

Strukture podataka i algoritmi 1

I kolokvijum

23.04.2016.

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*, rešenje 3. zadatka *mora* da se nalazi u fajlu *Zadatak3.c*. Od tri ponudjena zadatka birate dva koja ćete raditi.

Napisati C program koji:

1. sadrži funkciju **uređeniPar** koja za dva data karaktera (karakter i mogu biti samo mala i velika slova engleskog alfabeta i to nije potrebno proveravati) proverava da li su dati u nerastućem poretku (da li se prvi dati karakter nalazi ispred drugog u leksikografskom poretku). U glavnom delu uneti ceo broj *n*, a zatim *n* karaktera. Koristeći funkciju **uređeniPar** proveriti da li su uneti karakteri u nerastućem poretku. Ukoliko jesu ispisati *DA*, u suprotnom *NE*.

```
ulaz:          izlaz:
14             DA
ZzzsSmMkdddDbA
```

2. sadrži funkciju koja računa najveći broj poena koje je moguće dobiti pri jednom bacanju 6 kockica za JAMB. Igrač ima na raspolaganju da popuni jedno od 3 polja, a popunjava ono koje mu donosi maksimalni dobitak:

- 1) polje za triling – u koje se upisuju poeni ako su u bacanju dobijena 3 ista broja
- 2) polje za kare - u koje se upisuju poeni ako su u bacanju dobijena 4 ista broja
- 3) polje za jamb - u koje se upisuju poeni ako je u bacanju dobijeno 5 istih brojeva

Poeni koji se popunjavaju u odgovarajuća polja se računaju preko formule $cifra * mno + bonus$ pomoću funkcije `int mogućDobitak(int k[], int koliko, int mno, int bonus)` koja za date argumente:

- niz od 6 cifara, koje predstavljaju brojeve dobijene bacanjem 6 kockica (nije potrebno proveravati, sigurno su brojevi od 1-6),
- koliko mogućih istih cifara se traži,
- vrednost kojom se množi cifra koja se ponovila zahtevani broj puta,
- bonus koji se dodaje na to,

vraća mogući dobitak ukoliko postoji neka cifra ponovljena dati broj puta, inače vraća 0. Pravila koja se koriste prilikom igranja društvene igre, data su u tabeli i koriste istu funkciju mogućDobitak za izračunavanje mogućeg dobitka.

<i>Kockice</i>	<i>Polje</i>	<i>Dobitak</i>
3 iste cifre	triling	$cifra * 20$
4 iste cifre	kare	$cifra * 4 + 40$
5 istih cifara	jamb	$cifra * 5 + 60$

Sadrži f-ju **maxDobitak** koja za dati niz mogućih dobitaka vraća maksimalni dobitak.

U glavnom delu programa uneti brojeve koje su na kockicama dobili Maša i Majk (prvo šest brojeva koje je dobila Maša, a zatim 6 brojeva koje je dobio Majk). Koristeći napisane f-je štampati maksimalan dobitak svakog od njih, a zatim ime porednika. Izlaz sadrži u prvoj liniji dva cela broja razdvojena razmakom, u drugoj liniji ime Majk ili Masa.

```
ulaz:          izlaz:
1 2 2 4 2 1    40 100
2 5 2 5 2 5    Majk
```

3. učitava tekst i pronalazi pojavljivanja novčanih iznosa u okviru teksta. Novčanim iznosom smatra se svaki uzastopni niz cifara praćen nizom rsd (bez razmaka između poslednje cifre i niza rsd). Program treba da ispiše ukupnu sumu svih novčanih iznosa pomenutih u tekstu. Izlaz sadrži ukupnu sumu praćenu oznakom rsd (bez razmaka između poslednje cifre i niza rsd).

```
ulaz:          izlaz:
23.4.2016.    2045rsd
skuplja se dobrovoljni prilog za izgradnju
muzicke sobe na IMI-ju,4 profesora su dala po 500rsd,
specijalan prilog stigao je od dekana 1200rsd.
Broj indeksa 66/14 dao je 320rsd, pronasli so i
ostavljenih 20rsd, kao i metalne kovanice od po 5rsd i
10rss.
```

NAPOMENA: Nije dozvoljeno korišćenje funkcije `gets()`, niti funkcija iz biblioteke `string.h`

Broj poena: 1. zadatak – 6 poena, 2. zadatak – 10 poena, 3. zadatak – 11 poena

Vreme izrade: 90 minuta