

## Laboratorijske vježbe 8

1. Napisati program kojim se učitava niz **X** od **N** cijelih brojeva. Program treba da formira novi niz **Y** koji sadrži sve elemente niza **X** osim najmanjeg i najvećeg. Na kraju štampati dobijeni niz **Y**. Memoriju za nizove **X** i **Y** zauzeti dinamički.
2. Napisati funkciju **asciiKod** koja za argument ima niz cijelih brojeva **X**, dužine **N**. Funkcija treba da dinamički formira i vrati string koji se dobija tako što se svaki element niza za koji važi  $65 \leq X[i] \leq 90$  (opseg velikih slova u ASCII tabeli) ili  $97 \leq X[i] \leq 122$  (opseg malih slova u ASCII tabeli) konvertuje u odgovarajući ASCII karakter. Napisati i glavni program koji učitava niz, poziva napisanu funkciju i štampa rezultujuću string funkcije.  
**Primjer:** Za niz  $X=[65\ 66\ 99]$ , funkcija **asciiKod** treba da vrati string "ABc".
3. U svakom redu fajla **cijeli.txt** je upisano sedam cijelih brojeva. Napisati program koji formira fajl **naj.txt** čiji svaki red sadrži najveći i najmanji cijeli broj iz odgovarajućeg reda fajla **cijeli.txt**.
4. Napisati program koji dinamički formira niz cijelih brojeva **X**, dužine **N**, a zatim učitava i pozitivan jednocifren broj **K**. Program treba da formira fajl pod nazivom **brojevi.txt** koji u svakom redu sadrži elemente niza **X** čija je cifra najveće težine (prva cifra sa lijeve strane) jednaka broju **K**. U slučaju da **K** nije pozitivan jednocifren broj, izaći iz programa sa odgovarajućom porukom.
5. Tekstualni fajl **Brojevi.txt** sadrži 20 redova, pri čemu svaki red sadrži jedan 20-cifreni prirodan broj, kako je prikazano ispod. Napisati program koji određuje koja se cifra pojavljuje najviše puta.

```
37107287533902102798
46376937677490009712
...
44274228917432520349
54370070576826684624
```

### Za vježbu

1. Napisati program kojim se učitava niz **A** od **N** cijelih brojeva. Posmatrajmo cijeli broj **P**, takav da važi  $0 < P < N$ , koji predstavlja tačku presjeka niza na dva podniza:  
 $A[0], A[1], \dots, A[P-1]$  i  $A[P], A[P+1], \dots, A[N-1]$ .  
Napisati funkciju koja za argument ima niz **A** i njegovu dužinu **N**, i koja vraća minimalnu apsolutnu razliku između ova dva podniza za svako moguću vrijednost **P**.  
Na primjer, ako je niz  $A = [3, 1, 2, 4, 3]$ , ovaj se niz može podeliti na 4 različita načina, i vrijednosti apsolutnih razlika su:  
 $P = 1$ , razlika  $|3-10| = 7$   
 $P = 2$ , razlika  $|4-9| = 5$   
 $P = 3$ , razlika  $|6-7| = 1$   
 $P = 4$ , razlika  $|10-3| = 7$   
Dakle, funkcija treba da vrati broj 1 za ovaj niz **A**.  
Vremenska složenost funkcije treba da bude  $O(N)$ .
2. U fajlu **Tekst.txt** je dat proizvoljan tekst. Napisati program koji otvara ovaj fajl i određuje sljedeću statistiku fajla:
  - broj karaktera u fajlu,
  - broj riječi u fajlu (riječ predstavlja niz od jednog ili više uzastopnih slova ili cifara),
  - broj linija u fajlu,
  - najdužu riječ u fajlu i
  - liniju koja sadrži najviše riječi.Sve dobijene podatke odštampati u okviru glavnog programa. Pretpostaviti da je maksimalna dužina linije 200 karaktera.
3. Kreirati tekstualni fajl **Histogram.txt** koji sadrži proizvoljan tekst. Napisati program koji otvara ovaj tekstualni fajl, broji koliko puta se pojavljuje koje slovo u fajlu (ne pravi se razlika između malih i velikih slova). Broj ponavljanja pojedinih slova prikazati u obliku stupca visine jednake broju pojavljivanja slova. Stupce prikazati karakterom '#', a na y-osi prikazati broj pojavljivanja karaktera.

**Primjer:** Ukoliko fajl Histogram.txt sadrži tekst "**Za zivot je potrebno malo. Za nesrecan zivot trazi se mnogo vise.**", program treba da prikaže sliku ispod.

```
7|          #
6|   #          #
5|#  #          #          #
4|#  #  #  ##   #          #
3|#  #  #  ##  ### #  #
2|#  #  #  ###  ### #  #
1|### # # ## ##### ### #  #
-----
|abcdefghijklmnopqrstuvwxy
```