

13 Вежби

1. Да се напише програма која за низа од N внесени цели броеви потребно е да ја одреди средната вредност од минималниот и максималниот елемент од низата.

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main(int argc, char *argv[])
{
    int A[MAX],i,n,max,min;
    float avg;
    printf("Vnesi dolzina na nizata:");
    scanf("%d",&n);
    for(i=0;i<n;i++){
        printf("A[%d]=",i);
        scanf("%d",&A[i]);
    }
    min=max=A[0];
    for(i=1;i<n;i++)
    {
        if(min>A[i])min=A[i];
        if(max<A[i])max=A[i];
    }
    avg = max+min/2.0;
    printf("min element e %d, max element e %d, sredna vrednost e %.2f",min,max,avg);
    system("PAUSE");
    return 0;
}
```

2. Да се напише програма која добива низа од N реални броеви, а како резултат го враќа елементот кој е “најблиску” до средната вредност на елементите од низата.

Пример: 64.1 58.0 38.2 9.0 60.5 18.5 = средна вредност е 41.38 = најблиску е бројот 38.2

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main(int argc, char *argv[])
{ int i,n;
    float A[MAX],razlika,element;
    float avg;
    printf("Vnesi dolzina na nizata:");
    scanf("%d",&n);
    for(i=0;i<n;i++){
        printf("A[%d]=",i);
        scanf("%f",&A[i]);
    }
```

```

}
avg = 0.0;
for(i=0;i<n;i++){
    avg+=A[i];
}
avg/=n;
razlika = abs(A[0]-avg);
element = A[0];

for(i=1;i<n;i++){
    if(abs(avg-A[i])<razlika)
    {
        razlika = abs(avg-A[i]);
        element=A[i];
    }
}
printf("sredna vrednost e %.2f, najbliski do razlikata e elementot %.2f so rastojanie
%.2f",avg,element,razlika);
system("PAUSE");
return 0;
}

```

3. Да се напише програма која за низа од N броеви ќе проверува дали парот кој се наоѓа еден до друг е прост број.

Пример: 17402181137 = 17 е прост број, 40 не е прост број, 21 не е прост број, 81 не е прост број, 13 е прост број, 7 е прост број.

НАПОМЕНА: низата е составена од цифри од 0 до 9

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int e_prost(int el);
int main(int argc, char *argv[])
{
    int i,n,nov,A[MAX];
    printf("Vnesi dolzina na nizata:");
    scanf("%d",&n);
    for(i=0;i<n;i++){
        printf("A[%d]= ",i);
        scanf("%d",&A[i]);
    }
    for(i=0;i<n-1;i+=2){
        nov=A[i]*10 + A[i+1];

        if(e_prost(nov))
            printf("brojot %d e prost\n",nov);
        else
            printf("brojot %d NE e prost\n",nov);
    }
    if((n%2)&&(e_prost(A[n-1]))) //ако има непарен број на елементи во низата

```

```

    printf("brojot %d e prost\n",A[n-1]);

    system("PAUSE");
    return 0;
}
int e_prost(int el){
    int i;
    if((el==1)||(el==2)) return 0;
    if(el==3) return 1;
    for (i=2;i*i<=el;i++){
        if(el%i==0)return 0;
    }
    return 1;
}

```

4. Да се напише програма која чита две целобројни матрици со димензија $m \times n$, а потоа печати колку колони од првата матрица се наоѓаат во втората матрица.

Пример: Доколку корисникот ги внесе следните две матрици програмата треба да испечати 1, бидејќи само една колона од првата матрица е содржана во втората матрица. Во конкретниот случај втората колона од првата матрица е еднаква на третата колона од втората матрица.

$$\begin{bmatrix} 1 & 2 & 4 & 6 \\ 2 & 3 & 4 & 5 \\ 5 & 1 & 3 & 1 \\ 1 & 3 & 4 & 4 \end{bmatrix} \quad \begin{bmatrix} 2 & 2 & 2 & 4 \\ 3 & 2 & 3 & 5 \\ 5 & 2 & 1 & 3 \\ 1 & 1 & 3 & 4 \end{bmatrix}$$

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main(int argc, char *argv[])
{
    int i,j,m,pret,brojac,n,k,A[MAX][MAX],B[MAX][MAX];
    printf("Vnesi dimenzii na matrica A");
    scanf("%d%d",&m,&n);
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            printf("A[%d][%d]= ",i,j);
            scanf("%d",&A[i][j]);
        }
    }
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            printf("B[%d][%d]= ",i,j);
            scanf("%d",&B[i][j]);
        }
    }
    brojac = 0;
    for(j=0;j<n;j++){ // за секоја колона
        for(k=0;k<n;k++){
            pret=1;

```

```

    for(i=0;i<m;i++){
        if(A[i][j]!=B[i][k])
            {pret=0;break;}
    }
    if(pret)brojac++;
}
}
printf("ima %d koloni od prvata matrica vo vtorata matrica", brojac);
system("PAUSE");
return 0;
}

```

5. Да се напише програма која ќе прочита две целобројни квадратни матрици со димензија $n \times n$, а потоа ќе формира трета матрица која претставува производ на првите две матрици. Програмата потоа проверува дали добиената матрица е единечна (единечна матрица е матрица кај која сите елементи од главната дијагонала се 1, а сите останати се 0.) На крај програмата треба да ја прикаже резултантната матрица како и добиеното решение за тоа дали матрицата е единечна или не.

Пример: множење на две 2×2 матрици

$$1. \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{bmatrix}$$

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main(int argc, char *argv[])
{
    int i,j,m,l,pret,brojac,n,k,A[MAX][MAX],B[MAX][MAX],C[MAX][MAX];
    printf("Vnesi dimenzii na matrica A"); //mxn
    scanf("%d%d",&m,&n);
    printf("Vnesi ja vtorata dimenzija na matricata B"); //nxl
    scanf("%d",&l);
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            printf("A[%d][%d]= ",i,j);
            scanf("%d",&A[i][j]);
        }
    }
    for(i=0;i<n;i++){
        for(j=0;j<l;j++){
            printf("B[%d][%d]= ",i,j);
            scanf("%d",&B[i][j]);
        }
    }

    for(i=0;i<m;i++){ //mnozenje na matricata
        for(j=0;j<l;j++){
            C[i][j]=0;

```

```

        for(k=0;k<n;k++){
            C[i][j]+=A[i][k]*B[k][j];
        }
    }
}
for(i=0;i<m;i++){ //pecatenje na matricata
    for(j=0;j<l;j++){
        printf("%d\t",C[i][j]);
    }
    printf("\n");
}

pret = 1;
for(i=0;i<m;i++) //proverka dali e edinecna matricata
{
    for(j=0;j<l;j++){
        if((i==j)&&(C[i][j]!=1))
            {pret=0;break;}
        else if((i!=j)&&(C[i][j]!=0))
            {pret=0;break;}
    }
    if(!pret) break;
}
if(pret)
    printf("Matricite go zadovoluvaat uslovot");
else
    printf("Ne e zadovolen uslovot");

system("PAUSE");
return 0;
}

```

6. Да се напише програма која за секоја редица од дадена матрица ќе го најде максималниот и минималниот елемент и ќе им ги замени позициите.

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main(int argc, char *argv[])
{ int i,j,m,n,A[MAX][MAX];
  int max_element,min_element,max_indeks,min_indeks;
  printf("Vnesi dimenzii na matrica A");
  scanf("%d%d",&m,&n);
  for(i=0;i<m;i++){
    for(j=0;j<n;j++){
        printf("A[%d][%d]= ",i,j);
        scanf("%d",&A[i][j]);
    }
  }
  for(i=0;i<m;i++){ //ispitaj za sekoja redica

```

```

min_element = max_element = A[i][0]; //postavi inicijalen min i max
min_indeks=max_indeks=0;
for(j=1;j<n;j++){ //izmini gi ostanatite koloni
    if(A[i][j]<min_element){
        min_element=A[i][j];
        min_indeks=j;
    }
    if(A[i][j]>max_element){
        max_element=A[i][j];
        max_indeks=j;
    }
}
A[i][max_indeks]=min_element; //zamena na mestata
A[i][min_indeks]=max_element; //zamena na mestata
}

for(i=0;i<m;i++){ //pecatenje na matricata
for(j=0;j<n;j++){
    printf("%d\t",A[i][j]);
}
printf("\n");
}

system("PAUSE");
return 0;
}

```

7. Да се напише функција која на влез ќе добие низа од цели броеви. Функцијата како резултат враќа 1 доколку е строго растечка. Доколку низата не е строго растечка истата да се трансформира во строго растечка и како резултат да се врати 2. Во рамки на функцијата забрането е користење на средни загради []. Да се напише главна програма која ќе ја провери работата на функцијата.

НАПОМЕНА: Трансформацијата на низата во строго растечка се прави на следниот начин:

1. Се сортира во растечки редослед
2. Се отстрануваат дупликатите

Пример:

A=[1,2,3,5,10,20] => nizata e predefinirano strogo rastecka
A=[2,3,1,7,6] => nizata ne e strogo rastecka. A=[1,2,3,6,7]
A=[1,2,2,3,5,7,10]=> nizata ne e strogo rastecka. A=[1,2,3,5,7,10]

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 100
void sortiraj(int *a,int n);
void otfrli_duplikati(int *niza, int n, int *new);
void swap(int *x, int *y);
int funkcija(int *niza,int *n){
    int i,vrati;

```

```
vrati = 1;
for(i=0;i<*n-1;i++){
    if(*(niza +i)<*(niza+i+1))continue;
    else
        {   vrati=2;
            break;
        }
}
if(vrati==1) return 1;
sortiraj(niza,*n);
otfrli_duplikati(niza,*n,n);
return 2;
}
void sortiraj(int *a,int n){
    int i,j;
    for(i=0;i<n;i++){
        for(j=i+1;j<n;j++){
            if(a[i]> a[j])
                swap(&a[i],&a[j]);
        }
    }
}
void otfrli_duplikati(int niza[], int n, int *new){
    int i,j,brojac=0;
    for(i=0; i<n; i++)
    {
        for(j=i+1; j<n; j++)
        {
            if(niza[i] == niza[j])
            {
                for(k=j; k<n; k++)
                {
                    niza[k]=niza[k+1];
                }
                j=i;
                n=(n-1);
            }
        }
    }
    *new = n;
}
void swap(int *x,int *y){
    int t;
    t=*x;
    *x=*y;
    *y=t;
}
int main(int argc, char *argv[])
```

```

{
    int A[MAX],i,n,new;

    printf("Vnesi golemina na niza");
    scanf("%d",&n);
    for(i=0;i<n;i++){
        printf("A[%d]=",i);
        scanf("%d",&A[i]);
    }
    new = funkcija(A,&n);
    if(new==1) printf("Funkcijata e strogo rastecka");
    else
    {
        printf("funkcijata ne e strogo rastecka. Posle transformacijata    nizata e:\n");
        for(i=0;i<n;i++) printf("%d",A[i]);
    }
    system("PAUSE");
    return 0;
}

```

8. Да се напише функција која на влез добива низа и X целобројна променлива. Ако X е 1, тогаш низата да се сортира во растечки редослед, а доколку X е 2 тогаш да се сортира во опаѓачки. Како алгоритам за сортирање може да се користи bubble sort.

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 100
void swap(int *i,int *j)
{
    int temp;
    temp=*i;
    *i=*j;
    *j=temp;
}
void bubblesort(int a[],int n,int x)
{
    int i,j;
    if(x==1)
    {
        for (i=0; i<n; i++)
            for (j=0; j<n-i-1; j++)
                if (a[j]>a[j+1])
                    swap(&a[j],&a[j+1]);
    }
    else if(x==2)
    {
        for (i=0; i<n; i++)
            for (j=0; j<n-i-1; j++)
                if (a[j]>a[j+1])

```



```

        swap(&a[j],&a[j+1]);
    }

}

int main(int argc, char *argv[])
{ int i,a[MAX],n,x;
  printf("vnesi ja dolzinata na nizata\n");
  scanf("%d",&n);
  printf("vnesi ja nasokata na sortiranje\n");
  scanf("%d",&x);
  printf("vnesi ja nizata\n");
  for(i=0;i<n;i++)
  scanf("%d",&a[i]);
  bubblesort(a,n,x);
  for(i=0;i<n;i++)
  printf("%d\t",a[i]);

  system("PAUSE");
  return 0;
}

```

9. Да се напише програма која дадена квадратна матрица ќе ја измени на тој начин што елементите од споредната дијагонала ќе имаат вредност еднаква на сумата на црните полиња (обележани на сликата со правоаголник). Останатите полиња ќе останат непроменети. На крај да се отпечати ново добиената матрица. Максималната димензија на матрицата е 10x10.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 7 & 8 & 9 & 10 & 9 \\ 8 & 7 & 6 & 5 & 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 & 75 \\ 1 & 2 & 3 & 75 & 5 \\ 6 & 7 & 75 & 9 & 10 \\ 7 & 75 & 9 & 10 & 9 \\ 75 & 7 & 6 & 5 & 4 \end{bmatrix}$$

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main(int argc, char *argv[])
{ int A[MAX][MAX],i,j,n,suma;
  printf("Vnesi ja goleminata na matricata\n");
  scanf("%d",&n);
  for(i=0;i<n;i++){
  for(j=0;j<n;j++){
    printf("A[%d][%d]= ",i,j);
    scanf("%d",&A[i][j]);
  }
}
  suma=0;
  for(i=0;i<n;i++){
  for(j=0;j<n;j++){

```

```

    if(i%2){
        if(j%2)suma+=A[i][j]; //i и j се непарни позиции
    }
    else {
        if(j%2==0) suma+=A[i][j]; //i и j се парни позиции
    }
}
}
for(i=0;i<n;i++){
for(j=0;j<n;j++){
    if((i+j+1)==n)
        A[i][j]=suma;
    }
}
for(i=0;i<n;i++){
for(j=0;j<n;j++){
    printf("%d\t",A[i][j]);
    }
printf("\n");
}
system("PAUSE");
return 0;
}

```

10. Да се напише програма која ќе прими како аргументи три низи од знаци. Функцијата треба секоја појава на првата низа во втората низа да ја замени со третата низа (имплементација на search-and-replace алгоритам).

Пример: прва низа: prv

втора низа: ova е prv primer за prva niza.

трета низа: vtor

втора низа променета: ova е vtor primer за vtor niza

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100

int main(int argc, char *argv[])
{ char prva[MAX], vtor[MAX], tretata[MAX];
  int i,j,dolzina_prv,dolzina_vtor,dolzina_tret,prov,razlika;
  printf("vnesi ja prvata niza\n");
  gets(prva);
  printf("vnesi ja vtorata niza \n");
  gets(vtor);
  printf("vnesi ja tretata niza \n");
  gets(tretata);
  printf("Zamenuvam sekoe pojavuvanje na prvata niza vo vtorata niza
so tretata niza\n");
  dolzina_vtor = strlen(vtor);
  dolzina_tret = strlen(tretata);
  dolzina_prv = strlen(prva);

```

```

for(i=0;i<dolzina_vtor;i++){
    prov = 1;
    for(j=0;j<dolzina_prv;j++){
        if(*(vtora+i+j)!=*(prva+j)){prov=0;break;}
    }
    if((prov)&&((dolzina_vtor+dolzina_tret-dolzina_prv)<=MAX)){
        razlika = dolzina_tret-dolzina_prv;
        if(razlika>0){
            for(j=dolzina_vtor-1;j>=i+dolzina_prv;j--)
                vtora[j+razlika]=vtora[j]; //dodadi prostor
            dolzina_vtor+=razlika;
        }
        if(razlika<0){
            for(j=i+dolzina_tret;j<dolzina_vtor+razlika;j++)
                *(vtora+j)=*(vtora+j-razlika); //oslobodi prostor
            dolzina_vtor+=razlika;
        }
        for(j=0;j<dolzina_tret;j++)
            *(vtora+i+j)=*(treta+j);
        vtora[dolzina_vtor]='\0';
    }
}
puts(vtora);
system("PAUSE");
return 0;
}

```

11. Да се напише програма која ќе прочита датотека со име vlez.txt. Програмата треба да го испечати: бројот на линии, бројот на зборови, бројот на знаци (без празните места) и бројот на празни места во датотеката.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100
int AlphaNum(char c){
    int mala,golema,cifra;
    mala = (c>='a') && (c<='z');
    golema = (c>='A') && (c<='Z');
    cifra = (c>='0') && (c<='9');
    return (mala||golema||cifra);
}
int main(int argc, char *argv[])
{
    int i,zborovi,prazni,vozbor,linii,znaci;
    char c;
    FILE *fd;
    if((fd=fopen("vlez.txt","r"))==NULL){

```

```
printf("ne moze da se otvori datotekata vlez.txt");
system("PAUSE");
return -1;
}
linii=znaci=vozbor=prazni=zborovi=0;
while((c=fgetc(fd))!=EOF)
{
    znaci++;
    if(c=='\n'){
        linii++;
        if(vozbor){zborovi++;vozbor=0;}
        continue;
    }
    if(AlphaNum(c)){
        if(vozbor==0) vozbor =1;
    }
    else {
        if(vozbor) zborovi++;
        if(isspace(c))prazni++;
        vozbor=0;
    }
}
if(vozbor)zborovi++;

printf("ima %d linii, %d zborovi, %d znaci bez prazno mesto i %d prazni
mesta",linii,zborovi,znaci-prazni,prazni);
system("PAUSE");
return 0;
}
```