



REPUBLIKA E SHQIPËRISË
MINISTRIA E ARSIMIT
DHE SPORTIT
AGJENCIA KOMBËTARE E PROVIMEVE

OLIMPIADA KOMBËTARE E INFORMATIKËS

Viti mësimor 2016-2017

Faza e dytë

Klasa 12

Një mënyrë zgjidhje për ushtrimin 1 :

```
#include <stdio.h>
#define SIZE 5
void get_value(int arr[]);
void print_value(int arr[], int n);
void function_sort(int arr[]);
int find_intersection(int array1[], int array2[], int intersection_array[]);
int find_union(int array1[], int array2[], int union_array[]);

int main()
{
    int array1[SIZE], array2[SIZE], intersection_array[SIZE], union_array[SIZE*2];
    int num_elements;
    //Mbush vektorin e pare
    printf("\n Jepni elementet e vektorit te pare: \n");
    get_value(array1);
    printf("\n\n Elementet e vektorit te pare: ");
    print_value(array1, SIZE);

    //Rendit vektorin e pare
    function_sort(array1);
    printf("\n\n Vektori i pare i renditur: ");
    print_value(array1, SIZE);

    //Mbush vektorin e dyte
    printf("\n\n Jepni elementet e vektorit te dyte: \n");
    get_value(array2);
    printf("\n\n Elementet e vektorit te dyte: ");
    print_value(array2, SIZE);

    //Rendit vektorin e dyte
    function_sort(array2);
    printf("\n\n Vektori i dyte i renditur: ");
    print_value(array2, SIZE);

    //Prerja
    num_elements = find_intersection(array1, array2, intersection_array);
    printf("\n\n Prerja eshte: ");
    print_value(intersection_array, num_elements);

    //Bashkimi
    num_elements = find_union(array1, array2, union_array);
    printf("\n\n Bashkimi eshte: ");
    print_value(union_array, num_elements);
}

void get_value(int arr[])
```

```

{
    int i, j;
    for (i = 0; i < SIZE; i++)
    {
        j = i + 1;
        printf("\n Jepni elementin %d: ", j);
        scanf("%d", &arr[i]);
    }
}

```

```

void print_value(int arr[], int n)
{
    int i;
    printf("{ ");
    for (i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("}");
}

```

```

void function_sort(int arr[])
{
    int i, j, temp, swapping;

    for (i = 1; i < SIZE; i++)
    {
        swapping = 0;
        for (j = 0; j < SIZE-i; j++)
        {
            if (arr[j] > arr[j+1])
            {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
                swapping = 1;
            }
        }
        if (swapping == 0)
        {
            break;
        }
    }
}

```

```

int find_intersection(int array1[], int array2[], int intersection_array[])
{
    int i = 0, j = 0, k = 0;
    while ((i < SIZE) && (j < SIZE))
    {
        if (array1[i] < array2[j])
        {
            i++;
        }
        else if (array1[i] > array2[j])
        {
            j++;
        }
        else

```

```

    {
        intersection_array[k] = array1[i];
        i++;
        j++;
        k++;
    }
}
return(k);
}

```

```
int find_union(int array1[], int array2[], int union_array[])
```

```

{
    int i = 0, j = 0, k = 0;
    while ((i < SIZE) && (j < SIZE))
    {
        if (array1[i] < array2[j])
        {
            union_array[k] = array1[i];
            i++;
            k++;
        }
        else if (array1[i] > array2[j])
        {
            union_array[k] = array2[j];
            j++;
            k++;
        }
        else
        {
            union_array[k] = array1[i];
            i++;
            j++;
            k++;
        }
    }
    if (i == SIZE)
    {
        while (j < SIZE)
        {
            union_array[k] = array2[j];
            j++;
            k++;
        }
    }
    else
    {
        while (i < SIZE)
        {
            union_array[k] = array1[i];
            i++;
            k++;
        }
    }
    return(k);
}

```

Një mënyrë zgjidhje për ushtrimin 2 :

```
#include <stdio.h>
#include <conio.h>

void towers(int, char, char, char);
int main()
{
    int num;
    printf("Jep nr e disqeve : ");
    scanf("%d", &num);
    printf("Hapat e levizjes jane: :\n");
    towers(num, 'M', 'D', 'Q');
    getch();
    return 0;
}
void towers(int num, char frompeg, char topeg, char auxpeg)
{
    if (num == 1)
    {
        printf("\n Leviz diskun 1 nga shufra %c tek shufra %c", frompeg, topeg);
        return;
    }
    towers(num - 1, frompeg, auxpeg, topeg);
    printf("\n Leviz diskun %d nga shufra %c tek shufra %c", num, frompeg, topeg);
    towers(num - 1, auxpeg, topeg, frompeg);
}
```

Një mënyrë zgjidhje për ushtrimin 3 :

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    int number[30];
    int i, n, a, j;
    printf("Jep vleren e N \n");
    scanf("%d", &n);
    printf("Fut vlerat\n");
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);
    printf("Jep pozicionin ku do te ndahet vektori \n");
    scanf("%d", &a);
    for (i = 0; i < a; ++i)
    {
        number[n] = number[0];
        for (j = 0; j < n; ++j)
        {
            number[j] = number[j + 1];
        }
    }
    printf("Vektori i ri eshte\n");
    for (i = 0; i < n; ++i)
    {
        printf("%d\n", number[i]);
    }
    getch();
}
```

Një mënyrë zgjidhje për ushtrimin 4 :

```
#include <conio.h>
#include <cstdio>
#include <iostream>
#include <string.h>
#include <cstdlib>
using namespace std;
static int p = 0;
class a
{
    char busn[5], driver[10], arrival[5], depart[5], from[10], to[10], seat[8][4][10];
public:
    void install();
    void allotment();
    void empty();
    void show();
    void avail();
    void position(int i);
}
bus[10];
void vline(char ch)
{
    for (int i=80;i>0;i--)
        cout<<ch;
}
void a::install()
{
    cout<<"Jep nr e autobusit: ";
    cin>>bus[p].busn;
    cout<<"\nJep emrin e shoferit: ";
    cin>>bus[p].driver;
    cout<<"\nOra e mberritjes: ";
    cin>>bus[p].arrival;
    cout<<"\nOra e nisjes: ";
    cin>>bus[p].depart;
    cout<<"\nNga: \t\t\t";
    cin>>bus[p].from;
    cout<<"\nNe: \t\t\t";
    cin>>bus[p].to;
    bus[p].empty();
    p++;
}
void a::allotment()
{
    int seat;
    char number[5];
    top:
    cout<<"Autobusi nr: ";
    cin>>number;
    int n;
    for(n=0;n<=p;n++)
    {
        if(strcmp(bus[n].busn, number)==0)
            break;
    }
    while(n<=p)
    {
        cout<<"\nVendi nr: ";
```

```

cin>>seat;
if(seat>32)
{
    cout<<"\nJane vetem 32 vende te mundshme ne kete autobus.";
}
else
{
    if (strcmp(bus[n].seat[seat/4][(seat%4)-1], "Bosh")==0)

        {

            cout<<"Jep emrin e pasagjerit: ";
            cin>>bus[n].seat[seat/4][(seat%4)-1];
            break;
        }

    else

        cout<<"Vendi eshte i zene.\n";
        }

    if(n>p)
    {
        cout<<"Jep nr e sakte te autobusit.\n";
        goto top;
    }

}
void a::empty()
{

for(int i=0; i<8;i++)

{

for(int j=0;j<4;j++)

{

        strcpy(bus[p].seat[i][j], "Bosh");

    }

}

}

void a::show()
{

int n;
char number[5];
cout<<"Jep nr e autobusit: ";
cin>>number;
for(n=0;n<=p;n++)
{
    if(strcmp(bus[n].busn, number)==0)
        break;
}
}

```

```

while(n<=p)

{
    vline('*');
    cout<<"Autobusi nr: \t"<<bus[n].busn
    <<"\nShoferi: \t"<<bus[n].driver<<"\t\tOra e mberritjes: \t"
    <<bus[n].arrival<<"\tOra e nisjes:"<<bus[n].depart
    <<"\nNga: \t\t"<<bus[n].from<<"\t\tNe: \t\t"<<
    bus[n].to<<"\n";
    vline('*');
    bus[0].position(n);
    int a=1;
    for (int i=0; i<8; i++)

    {

        for(int j=0;j<4;j++)

        {

            a++;
            if(strcmp(bus[n].seat[i][j],"Bosh")!=0)

                cout<<"\nVendi nr "<<(a-1)<<" eshte i rezervuar per "<<bus[n].seat[i][j]<<". ";

        }

    }

    break;
}

if(n>p)
    cout<<"Jep nr e sakte te autobusit: ";

}

void a::position(int l)
{
    int s=0;p=0;
    for (int i =0; i<8;i++)
    {
        cout<<"\n";
        for (int j = 0;j<4; j++)
        {

            s++;
            if(strcmp(bus[l].seat[i][j], "Bosh")==0)
            {
                cout.width(5);
                cout.fill(' ');
                cout<<s<<". ";
                cout.width(10);
                cout.fill(' ');
                cout<<bus[l].seat[i][j];
                p++;
            }

            else
            {

```

```

        cout.width(5);
        cout.fill(' ');
        cout<<s<<". ";
        cout.width(10);
        cout.fill(' ');
        cout<<bus[l].seat[i][j];
    }
}
}

cout<<"\n\nJane "<<p<<" vende bosh ne autobusin nr: "<<bus[l].busn;
}
void a::avail()
{
for(int n=0;n<p;n++)
{
vline('*');
cout<<"Autobusi nr: \t"<<bus[n].busn<<"\nShoferi: \t"<<bus[n].driver
<<"\t\tOra e mberritjes: \t"<<bus[n].arrival<<"\tOra e nisjes: \t"
<<bus[n].depart<<"\nNga: \t\t"<<bus[n].from<<"\t\tNe: \t\t\t"
<<bus[n].to<<"\n";
vline('*');
vline('_');
}
}
}
int main()
{
system("cls");
int w;
while(1)
{
cout<<"\n\n\n\n\n";
cout<<"\t\t\t1.Instalo\n\t\t\t"
<<"2.Rezervo\n\t\t\t"
<<"3.Trego\n\t\t\t"
<<"4.Autobuset ku mund te rezervosh. \n\t\t\t"
<<"5.Dil";
cout<<"\n\t\t\tZgjidh nje alternative:-> ";
cin>>w;
switch(w)
{
case 1: bus[p].install();
break;
case 2: bus[p].allotment();
break;
case 3: bus[0].show();
break;
case 4: bus[0].avail();
break;
case 5: exit(0);
}
}
return 0;
}

```


Një mënyrë zgjidhje për ushtrimin 5 :

```
#include <stdio.h>
#include <conio.h>
int main()
{

    static int array[10][10];

    int i, j, m, n;

    printf("Jep permasat e matrices \n");

    scanf("%d %d", &m, &n);

    printf("Jep elementet e matrices\n");

    for (i = 0; i < m; ++i)
    {

        for (j = 0; j < n; ++j)
        {

            scanf("%d", &array[i][j]);

        }

    }

    printf("Matrica fillestare eshte\n");

    for (i = 0; i < m; ++i)
    {

        for (j = 0; j < n; ++j)
        {

            printf(" %d", array[i][j]);

        }

        printf("\n");

    }

    printf("Matrica e transpozuar eshte \n");

    for (j = 0; j < n; ++j)
    {

        for (i = 0; i < m; ++i)
```

```
{  
    printf("%d", array[i][j]);  
}  
  
printf("\n");  
  
}  
getch();  
return 0;  
}
```

Kërkesa	1	2	3	4	5
Pikët	5	10	5	20	10