
}
```

5. zadatak

Koristeći prostor nad crtama, nadopunite dijelove programa (**definiranje novog tipa, definiranje varijable tog tipa**) kako bi programski odsječak bio ispravan.

```
#include <stdio.h>
int main() {
    __typedef__ struct {

        int __a_____;

    } __mojtip_____;

    mojtip __x_____;
    scanf( "%d", &x.a);
    ...
}
```

6. zadatak

- Što će se ispisati obavljanjem sljedećeg programa:

```
int main () {  
    char s[15]= {0};  
    gets(s);  
    gets(s);  
    printf("%s",s);  
    puts(s);  
    return 0;  
}
```

- ukoliko korisnik unese:

123 456<Enter>

A B<Enter>

- **Rješenje:** A BA B

7. zadatak

- Što će se nalaziti zapisano u datoteci nakon izvođenja sljedećeg programa (gledajući po bajtovima) ?

```
#include <stdio.h>
int main() {
    char a = 'A';
    FILE *f;
    f = fopen("ime", "w");
    fprintf(f, "%3d", a);
    return 0;
}
```

- Ponuđena rješenja:
 - a) 65
 - b) 32 54 53
 - c) 32 32 65
 - d) 65 32 32

8. zadatak

- U formatiranoj datoteci ***racun.txt*** nalazi se popis kupljenih proizvoda. U svakom retku datoteke nalazi se šifra proizvoda (cijeli broj), količina kupljenog proizvoda (cijeli broj) i cijena (realni broj) odvojeni prazninom.
- U formatiranoj datoteci ***proizvodi.txt*** nalazi se popis proizvoda. U svakom retku datoteke nalazi se šifra proizvoda te naziv proizvoda (niz duljine do 50 znakova koji može sadržavati praznine).
- Napisati program koji će na zaslon ispisati naziv onog proizvoda čija je cijena u datoteci ***racun.txt*** najveća. Možete pretpostaviti da su cijene svih proizvoda različite.

8. zadatak

```
int main() {
    FILE *fracun, *fproizvod;
    int sifra, kolicina, sifmax;
    float cijena, max = 0;
    char naziv[50+1];
    fracun = fopen("racun.txt", "r");
    fproizvod = fopen("proizvodi.txt", "r");
    while (fscanf(fracun, "%d %d %f", &sifra, &kolicina, &cijena) == 3){
        if (max == 0 || cijena > max) {
            max = cijena;
            sifmax = sifra;
        }
    }
    while (fscanf(fproizvodi, "%d %50[^\n]", &sifra, naziv) == 2) {
        if (sifra == sifmax) {
            printf("Naziv najskupljeg proizvoda je %s", naziv);
            break;
        }
    }
    fclose(fracun);
    fclose(fproizvodi);
    return 0;
}
```

9. zadatak

- A)** Svaki zapis datoteke **slucajni.bin** treba sadržavati neki cijeli broj n koji može poprimiti vrijednosti iz intervala $[2, 8]$, te n cijelih brojeva koji poprimaju vrijednosti iz intervala $[150, 160]$. Napisati program koji će u novu neformatiranu datoteku **slucajni.bin** upisati 20 zapisa koji se formiraju na slučajan način. Za svaki zapis prvo treba na slučajan način odrediti n , a zatim na slučajan način odrediti svaku od n vrijednosti koje čine ostatak zapisa.
- B)** Napisati program koji će ispisati na zaslon sve zapise datoteke **slucajni.bin**

9. Zadatak A)

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#define MIN_N 2
#define MAX_N 8
#define DG 150
#define GG 160
#define BROJ_ZAPISA 20

int main () {
    FILE *izTok;
    int i, j, n, polje[MAX_N];
    izTok = fopen ("slucajni.bin", "wb");
    srand ((unsigned) time(NULL));
    for (i = 1; i <= BROJ_ZAPISA; i++) {
        n = rand() % (MAX_N - MIN_N + 1) + MIN_N;
        for (j = 0; j < n; j++)
            polje[j] = rand() % (GG - DG + 1) + DG;
        fwrite(&n, sizeof(n), 1, izTok);
        fwrite(polje, sizeof(int), n, izTok);
        /* ili fwrite(polje, sizeof(int)*n, 1, izTok); */
    }
    fclose (izTok);
    return 0;
}
```

9. Zadatak B)

```
#include <stdio.h>
#define MAX_N 8
int main () {
    FILE *ulTok;
    int i, n, polje[MAX_N];
    ulTok = fopen ("slucajni.bin", "rb");
    while (fread (&n, sizeof(n), 1, ulTok) == 1) {
        fread (polje, sizeof(int), n, ulTok);
        /* ili fread (polje, sizeof(int)*n, 1, ulTok); */
        printf("%d ", n);
        for (i = 0; i < n; i++)
            printf("%d ", polje[i]);
        printf("\n");
    }
    fclose (ulTok);
    return 0;
}
```

10. Zadatak (za vježbu)

- Prepisati prethodni primjer tako da se obje funkcionalnosti objedine u isti program sa funkcijama:
 - `napuniDatoteku() ;`
 - `ispisiDatoteku() ;`

 - Gdje i kako otvaramo datoteku?
 - Što sve prenosimo kao parametar?
 - Što ćemo sa poljem?
 - Gdje stavljamo `rewind`?

11. zadatak

- U formatiranu datoteku **v_s.txt** editorom su upisani podaci o visinama snijega na mjernim postajama (visina snijega u cm i naziv postaje). Primjer sadržaja datoteke:

122 Begovo Razdolje

7 Parg-Čabar

8 Sljeme

6 Zagreb-Grič

24 Zavižan

- Napisati program koji će sve zapise sa nadprosječnom visinom snijega prepisati u novu neformatiranu datoteku **v_s.bin**. Jedan zapis datoteke **v_s.bin** sadrži: mjesto(15+1 znak) i visinu snijega (int).

11. zadatak

```
int main () {  
  
    FILE *ulTok, *izTok;  
  
    int visina, n=0;  
    float srednja = 0;  
    char mjesto[15+1];  
  
    ulTok = fopen ("v_s.txt", "r");  
    izTok = fopen ("v_s.bin", "wb");  
  
    while (fscanf(ulTok, "%d %15[^\n]", &visina, mjesto) == 2) {  
        srednja += visina;  
        n++;  
    }  
}
```

10. zadatak

```
srednja /= n;

rewind(ulTok); //isto kao fseek(ulTok, 0L, SEEK_SET);

while (fscanf(ulTok, "%d %15[^\n]", &visina, mjesto) == 2)
    if (visina > srednja) {
        fwrite(&mjesto, sizeof(mjesto), 1, izTok);
        fwrite(&visina, sizeof(visina), 1, izTok);
    }

fclose (ulTok);
fclose (izTok);

return 0;
}
```