

Алгоритми

1. Да се напише програма во која од тастатура се внесуваат низа од целобројни вредности и број на елементи во низата. Потоа од тастатура да се внесе вредност на елемент по што да се изврши пребарување на внесената низа и на излез да се врати индексот на позицијата на која е најден дополнително внесениот елемент. За пребарувањето да се напишат посебни функции и тоа за линеарно и бинарно пребарување на вектори. Во програмата од тастатура да се внесе со која функција да се прави пребарувањето.

Решение:

```
#include <stdio.h>
#define Max 100

void citajNiza(int x[], int n) {
    int i;

    for (i=0; i<n; i++)
        scanf("%d", &x[i]);
}

void pecatiNiza(int x[], int n) {
    int i;

    for (i=0; i<n; i++)
        printf("%d ", x[i]);

    printf("\n");
}

int linearBaraj(int a[], int n, int key) {
    int i, lb;

    lb=-1;
    i=0;

    while ((lb== -1) && (i<n)) {
        if (a[i]==key) {
            lb=i;
        }
        i++;
    }

    return lb;
}

int linearBaraj1(int a[], int n, int key) {
    int i;

    i=0;
    a[n]=key; // postavuvame cuvar

    while (a[i] != key)
        i++;

    if (i==n)
        return -1;
}
```

```

        else
            return i;
    }

int binarnoBaraj(int a[], int n, int key){
    int pocetok,kraj,sredina, najden;

    pocetok=0;
    kraj=n-1;
    najden=0;
    while (!najden && (pocetok <= kraj)){
        sredina=(pocetok+kraj)/2;
        if (key < a[sredina])
            kraj=sredina-1;
        else
            if (key > a[sredina])
                pocetok=sredina+1;
            else // key==a[sredina]
                najden=1;
    }

    if (najden)
        return sredina;
    else
        return -1;
}

int main(){
    int a[Max];
    int n,k,l, tip;
    printf("Kolkava niza? \n"); scanf("%d",&n);
    citajNiza(a,n);
    printf("Vlezna niza : \n");
    pecatiNiza(a,n);
    printf("Sto da baram? "); scanf("%d",&k);

    printf("Kako da prebaruvam (0 za linearno, 1 za binarno)? ");
    scanf("%d",&tip);

    if (tip==0){
        l=linearBaraj(a,n,k);
        // l=linearBarajl(a,n,k);
    }
    else
        l=binarnoBaraj(a,n,k); // nizata treba da e sortirana

    if (l==-1)
        printf("Baraniot element ne e najden.\n");
    else
        printf("Baraniot element e najden na pozicija%d\n",l);

    return 0;
}

```

2. Да се напише програма во која ќе се внесат низа од целобројни вредности и бројот на елементи во низата. Како дел од програмата, да се напишат функции за два различни алгоритми за сортирање: SimpleSort, BubbleSort и притоа и притоа сортирањето да се во растечки редослед. Во програмата, од тастатура да се внесе тип на сортирање кој го

преферира корисникот и потоа типот на сортирање да се примени на влезната низа. Да се испечати сортираната низа.

Решение:

```
#include <stdio.h>
#define Max 100

void citajNiza(int x[], int n) {
    int i;

    for (i=0; i<n; i++)
        scanf("%d", &x[i]);
}

void pecatiNiza(int x[], int n) {
    int i;

    for (i=0; i<n; i++)
        printf("%d ", x[i]);
    printf("\n");
}

void swap(int *i, int *j) {
    int temp;

    temp=*i;
    *i=*j;
    *j=temp;
}

void simpleSort(int a[], int n) {
    int i, j;

    for (i=0; i<n-1; i++)
        for (j=i+1; j<n; j++)
            if (a[i]>a[j])
                swap(&a[i], &a[j]);
}

void bubbleSort(int a[], int n) {
    int i, j;

    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 a[Max], b[Max];
    int n, k, l, c;

    printf("Kolkava niza? \n"); scanf("%d", &n);
    citajNiza(a, n);
    printf("Input : \n");
    pecatiNiza(a, n);
    printf("po koj metod?\n");
    scanf("%d", &c);

    switch (c) {
        case 1: simpleSort(a, n);
```

```

        break;
    case 2: bubbleSort(a, n);
        break;
    default: printf("nema takov\n");
}

printf("Output : \n");
pecatiNiza(a, n);

return 0;
}

```

Kolkava niza? 10

73 65 52 24 83 17 35 96 41 9

===== SimpleSort =====

Input : 73 65 52 24 83 17 35 96 41 9

Pass 1: 65 73 52 24 83 17 35 96 41 9

Pass 1: 52 73 65 24 83 17 35 96 41 9

Pass 1: 24 73 65 52 83 17 35 96 41 9

Pass 1: 17 73 65 52 83 24 35 96 41 9

Pass 1: 9 73 65 52 83 24 35 96 41 17

Pass 2: 9 65 73 52 83 24 35 96 41 17

Pass 2: 9 52 73 65 83 24 35 96 41 17

Pass 2: 9 24 73 65 83 52 35 96 41 17

Pass 2: 9 17 73 65 83 52 35 96 41 24

Pass 3: 9 17 65 73 83 52 35 96 41 24

Pass 3: 9 17 52 73 83 65 35 96 41 24

Pass 3: 9 17 35 73 83 65 52 96 41 24

Pass 3: 9 17 24 73 83 65 52 96 41 35

Pass 4: 9 17 24 65 83 73 52 96 41 35

Pass 4: 9 17 24 52 83 73 65 96 41 35

Pass 4: 9 17 24 41 83 73 65 96 52 35

Pass 4: 9 17 24 35 83 73 65 96 52 41

Pass 5: 9 17 24 35 73 83 65 96 52 41

Pass 5: 9 17 24 35 65 83 73 96 52 41

Pass 5: 9 17 24 35 52 83 73 96 65 41

Pass 5: 9 17 24 35 41 83 73 96 65 52

Pass 6: 9 17 24 35 41 73 83 96 65 52

Pass 6: 9 17 24 35 41 65 83 96 73 52

Pass 6: 9 17 24 35 41 52 83 96 73 65

Pass 7: 9 17 24 35 41 52 73 96 83 65

Pass 7: 9 17 24 35 41 52 65 96 83 73

Pass 8: 9 17 24 35 41 52 65 83 96 73

Pass 8: 9 17 24 35 41 52 65 73 96 83

Pass 9: 9 17 24 35 41 52 65 73 83 96

Output : 9 17 24 35 41 52 65 73 83 96

===== BubbleSort =====

Input : 73 65 52 24 83 17 35 96 41 9

Pass 1: 65 73 52 24 83 17 35 96 41 9

Pass 1: 65 52 73 24 83 17 35 96 41 9

Pass 1: 65 52 24 73 83 17 35 96 41 9

Pass 1: 65 52 24 73 17 83 35 96 41 9

Pass 1: 65 52 24 73 17 35 83 96 41 9

Pass 1: 65 52 24 73 17 35 83 41 96 9
 Pass 1: 65 52 24 73 17 35 83 41 9 96
 Pass 2: 52 65 24 73 17 35 83 41 9 96
 Pass 2: 52 24 65 73 17 35 83 41 9 96
 Pass 2: 52 24 65 17 73 35 83 41 9 96
 Pass 2: 52 24 65 17 35 73 83 41 9 96
 Pass 2: 52 24 65 17 35 73 41 83 9 96
 Pass 2: 52 24 65 17 35 73 41 9 83 96
 Pass 3: 24 52 65 17 35 73 41 9 83 96
 Pass 3: 24 52 17 65 35 73 41 9 83 96
 Pass 3: 24 52 17 35 65 73 41 9 83 96
 Pass 3: 24 52 17 35 65 41 73 9 83 96
 Pass 3: 24 52 17 35 65 41 9 73 83 96
 Pass 4: 24 17 52 35 65 41 9 73 83 96
 Pass 4: 24 17 35 52 65 41 9 73 83 96
 Pass 4: 24 17 35 52 41 65 9 73 83 96
 Pass 4: 24 17 35 52 41 9 65 73 83 96
 Pass 5: 17 24 35 52 41 9 65 73 83 96
 Pass 5: 17 24 35 41 52 9 65 73 83 96
 Pass 5: 17 24 35 41 9 52 65 73 83 96
 Pass 6: 17 24 35 9 41 52 65 73 83 96
 Pass 7: 17 24 9 35 41 52 65 73 83 96
 Pass 8: 17 9 24 35 41 52 65 73 83 96
 Pass 9: 9 17 24 35 41 52 65 73 83 96
 Output : 9 17 24 35 41 52 65 73 83 96

3. Да се напише програма којашто влезната матрица A со димензии $m \times n$ ќе ја преуреди така што колоните да и се во растечки редослед според максималниот елемент на колоната.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 1 \\ 8 & 6 & 6 & 3 & 3 \\ 0 & 3 & 9 & 2 & 7 \\ 8 & 6 & 9 & 4 & 7 \end{bmatrix} \Rightarrow \begin{bmatrix} 4 & 2 & 1 & 1 & 3 \\ 3 & 6 & 3 & 8 & 6 \\ 2 & 3 & 7 & 0 & 9 \\ 4 & 6 & 7 & 8 & 9 \end{bmatrix}$$

Решение:

```
#include <stdio.h>
#define Mx 100

int main()
{
    int m,n,i,j,k;
    float a[Mx][Mx], max[Mx], t;

    printf("Vnesi gi димензиите на матрицата\n");
    scanf("%d %d", &m, &n);
    printf("Vnesi ja матрицата...\n");
    for (i=0; i<m; i++)
        for (j=0; j<n; j++)
    {
        printf("a[%d, %d]=", i, j);
        scanf("%f", &a[i][j]);
    }
}
```

```

for (j=0; j<n; j++)
{
    max[j]=a[0][j];
    for (i=1; i<m; i++)
        if (max[j]<a[i][j])
            max[j]=a[i][j];
}

for (i=0; i<n; i++)
    for (j=i+1; j<n; j++)
        if (max[i] > max[j])
        {
            t=max[i]; max[i]=max[j]; max[j]=t;
            for (k=0; k<m; k++)
            {
                t=a[k][i];
                a[k][i]=a[k][j];
                a[k][j]=t;
            }
        }

for (j=0; j<n; j++)
{
    printf("%5f\t", max[j]);
}

for (i=0; i<m; i++)
{
    printf("\n");
    for (j=0; j<n; j++)
        printf("%5.2f\t", a[i][j]);
}
return 0;
}

```

4. Дадени се две низи со податоци за тежина и висина на неколку луѓе (бројот на луѓе се внесува од тастатура). Да се провери дали луѓето можат да формираат кула од луѓе, т.е. секој човек да биде понизок и полесен од оној под него.

Забелешка: Двете низи треба да се сортираат во **растечки** редослед по **висина**. По сортирањето, двете низи **истовремено** се изминуваат и се проверува дали соседните членови го задоволуваат условот (секој човек да биде и полесен и понизок од оној после него).

Пример:

За внесени вредности во облик **(тежина,висина)** = **(55,155), (50,150), (80,200), (69,170)** сортираниите вредности по висина се: **(50,150), (55,155), (69,170),(80,200)** и го задоволуваат условот, **но**, ако се внесат вредности **(тежина,висина)** = **(55,155), (50,160), (80,200), (69,170),** сортираниите вредности се: **(55,155), (50,160), (69,170),(80,200)** и **не** го задоволуваат условот

За дома: ако сортираната низа не го задоволува условот, од неа да се исфрлат сите членови кои го нарушуваат истиот, а остатокот од низата да го задоволува условот

Решение:

```

#include <stdio.h>
#define MAX 10

void simpleSortVisina( int v[], int t[], int n){
    int i,j;
    int pom;
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
            if(v[i]>v[j]) /* gi sortirame po visina, no istovremeno gi
menuvame i vrednostite vo listata za tezhini na istite pozicii */
    {
        pom=v[i];
        v[i]=v[j];
        v[j]=pom;
        pom=t[i];
        t[i]=t[j];
        t[j]=pom;
    }
}

int main(){
    char *niza[MAX]={"prviot", "vtoriot", "tretiot", "chetvrtoit", "pettiot",
"shestiot", "sedmiot", "osmiot", "devettiot", "desettiot"};
    char *niza2[MAX]={"Prv", "Vtor", "Tret", "Chetvrt", "Petti", "Shesti",
"Sedmi", "Osmi", "Devetti", "Desetti"};
    int visina[MAX], tezhina[MAX];
    int brLugje,i;

    printf("Vnesete broj na lugje: \n");
    scanf("%d",&brLugje);
    printf("Vnesete gi podatocite za tezhina: \n");
    for(i=0;i<brLugje;i++){
        printf("Podatok za %s chovek:",niza[i]);
        scanf("%d",&tezhina[i]);
    }
    printf("Vnesete gi podatocite za visina: \n");
    for(i=0;i<brLugje;i++){
        printf("Podatok za %s chovek:",niza[i]);
        scanf("%d",&visina[i]);
    }
/*sega kje si sortirame site clenovi od dvete nizi istovremeno po visina*/
    simpleSortVisina(visina,tezhina,brLugje);

    printf("Vaka izgledaat chlenovite sortirani po visina:\n");
    for(i=0;i<brLugje;i++){
        printf("%-10s: ",niza2[i]);
        printf("tezhina = %d, visina = %d\n",tezhina[i],visina[i]);
    }

    for(i=0;i<brLugje-1;i++){
        if (tezhina[i]>tezhina[i+1] || visina[i]>visina[i+1])
            break;
    }
    if(i==brLugje-1)
        printf("Mozhe da formiraat kula od lugje\n");
    else
        printf("Ne mozhe da formiraat kula od lugje\n")

    return 0;
}

```

5. Да се напише програма во која од тастатура се внесуваат вредности за индекси (вкупниот број на индекси исто така се внесува од тастатура). Индексите се внесуваат во формат број_на_индекс/година_на_индекс. Програмата треба да ги сортира индексите во растечки редослед според годината на индексот. Индексите од иста година се сортираат во растечки редослед според бројот на индексот.

Пример: ако се внесат вредности за индекси во облик

број/година = 157/2009, 20/2009, 50/2010, 300/2005, 22/2010, 250/2005, 20/2010

по сортирањето се добива

број/година = 250/2005, 300/2005, 20/2009, 157/2009, 20/2010, 22/2010, 50/2010

Решение:

```
#include <stdio.h>
#define MAX 30

void swap(int *a, int *b) {
    int pom;
    pom=*a;
    *a=*b;
    *b=pom;
}

void bubbleSortGodinaIndeks( int g[],int b[], int n){
    int i,j;
    int pom;
    for(i=n-1;i>0;i--)
        for(j=0;j<i;j++) {
            if(g[j]>g[j+1]){
                swap(&g[j],&g[j+1]);
                swap(&b[j],&b[j+1]);
            }
            else if(g[j]==g[j+1])
            {
                if(b[j]>b[j+1])
                {
                    swap(&g[j],&g[j+1]);
                    swap(&b[j],&b[j+1]);
                }
            }
        }
}

int main(){
    int broj[MAX], godina[MAX];
    int brIndeksi,i;

    printf("Vnesete broj na indeksi za sortiranje: \n");
    scanf("%d",&brIndeksi);
    printf("Vnesete gi vrednostite za indeksite: \n");
    for(i=0;i<brIndeksi;i++){
        printf("Broj: ");
        scanf("%d",&broj[i]);
        printf("Godina: ");
        scanf("%d",&godina[i]);
    }
    /* sega kje si sortirame site clenovi od dvete nizi istovremeno, ako
    godinitate se razlichni sortirame samo po godina */
}
```

```
/* ako godinite se isti, prodolzhevame da sortirame i po broj na indeks */

bubbleSortGodinaIndeks(godina,broj,brIndeksi);

printf("Sortiranite indeksi se slednive:\n\n");
printf("broj/godina:\n");

for(i=0;i<brIndeksi;i++)
    printf("%d/%d\n",broj[i],godina[i]);

return 0;
}
```