**Laboratorijske vježbe 6**

1. Kreirati funkciju **sortiraj** koja vrši sortiranje elemenata niza u rastući redosljed. Sortiranje se vrši tako što se prođe kroz cio niz i pronađe najmanji element, koji se zatim smješta na prvu poziciju. Tada se pretražuje ostatak niza (bez prvog elementa) kako bi se pronašao drugi najmanji element koji se smješta na drugu poziciju u nizu. Postupak se zatim ponavlja za sve elemente osim prva dva koji su već sortirani, itd.

2. Napisati funkciju **ponavljanje** koja za argumente ima string **S** i cio broj **N**. Funkcija treba da vrati posljednji karakter stringa S koji se pojavljuje tačno N puta. Ukoliko takav karakter ne postoji, funkcija vraća terminacioni karakter. U funkciji main() pozvati napisanu funkciju i štampati karakter koji se ponavlja ili odgovarajuće obavještenje ukoliko nema takvog karaktera.

   **Primjer:** Ako funkciji proslijedimo string S="abbbc#DcDFc1D3*" i broj N=3, funkcija treba da vrati karakter 'D'.

3. Napisati funkciju **poredjenje** koja za argument ima string **S** koji predstavlja nejednakost zadatu u formatu **AopB**, pri čemu su A i B cijeli brojevi, a op može biti '==' ili '**!=**'. Funkcija treba da provjeri da li je izraz zadat stringom S tačan (vraća broj 1) ili ne (vraća broj 0). U funkciji main() učitati string, pozvati funkciju i štampati odgovarajuće obavještenje. Nije potrebno provjeravati ispravnost formata stringa S.
   **Primjer:** Funkcija poredjenje("34!=56") treba da vrati broj 1.

4. Napisati program koji učitava matricu cijelih brojeva **X**, dimenzija **NxN**. Na matrici je potrebno primijeniti **medijan** proceduru koji funkcioniše na sljedeći način:

   - uzeti „prozor“ (okvir) dimenzija 3x3 i proći kroz sve elemente matrice tako da prozor ne ispada van njenih dimenzija;
   - u svakom prozoru potrebno je odrediti medijan koji predstavlja srednji elemenat po vrijednosti;
   - centralni elemenat u prozoru potrebno je zamijeniti pronađenim medijanom.

   **Primjer**: Ako jedan prozor od 9 elemenata matrice, posloženih u niz, izgleda ovako [5,1,4,7,8,9,2,3,6] onda je medijan tog niza broj 5 (srednji elemenat po vrijednosti).

5. Napisati program koji učitava niz cijelih brojeva **X**, dužine **N**, koji vrši sortiranje tog niza korišćenjem funkcije **shell**, koja implementira **Shell sort** algoritam sortiranja.

   Kod Shell sort algoritma, dozvoljena je zamjena nesusjednih elemenata, za razliku od Insertion sort i Bubble sort algoritma. Ideja je da se elementi urede u niz, tako da svaki njegov podniz koji počinje na bilo kom indeksu *k*, i sadrži svaki naredni *h*-ti element (dakle elementi na pozicijama *k*, *k*+*h*, *k*+2*h*…), bude sortiran. Za ovakav niz se kaže da je *h*-sortiran. Počinje se sa velikom vrijednošću *h*, što omogućava premještanje udaljenih elemenata u originalnom nizu, i time smanjuje broj premještanja u odnosu na slučaj kada je dozvoljena zamjena samo susjednih elemenata. Ako je niz *p*-sortiran, za *p* manje od *h*, on je i *h*-sortiran. Smanjujući *h* u svakoj iteraciji do vrijednosti 1 algoritam garantuje sortiran niz na kraju izvršavanja.

   Ispod je dat primjer izvršavanja Shell sort algoritma, sa raskoracima 5, 3 i 1:

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| input data: | 62 | 83 | 18 | 53 | 07 | 17 | 95 | 86 | 47 | 69 | 25 | 28 |
| after 5-sorting: | 17 | 28 | 18 | 47 | 07 | 25 | 83 | 86 | 53 | 69 | 62 | 95 |
| after 3-sorting: | 17 | 07 | 18 | 47 | 28 | 25 | 69 | 62 | 53 | 83 | 86 | 95 |
| after 1-sorting: | 07 | 17 | 18 | 25 | 28 | 47 | 53 | 62 | 69 | 83 | 86 | 95 |

   U prvom prolasku, sortiraju se elementi na udaljenosti 5, odnosno sortiraju se podnizovi:
   (*a1, a6, a11*), (*a2, a7, a12*), (*a3, a8*), (*a4, a9*), (*a5, a10*).
   Na primjer, podniz (*a1, a6, a11*) se od (62, 17, 25) transformiše u (17, 25, 62).

   U sljedećem prolazu vrši se 3-sortiranje na podnizovima (*a1, a4, a7, a10*), (*a2, a5, a8, a11*), (*a3, a6, a9, a12*). U posljednjem prolazu 1-sortiranje je uobičajeno sortiranje umetanjem nad cijelim nizom (*a1,..., a12*).