

PROGRAMIRANJE II – K1–GRUPA A
(13.04.2022)

- 1 (25 bodova) Neka je data matrica dimenzija $m \times n$. Napisati rekurzivnu funkciju koja računa ukupan broj putanja kojima se može doći od prve (0, 0) do posljednje ćelije u matrici ($m-1$, $n-1$). Moguće putanje za naredni korak su desno i dole u odnosu na početnu ćeliju. Smatrati da iz posljednjeg reda ($m-1$) ili posljednje kolone ($n-1$) postoji samo jedna putanja do posljednje ćelije. Prototip funkcije je:

```
int countPaths(int currentRow, int  
currentColumn, int m, int n)
```

- 2 (25 bodova) Neka je dat tip:

```
typedef struct predmet {  
    char id[15]; //identifikator  
    char naziv[64];  
    int ects;  
} PREDMET;
```

kojim se reprezentuju informacije o jednom predmetu.

Napisati funkciju *fun* sa promjenljivim brojem argumenata koja prihvata pokazivač na funkciju (parametar *u*), cijeli broj *n* i *n* pokazivača na PREDMET (neobavezni argumenti).

Parametar *u* predstavlja pokazivač na funkciju koja provjerava da li je ispunjen neki uslov za dati PREDMET (funkcija vraća vrijednost 0 ako uslov nije ispunjen, a vrijednost 1 ako uslov jeste ispunjen).

Funkcija *fun* treba da podatke o predmetima (neobavezni argumenti) koji ispunjavaju dati uslov upiše u binarnu datoteku sa nazivom "predmeti_izlazna.dat".

Prototip funkcije *fun* je:

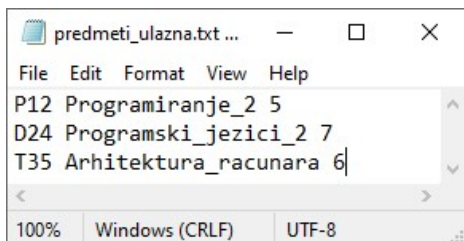
```
void fun(int (*u)(PREDMET *), int n,...);
```

Napisati funkciju *provjeri_sadrzaj* koja za zadati predmet (parametar *p*) provjerava da li se nalazi u tekstualnoj datoteci sa nazivom "predmeti_ulazna.txt". Ukoliko su podaci o predmetu upisani u datoteku, funkcija treba da vrati 1, inače treba da vrati 0. Predmeti su jednaki ukoliko su im jednaki identifikatori. U svakom redu datoteke je upisan podatak o jednom predmetu u sljedećem formatu: *identifikator naziv ects*.

Prototip funkcije *provjeri_sadrzaj* je:

```
int provjeri_sadrzaj(PREDMET *p);
```

Primjer ulazne datoteke je prikazan na sljedećoj slici:



- 3 (25 bodova) Neka je dat tip IMAGE i funkcija *cmpimg* koja poredi dva podatka tipa IMAGE. Funkcija *cmpimg* prihvata adrese dva podatka tipa IMAGE, a vraća vrijednost -1 (prvi podatak manji od drugog), 0 (podaci su jednaki) ili 1 (prvi podatak veći od drugog). Prototip funkcije je:

```
int cmpimg(const IMAGE *podatak1, const IMAGE  
*podatak2);
```

Napisati funkciju (korištenjem funkcije *cmpimg*) koja vrši indirektno (originalni niz ne smije biti modifikovan) sortiranje niza od *n* podataka tipa IMAGE. Funkcija treba da vrati dinamički niz adresa elemenata originalnog niza sortiranih u rastućem redoslijedu prema vrijednostima elemenata originalnog niza. Sortiranje treba izvršiti *bubble-sort* algoritmom. Prototip funkcije je:

```
IMAGE** sorting(IMAGE *niz, int n);
```

- 4 (25 bodova) Neka je dat tip:

```
typedef struct student {  
    char ime[64];  
    char prezime[64];  
    double prosjek;  
} STUDENT;
```

Napisati funkciju koja vrši binarnu pretragu studenata po svim poljima tipa STUDENT, a čiji je prototip:

```
STUDENT* pretraga(STUDENT *niz, int  
*sortiraniIndeksi, int n)
```

a koja vraća adresu studenta ukoliko je pronađen, u suprotnom se vraća vrijednost NULL. Pri tome parametar *niz* predstavlja adresu niza studenata (studenti u nizu nisu sortirani), parametar *sortiraniIndeksi* predstavlja adresu niza u kojem su sortirani indeksi originalnog niza (indirektno sortiranje po pravilu prvo ime, pa prezime pa prosjek studenta iz originalnog niza) i parametar *n* koji predstavlja dužinu prethodno navedenih nizova.