Strukture podataka i algoritmi 1

II kolokvijum

05. 06. 2019.

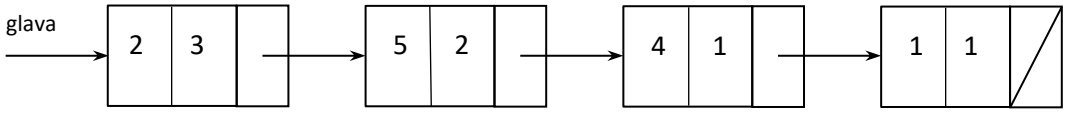
Na **Desktop-u** u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksa** i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka **mora** da se nalazi u fajlu **Zadatak1.c**, rešenje 2. zadatka **mora** da se nalazi u fajlu **Zadatak2.c**. Rešavaju se **oba** zadatka.

1. Napisati funkciju **Kreiraj** koja prihvata ceo broj i formira listu, tako da svaki element liste sadrži cifru broja i broj pojavljivanja te cifre u broju. Napisati program koji za uneta **dva cela broja** treba da formira **dve liste** tako da svaka sadrži cifru i broj pojavljivanja te cifre (pogledati primer). Nakon formiranja obe liste, ispisati liste (svaki element u novom redu). Proveriti da li je od prvog broja moguće formirati drugi broj (**ne moraju** da se iskoriste sve cifre prvog broja). Ukoliko je to moguće, na izlaz ispisati samo **DA**, u suprotnom **NE**. Ukoliko nije moguće formrati drugi broj od prvog, ispisati koje cifre nedostaju i koliko primeraka te cifre nedostaje.

Primer kako treba formirati listu:

Broj: 2145252

Lista: glava



Ulaz: 2214525 215245
Izlaz:

5 2
2 3
4 1
1 1

5 2
4 1
2 2
1 1
DA

Ulaz: 214525 2195545
Izlaz:
5 2
2 2
4 1
1 1

5 3
4 1
9 1
1 1
2 1
NE
5 1
9 1

Napomena. Zadatak nosi **14** poena. Liste je dozvoljeno kreirati bilo dodavanjem elemenata na početak, bilo na kraj liste. Ukoliko elementi liste ne sadrže informaciju o broju pojavljivanja cifre, već samo cifre, vrednost zadatka se umanjuje za 15%. Ukoliko se liste kreiraju u glavnom delu programa, vrednost zadatka se umanjuje za 15%. Ukoliko se ne proverava da li se drugi broj može formirati od prvog, vrednost zadatka se umanjuje za 20%. Ukoliko se ne ispiše koje cifre nedostaju za formiranje drugog, vrednost zadatka se umanjuje za 10%.

2. U toku je jedan od najvećih teniskih turnira na svetu – **Roland Garros**. Na turniru se takmiči **N** tenisera (**N** će na ulazu biti zadato kao stepen dvojke – 1, 2, 4, 8, ...). Za svakog **tenisera**, pamti se **redni broj**, **ime** (više reči), **plasman** (reč oblika „*1. kolo*“, „*2. kolo*“, „*3. kolo*“, ...) i **završio takmičenje** (1 – teniser je završio takmičenje, 0 – teniser je i dalje na turniru).
    a) Definisati odgovarajuću strukturu podataka koja opisuje **tenisera**.
    b) Napisati funkciju ***UnesiTenisere*** koja iz datoteke sa nazivom *Teniseri.txt* učitava broj tenisera na turniru, kao i njihova imena i smešta ih u odgovarajuće strukture podataka, postavljajući im redom redne brojeve od 1 do N, pri čemu je svaki teniser u prvom kolu i dalje je na turniru.
    c) Napisati funkciju ***IspisiTenisere*** koja ispisuje sve podatke o svim teniserima koji učestvuju na turniru.
    d) Napisati funkciju ***Igra*** koja simulira odigravanje mečeva sa turnira. U datoteci sa nazivom *Mecevi.txt* se nalaze podaci o odigranim mečevima na turniru. U prvom kolu igrač sa rednim brojem 1 igra sa igračem sa rednim brojem 2, igrač sa rednim brojem 3 igra sa igračem sa rednim brojem 4, ..., igrač sa rednim brojem N-1 igra sa igračem sa rednim brojem N. Svaki red u datoteci predstavlja jedan odigran meč, rezultati mečeva su dati redom (videti sliku ***Zreb.png***). Ukoliko se u redu nalazi broj **1** pobedio je igrač koji je gornji u žrebu (ima manji redni broj), a ukoliko se nalazi broj **2** pobedio je igrač koji je donji u žrebu (ima veći redni broj). Igra traje dok se ne odigraju svi mečevi i dok turnir ne dobije pobednika. Na kraju svakog kola, ispisivati podatke o svim teniserima na turniru.
    e) Napisati funkciju ***main*** koja unosi tenisere. Zatim ispisati podatke o svim teniserima. Nakon toga simulirati odigravanje turnira koristeći funkciju ***Igra***. Na kraju ispisati pobednika turnira.

Teniseri.txt	Mecevi.txt	Izlaz
8 Novak Djokovic Jo-Wilfried Tsonga Andy Muray Dominic Thiem Rafael Nadal Gael Monfils Roger Federer Alexander Zverev	1 2 1 1 1 1 1	Pogledati fajl **`izlaz.txt`**

Napomena. Zadatak nosi **14** poena. Ukoliko se podaci ne učitavaju iz datoteke, već sa standardnog ulaza, vrednost zadatka se umanjuje za 10%. Ukoliko se imena tenisera pamte kao jedna reč, vrednost zadatka se umanjuje za 10%. Ukoliko se **plasman** pamti kao broj, umesto kao reč, vrednost zadatka se umanjuje za 10%. Realizacija fukcije **Igra** predstavlja 50% vrednosti zadatka.

Slika ***Zreb.png*** odgovara test primeru iznad!

Rešenje prvog zadatka:

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

typedef struct element
{
	int cifra, brojPojavljivanja;
	struct element *sledeci;
} Element;

Element* alociraj(int cifra)
{
	Element *novi = (Element*)malloc(sizeof(Element));
	novi->cifra = cifra;
	novi->brojPojavljivanja = 1;
	novi->sledeci = NULL;
}

void dodaj(Element **p, int cifra)
{
	if (*p == NULL)
		*p = alociraj(cifra);
	else
	{
		Element *pomPr;
		Element *pom = *p;

		while (pom)
		{
			if (pom->cifra == cifra)
			{
				pom->brojPojavljivanja++;
				return;
			}

			pomPr = pom;
			pom = pom->sledeci;
		}

		pomPr->sledeci = alociraj(cifra);
	}
}

void ispis(Element *p)
{
	while (p)
	{
		printf("%d %d\n", p->cifra, p->brojPojavljivanja);
		p = p->sledeci;
	}
}
```

```
void formiraj(Element **p, int x)
{
	while (x > 0)
	{
		dodaj(p, x % 10);
		x /= 10;
	}
}

void ucitaj(Element **p, Element **q)
{
	int x, y;
	scanf("%d %d", &x, &y);

	formiraj(p, x);
	formiraj(q, y);

}

int proveriBrojeve(Element *p, Element *q)
{
	while (q)
	{
		int postojiCifra = 0;
		Element *pom = p;

		while (pom)
		{
			if (q->cifra == pom->cifra)
			{
				if(q->brojPojavljivanja > pom->brojPojavljivanja)
					return 0;

				postojiCifra = 1;
			}

			pom = pom->sledeci;
		}

		if (postojiCifra == 0)
			return 0;

		q = q->sledeci;
	}

	return 1;
}
```

```
void nadjiRazliku(Element *p, Element *q)
{
	while (q)
	{
		int postojiCifra = 0;
		Element *pom = p;

		while (pom)
		{
			if (q->cifra == pom->cifra)
			{
				if(q->brojPojavljivanja > pom->brojPojavljivanja)
					printf("%d  %d\n",  q->cifra,  q->brojPojavljivanja  -  pom-
>brojPojavljivanja);

				postojiCifra = 1;
			}

			pom = pom->sledeci;
		}

		if (postojiCifra == 0)
			printf("%d %d\n", q->cifra, q->brojPojavljivanja);

		q = q->sledeci;
	}

}

int main()
{
	Element *p = NULL;
	Element *q = NULL;

	ucitaj(&p, &q);
	ispis(p);
	printf("\n");
	ispis(q);

	if (proveriBrojeve(p, q) == 0)
	{
		printf("NE\n");
		nadjiRazliku(p, q);
	}
	else
		printf("DA\n");

	return 0;
}
```

Rešenje drugog zadatka:

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

typedef struct {
    int rbr;
    char ime[30];
    char plasman[10];
    int ispao;
} Teniser;

Teniser *UnesiTenisere(int *N) {
    int i;
    char s[30];

    FILE *f = fopen("Teniseri.txt", "r");
    fscanf(f, "%d", N);
    fgetc(f);

    Teniser *igraci = (Teniser *)malloc(*N * sizeof(Teniser));
    for (i = 0; i < *N; i++) {
        fgets(s, 30, f);
        if (s[strlen(s) - 1] == '\n')
            s[strlen(s) - 1] = 0;
        strcpy(igraci[i].ime, s);
        strcpy(igraci[i].plasman, "1. kolo");
        igraci[i].rbr = i + 1;
        igraci[i].ispao = 0;
    }

    fclose(f);
    return igraci;
}

void IspisiTenisere(Teniser *igraci, int N) {
    int i;

    for (i = 0; i < N; i++)
        printf("Redni broj: %d\tIme: %-20s\tPlasman: %s\tIspao: %d\n",
               igraci[i].rbr, igraci[i].ime, igraci[i].plasman, igraci[i].ispao);
    printf("---------------------------------\n");
}
```

```
void Igra(Teniser *igraci, int N) {
    int i, kolo = 1, brMeceva = N / 2;
    Teniser *prvi, *drugi;
    FILE *f = fopen("Mecevi.txt", "r");

    while (brMeceva > 0) {
        prvi = drugi = NULL;
        for (i = 0; i < N; i++) {
            if (igraci[i].ispao == 0 && prvi == NULL) {
                prvi = &igraci[i];
                continue;
            }
            if (igraci[i].ispao == 0 && prvi != NULL) {
                drugi = &igraci[i];

                int p;
                fscanf(f, "%d", &p);
                if (p == 1) {
                    prvi->plasman[0]++;
                    drugi->ispao = 1;
                } else {
                    drugi->plasman[0]++;
                    prvi->ispao = 1;
                }

                prvi = drugi = NULL;
            }
        }
        printf("\n--------------- Rezultati %d. kola ----------------\n", kolo);
        IspisiTenisere(igraci, N);

        kolo++;
        brMeceva /= 2;
    }
    fclose(f);
}

int main() {
    int i, N;
    Teniser *igraci = UnesiTenisere(&N);

    printf("--------------- Svi teniseri na turniru ----------------\n");
    IspisiTenisere(igraci, N);

    Igra(igraci, N);

    for (i = 0; i < N; i++) {
        if (igraci[i].ispao == 0) {
            printf("\n----- Pobednik je: *** %s *** -------\n", igraci[i].ime);
            break;
        }
    }

    return 0;
}
```