

Osnovi programiranja

III kolokvijum

10.1.2018.

I grupa

Na **Desktop**-u u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksa** i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka **mora** da se nalazi u fajlu **Zadatak1.c**, rešenje 2. zadatka **mora** da se nalazi u fajlu **Zadatak2.c**. Od dva ponuđena zadatka birate jedan koji ćete raditi. Nije dozvoljeno korišćenje globalnih promenljivih.

1. Napisati program u kome se najpre unose ime tekstualne datoteke sa podacima o **putnicima**, zatim ime tekstualne datoteke sa podacima o **autobusima** i ime izlazne datoteke. Definirati strukturu **putnik** koja sadrži sledeće podatke:

- **ID putnika (ceo broj),**
- **Ime putnika (jedna reč),**
- **Prezime putnika (jedna reč).**
- **Klasa karte(ceo broj)**

Definirati strukturu **autobus** koja sadrži sledeće podatke:

- **ID autobusa (ceo broj),**
- **Vreme polaska u satima i vreme polaska u minutima (dva cela broja),**
- **Vreme dolaska u satima i vreme dolaska u minutima (dva cela broja).**
- **Broj putnika (ceo broj)**

Napisati sledeće funkcije:

- Funkciju **unosPutnika** koja iz datoteke **datog naziva** čita podatke o putnicima. Na početku datoteke se nalazi broj putnika a zatim u svakom redu po ime, prezime i klasa karte svakog putnika. Podaci o putnicima se smeštaju u niz putnika i vraća se ceo broj koji predstavlja broj učitanih putnika. ID putnika se ne nalazi u datoteci već se dodeljuje na osnovu rednog broja učitanih putnika (prvi učitani putnik ima ID 1).
- Funkciju **ispisPutnika** koja u dati **fajl** ispisuje SVE podatke o putnicima koji se nalaze u nizu date dužine tako da su podaci o jednom putniku napisani u jednom redu, o drugom u drugom redu itd. *Jedan od argumenata funkcije je i fajl (nije naziv datoteke) u koji se upisuju podaci.*
- Funkciju **unosAutobusa** koja iz datog **fajla** čita sve podatke o autobusima, smešta ih u niz autobusa i vraća broj učitanih autobusa. Ne zna se koliko autobusa se nalazi u fajlu. Za svaki autobus se čitaju četiri cela broja, vreme polaska u satima (0-23), vreme polaska u minutima (0-59) i vreme dolaska u satima (0-23), vreme dolaska u minutima (0-59) (nije potrebno proveravati opseg unetih brojeva). ID autobusa se ne čita već se dodeljuje na osnovu rednog broja učitanih autobusa (prvi učitani autobus ima ID 1). Broj putnika za svaki autobus je u startu inicijalizovan na 0.
- Funkciju **rasporediPutnike** koja date putnike raspoređuje u date autobuse. Zna se da je broj mesta u svakom autobusu 5 i da svaki autobus ima isto polazište i odredište. Putnici se raspoređuju u autobuse koji imaju mesta i najkraće vreme putovanja. Prednost imaju putnici čija klasa je manji broj (na pr ako jedan putnik ima kartu klase 1 a jedan kartu klase 2, prednost ima putnik klase 1). Putnici koji se smeste u autobuse se izbacuju iz niza putnika. Smeštanje putnika u autobus podrazumeva ažuriranje broja putnika tog autobusa. Može da se desi da u autobusima ostanu prazna mesta, ili da ostanu neraspoređeni putnika. Funkcija vraća broj neraspoređenih putnika.
- Funkciju **ispisAutobusa** koja u dati **fajl** ispisuje SVE podatke o autobusima koji se nalaze u nizu date dužine tako da su podaci o jednom autobusu napisani u jednom redu, o drugom u drugom redu itd. *Jedan od argumenata funkcije je i fajl (nije naziv datoteke) u koji se upisuju podaci.*
- Funkciju **formSifra** koja na osnovu imena, prezimena i klase karte datog putnika formira njegovu šifru po šablonu **ImeKlasaPrezime**. (na primer ako se putnik zove Ana, preziva Popovic i ima kartu klase 1, njena šifra je Ana1Popovic)

U glavnom delu programa učitati nazive ulaznih i izlaznih datoteka, zatim formirati niz od putnika koji se nalaze u prvoj ulaznoj datoteci koristeći funkciju **unosPutnika** i ispisuje ga u izlazni fajl koristeći funkciju **ispisPutnika**. Zatim:

- koristeći funkciju **unosAutobusa** formirati niz autobusa koji se nalaze u drugoj ulaznoj datoteci i ispisati dati niz u isti izlazni fajl ispod niza putnika, koristeći funkciju **ispisAutobusa**.
- za svakog putnika u nizu putnika u izlazni fajl štampati formiranu šifru (jednu ispod druge) koristeći funkciju **formSifra**.
- koristeći funkciju **rasporediPutnike** rasporediti putnike u autobuse.

- u izlaznu datoteku koristeći funkciju **ispisAutobusa** štampati podatke o stanjima svih autobusa, zatim koristeći funkciju **ispisPutnika** štampati u izlaznu datoteku podatke o preostalim putnicima ako ih ima, ako ih nema u datoteci odštampati „SVI PUTNICI SU RASPOREDJENI“.

NAPOMENA: Redosled autobusa u nizu je moguće menjati dok redosled putnika nije. Ukoliko se zadatak uspešno reši raspoređivanjem putnika po autobusima onako kako se oni nalaze u nizu, bez obzira na dužinu puta, zadatak vredi 3 poena manje.

Ulaz1.txt	Ulaz2.txt	Izlaz.txt
15 Leka Lekic 4 Ana Petrovic 1 Milica Nikolic 3 Petar Petrovic 1 Jana Janic 4 Milos Tasic 2 Filip Maric 2 Sanja Saric 3 Milica Milic 1 Milos Ilic 2 Danilo Dacic 3 Ilija Ilic 1 Mario Tomas 3 Marko Markovic 1 Stefan Stevic 5	10 55 12 30 13 25 14 00	1 Leka Lekic 4 2 Ana Petrovic 1 3 Milica Nikolic 3 4 Petar Petrovic 1 5 Jana Janic 4 6 Milos Tasic 2 7 Filip Maric 2 8 Sanja Saric 3 9 Milica Milic 1 10 Milos Ilic 2 11 Danilo Dacic 3 12 Ilija Ilic 1 13 Mario Tomas 3 14 Marko Markovic 1 15 Stefan Stevic 5 1 10 55 12 30 0 2 13 25 14 00 0 2 13 25 14 00 5 1 10 55 12 30 5 1 Leka Lekic 4 5 Jana Janic 4 11 Danilo Dacic 3 13 Mario Tomas 3 15 Stefan Stevic 5

- Napisati program u kome se najpre unose ime tekstualne datoteke sa podacima o ciframa koje deca prijavljuju na takmičenju, zatim ime tekstualne datoteke sa podacima o mogućim ciframa u bazi i ime izlazne datoteke. Deca mogu prijaviti cifre od 1 do 9. Napisati sledeće funkcije:
 - Funkciju **unosPrijavljenihCifara** koja iz datoteke **datog naziva** čita cifre. Na početku datoteke se nalazi broj koji predstavlja koliko je bilo dece a zatim cifre koje je svako dete prijavilo. Prijavljene cifre se smeštaju u niz celih brojeva i vraća se ceo broj koji predstavlja broj učitanih cifara, odnosno broj dece.
 - Funkciju **ispisCifara** koja u **dati fajl** ispisuje cifre koje se nalaze u nizu date dužine tako da se u prvom redu nalazi broj elemenata niza a u drugom redu elementi niza razdvojeni razmakom. *Jedan od argumenata funkcije je i fajl (nije naziv datoteke) u koji se upisuju podaci.*
 - Funkciju **unosCifaraBaze** koja iz datoteke **datog naziva** čita cifre i smešta ih u niz celih brojeva i vraća ceo broj koji predstavlja broj učitanih cifara, odnosno broj cifara u bazi. Ne zna se koliko cifara ima u datoteci.
 - Funkciju **pokriivanjeCifara** koja za dati niz prijavljenih cifara i dati niz cifara koje se nalaze u bazi vrši sledeću proveru - ukoliko se permutacijom cifara u bazi može dobiti niz prijavljenih cifara tada prijavljeni niz cifara pokriva cifre baze. U tom slučaju funkcija vraća 1 u suprotnom 0.
 - Funkciju **formSifra** koja od datog niza prijavljenih cifara formira šifru sastavljenu od malih slova engleske abecede. Šifra se formira tako što se svaka cifra iz niza šifruje slovom čiji je to redni broj u engleskom alfabetu malih slova. (primer: od niza **3 1 2** dobija se šifra **cab**)

U glavnom delu programa učitati nazive ulaznih i naziv izlazne datoteke, zatim formirati niz cifara koje su prijavila deca i koji se nalaze u prvoj ulaznoj datoteci koristeći funkciju **unosPrijavljenihCifara** i ispisuje ga u izlazni fajl koristeći funkciju **ispisCifara**. Zatim:

- koristeći funkciju **unosCifaraBaze** iz druge ulazne datoteke učitati cifre koje predstavljaju cifre u bazi i korišćenjem funkcije **ispisCifara** upisati niz u isti izlazni fajl.

- koristeći funkciju **pokrivanjeCifara** ispitati da li je moguće izvršiti pokrivanje učitanih cifara ciframa koje se nalaze u bazi. Korišćenjem funkcije **formSifra**, formirati šifru na osnovu niza prijavljenih cifara i upisati je u isti izlazni fajl. Ukoliko je moguće izvršiti pokrivanje, ispred šifre dodati poruku „POKRIVENO“, a ako nije moguće, dodati poruku „NEPOKRIVENO“

Ulaz1.txt	Ulaz2.txt	Izlaz.txt
8	4	8
1	2	1 4 1 5 6 7 3 2
4	1	8
1	6	4 2 1 6 1 5 3 7
5	1	POKRIVENOadaefgcb
6	5	
7	3	
3	7	
2		

Broj poena: 1. zadatak – 21 poena, 2. zadatak – 14 poena

Vreme izrade: 120 minuta

Osnovi programiranja III kolokvijum

10.1.2018.

II grupa

Na **Desktop**-u u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksa** i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka **mora** da se nalazi u fajlu **Zadatak1.c**, rešenje 2. zadatka **mora** da se nalazi u fajlu **Zadatak2.c**. Od dva ponuđena zadatka birate jedan koji ćete raditi. Nije dozvoljeno korišćenje globalnih promenljivih.

- Napisati program u kome se najpre unose ime tekstualne datoteke sa podacima o **ucenicima**, zatim ime tekstualne datoteke sa podacima o **ucionicama** i ime izlazne datoteke. Definisati strukturu **ucenik** koja sadrži sledeće podatke:
 - **ID ucenika (ceo broj)**,
 - **Ime ucenika (jedna reč)**,
 - **Prezime ucenika (jedna reč)**.
 - **Prosek (realan broj)**

Definisati strukturu **ucionica** koja sadrži sledeće podatke:

- **ID ucionice (ceo broj)**,
- **Vreme od kada je slobodna u satima i u minutima (dva cela broja)**,
- **Vreme do kada je slobodna u satima i u minutima (dva cela broja)**.
- **Broj zauzetih mesta (ceo broj)**

Napisati sledeće funkcije:

- Funkciju **unosUcenika** koja iz datoteke **datog naziva** čita podatke o ucenicima. Na početku datoteke se nalazi broj ucenika a zatim u svakom redu po ime, prezime i prosek svakog ucenika. Podaci o ucenicima se smeštaju u niz ucenika i vraća se ceo broj koji predstavlja broj učitanih ucenika. ID ucenika se ne nalazi u datoteci već se dodeljuje na osnovu rednog broja učitanih ucenika (prvi učitani ucenik ima ID 1).
- Funkciju **ispisUcenika** koja u dati **fajl** ispisuje SVE podatke o ucenicima koji se nalaze u nizu date dužine tako da su podaci o jednom uceniku napisani u jednom redu, o drugom u drugom redu itd. *Jedan od argumenata funkcije je i fajl (nije naziv datoteke) u koji se upisuju podaci.*
- Funkciju **unosUcionica** koja iz datog **fajla** čita sve podatke o ucionicama, smešta u niz ucionica i vraća broj učitanih ucionica. Ne zna se koliko ucionica se nalazi u fajlu. Za svaku ucionicu se čitaju četiri cela broja, vreme od kada je slobodna u satima (0-23), vreme od kada je slobodna u minutima (0-59), vreme do kada je slobodna u satima (0-23) i vreme do kada je slobodna u minutima (0-59) (nije potrebno proveravati opseg unetih brojeva). ID ucionice se ne čita već se

dodeljuje na osnovu rednog broja učitane ucionice (prva učitana ucionica ima ID 1). Broj zauzetih mesta za svaku ucionicu je u startu inicijalizovan na 0.

- j) Funkciju **rasporediUcenike** koja date ucenike raspoređuje u date ucionice. Zna se da je broj mesta u svakoj ucionici 5. Ucenici se raspoređuju u ucionice koje imaju mesta i najduže su slobodne. Prednost imaju ucenici koji imaju veći prosek (npr. ako jedan ucenik ima prosek 9.5 a jedan 8.3, prednost ima prvi ucenik). Ucenici koji se smeste u ucionice se izbacuju iz niza ucenika. Smeštanje ucenika u ucionicu podrazumeva ažuriranje broja zauzetih mesta te ucionice. Može da se desi da u ucionicama ostane praznih mesta, ili da ostane neraspoređenih ucenika. Funkcija vraća broj neraspoređenih ucenika.
- k) Funkciju **ispisUcionica** koja u dati **fajl** ispisuje SVE podatke o ucionicama koje se nalaze u nizu date dužine tako da su podaci o jednoj ucionici napisani u jednom redu, o drugoj u drugom redu itd. *Jedan od argumenata funkcije je i fajl (nije naziv datoteke) u koji se upisuju podaci.*
- l) Funkciju **formSifra** koja na osnovu imena, prezimena i ID-a datog ucenika formira njegovu sifru po šablonu **ImeIDPrezime**. (na primer ako se ucenik zove Ana, preziva Popovic i ima ID 1, njena šifra je Ana1Popovic)

U glavnom delu programa učitati nazive ulaznih i izlazne datoteke, zatim formirati niz od ucenika koji se nalaze u prvoj ulaznoj datoteci koristeći funkciju **unosUcenika** i ispisuje ga u izlazni fajl koristeći funkciju **ispisUcenika**. Zatim:

- koristeći funkciju **unosUcionica** formirati niz ucionica koje se nalaze u drugoj ulaznoj datoteci i ispisati dati niz u isti izlazni fajl ispod niza ucenika, koristeći funkciju **ispisUcionica**.
- za svakog ucenika u nizu ucenika u izlazni fajl štampati formiranu šifru (jednu ispod druge) koristeći funkciju **formSifra**.
- koristeći funkciju **rasporediUcenike** rasporediti ucenike u ucionice.
- u izlaznu datoteku koristeći funkciju **ispisUcionica** štampati podatke o stanjima svih ucionica, zatim koristeći funkciju **ispisUcenika** štampati u izlaznu datoteku podatke o preostaim ucenicima ako ih ima, ako ih nema u datoteci odštampati „SVI UCENICI SU RASPOREDJENI“.

NAPOMENA: Redosled učionica u nizu je moguće menjati dok redosled učenika nije. Ukoliko se zadatak uspešno reši raspoređivanjem učenika po učionicama onako kako se one nalaze u nizu, bez obzira na dužinu slobodnog vremena, zadatak vredi 3 poena manje.

Ulaz1.txt	Ulaz2.txt	Izlaz.txt
15	13 25	1 Leka Lekic 4.5
Leka Lekic 4.5	14 00	2 Ana Petrovic 1.6
Ana Petrovic 1.6	10 55	3 Milica Nikolic 3.8
Milica Nikolic 3.8	12 30	4 Petar Petrovic 1.2
Petar Petrovic 1.2		5 Jana Janic 4.9
Jana Janic 4.9		6 Milos Tasic 2.1
Milos Tasic 2.1		7 Filip Maric 2.6
Filip Maric 2.6		8 Sanja Saric 3.5
Sanja Saric 3.5		9 Milica Milic 1.1
Milica Milic 1.1		10 Milos Ilic 2.3
Milos Ilic 2.3		11 Danilo Dacic 3.7
Danilo Dacic 3.7		12 Ilija Ilic 1.9
Ilija Ilic 1.9		13 Mario Tomas 3.3
Mario Tomas 3.3		14 Marko Markovic 4.5
Marko Markovic 4.5		15 Stefan Stevic 5.0
Stefan Stevic 5.0		1 13 25 14 00 0
		2 10 55 12 30 0
		2 10 55 12 30 5
		1 13 25 14 00 5
		2 Ana Petrovic 1.6
		4 Petar Petrovic 1.2
		6 Milos Tasic 2.1
		9 Milica Milic 1.1
		12 Ilija Ilic 1.9

4. Napisati program u kome se najpre unose ime tekstualne datoteke sa podacima o celim brojevima koje deca prijavljuju na takmičenju, zatim ime tekstualne datoteke sa podacima o mogućim brojevima u bazi i ime izlazne datoteke. Napisati sledeće funkcije:

- f) Funkciju **unosPrijavljenihBrojeva** koja iz datoteke **datog naziva** čita pozitivne cele brojeve. Na početku datoteke se nalazi broj koji predstavlja koliko je bilo dece a zatim brojevi koje je svako dete prijavilo. Prijavljeni brojevi se smeštaju u niz celih brojeva i vraća se ceo broj koji predstavlja broj učitanih brojeva, odnosno broj dece.
- g) Funkciju **ispisBrojeva** koja u **dati fajl** ispisuje brojeve koji se nalaze u nizu date dužine tako da se u prvom redu nalazi broj elemenata niza a u drugom redu elementi niza razdvojeni razmakom. *Jedan od argumenata funkcije je i fajl (nije naziv datoteke) u koji se upisuju podaci.*
- h) Funkciju **unosBrojevaBaze** koja iz datoteke datog naziva čita brojeve i smešta ih u niz celih brojeva a vraća ceo broj koji predstavlja broj učitanih brojeva, odnosno broj brojeva u bazi. Ne zna se koliko brojeva ima u datoteci.
- i) Funkciju **pokrivanjeBrojeva** koja za dati niz prijavljenih brojeva i dati niz brojeva koje se nalaze u bazi vrši sledeću proveru - ukoliko se permutacijom brojeva u bazi može dobiti niz prijavljenih brojeva tada prijavljeni niz brojeva pokriva brojeve baze. U tom slučaju funkcija vraća 0, u suprotnom 1.
- j) Funkciju **formSifra** koja od datog niza prijavljenih cifara formira šifru sastavljenu od velikih slova engleske abecede. Šifra se formira tako što se svaki broj iz niza šifruje slovom čiji je to redni broj u engleskom alfabetu velikih slova. (primer: od niza **10 1 20** dobija se šifra **JAT**).

U glavnom delu programa učitati nazive ulaznih i naziv izlazne datoteke, zatim formirati niz brojeva koje su prijavila deca i koji se nalaze u prvoj ulaznoj datoteci koristeći funkciju **unosPrijavljenihBrojeva** i ispisuje ga u izlazni fajl koristeći funkciju **ispisBrojeva**. Zatim:

- koristeći funkciju **unosBrojevaBaze** iz druge ulazne datoteke učitati brojeve koje predstavljaju brojeve u bazi i korišćenjem funkcije **ispisCifara** upisati niz u isti izlazni fajl.
- koristeći funkciju **pokrivanjeBrojeva** ispitati da li je moguće izvršiti pokrivanje učitanih brojeva brojevima koji se nalaze u bazi. Korišćenjem funkcije **formSifra**, formirati šifru i upisati je u isti izlazni fajl. Ukoliko je moguće izvršiti pokrivanje, ispred šifre dodati poruku „POKRIVENO“, a ako nije moguće, dodati poruku „NEPOKRIVENO“

Ulaz1.txt	Ulaz2.txt	Izlaz.txt
8	4	8
10	20	10 4 1 5 6 7 3 20
4	10	8
1	6	4 20 10 6 1 5 3 7
5	1	POKRIVENOJDAEFGCT
6	5	
7	3	
3	7	
20		

Broj poena: 1. zadatak – 21 poena, 2. zadatak – 14 poena

Vreme izrade: 120 minuta