

Name: Enes

Surname: Erten

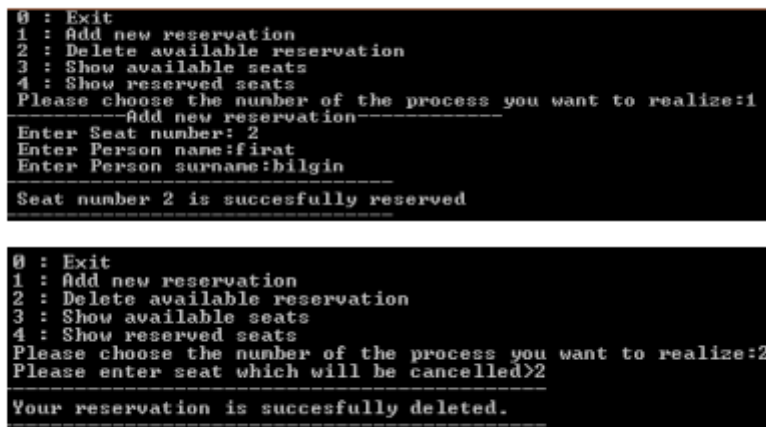
Number: 2018502036

Lab Work: 5

Section: 1(theory) 4(Lab)

**Task 1 :** Intercity Bus Firm has Izmir-Istanbul line and seats are reserved in advance for this journey. The bus has seating capacity of 25. Write a function that keeps track of the available seats. Your program

1. Should use an array of structures
2. Each structure should hold a seat ID number, name and surname of person the seat reserved for.
3. Main Menu should look like in figure 1.1.



```

0 : Exit
1 : Add new reservation
2 : Delete available reservation
3 : Show available seats
4 : Show reserved seats
Please choose the number of the process you want to realize:1
-----Add new reservation-----
Enter Seat number: 2
Enter Person name: first
Enter Person surname: bilgin
-----
Seat number 2 is succesfully reserved

0 : Exit
1 : Add new reservation
2 : Delete available reservation
3 : Show available seats
4 : Show reserved seats
Please choose the number of the process you want to realize:2
Please enter seat which will be cancelled>2
-----
Your reservation is succesfully deleted.
    
```

Figure 1.1: Output of Laboratory study task1

The Code:

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

```
#include<stdlib.h>
```

```
struct ticket_menu{ //struct(new variable type definition
```

```
    char name[20]; //string name
```

```
    char sname[20]; //string surname
```

```
};
```

```
typedef ticket_menu tm; //ticket name is so long to make it shorter (tm)
```

```

main()
{
    tm seat[25]={""}; //defining array
    int i,a=1,b,c;
    while(a)
    {
        printf("0 : Exit\n");
        printf("1 : Add new reservation\n");
        printf("2 : Delete available reservation\n");
        printf("3 : Show available seats\n");
        printf("4 : Show reserved seats\n");
        printf("please choose the number of the process you want to realize : ");
        scanf("%d",&a); //take the option

        if(a==1) //if a choosen 1
        {
            printf("Enter Seat Number : ");
            scanf("%d",&c); //get seat number
            printf("Enter Person Name : ");
            scanf("%s",seat[c].name); //get name
            printf("Enter Person Surname : ");
            scanf("%s",seat[c].sname); //get surname
            printf("The reservation succesfully reserved\n");
        }
        else if(a==2) //if a choosen 2
        {
            printf("Please enter seat which will be cancelled>>");
            scanf("%d",&c); //get number of seat
            seat[c].name[0]='\0'; //assign to name NULL
            seat[c].sname[0]='\0'; //assign to surname Null
            printf("The reservation succesfully deleted");
        }
        else if(a==3) //if a choosen 3
        {
            for(i=0;i<25;i++)
            {
                if(!seat[i].name[0]) //if the seat is empty
                    printf("%d\n",i);
            }
        }
    }
}

```

```

    }

}

else if(a==4) //if a choosen 4
{

    for(i=0;i<25;i++)

    {

        if(seat[i].name[0])

            printf("%d\n",i);

    }

}

else if(a<0 || a>4) //if a diffrent from 0,1,2,3,4

    printf("Please enter again\n");

}

}

```

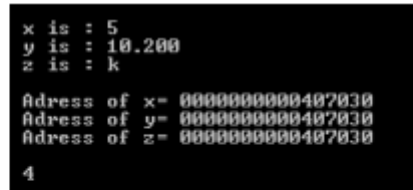
The outputs:

The screenshots show the following sequence of events:

- Top Left:** The program starts with a menu. The user selects option 4 (Show reserved seats). The program prompts for a process number (4) and a seat number (1). It then prompts for a person's name (ENES) and surname (ERIDH). The reservation is successfully reserved.
- Top Right:** The user selects option 2 (Delete available reservation). The program prompts for a process number (2) and a seat number (1). The reservation is successfully deleted.
- Bottom Left:** The user selects option 1 (Add new reservation). The program prompts for a process number (1) and a seat number (1). It then prompts for a person's name (enes) and surname (eridh). The reservation is successfully reserved.
- Bottom Right:** The user selects option 2 (Delete available reservation). The program prompts for a process number (3) and a seat number (1). The reservation is successfully deleted.

**Task 2:** Describe a union which members are an integer, a float and a char.

- (a) Write a program that can print size of the union, memory addresses and values of the variables. Interpret the results. (You can use **sizeof** command. )
- (b) Then describe variables as double and interpret the results.



```
x is : 5
y is : 10.200
z is : k

Address of x= 0000000000407030
Address of y= 0000000000407030
Address of z= 0000000000407030

4
```

Figure 2.1: Output of Laboratory study task2

The code:

```
#include<stdio.h>

#include<stdlib.h>

union ndc{ //union declaration

    int x; //declare int x

    double y; //declare double y

    char z; //declare char z

};

main()

{

    union ndc t; //declare new variable type of t

    t.x=5;

    printf("x is : %d\n",t.x);

    t.y=10.2;

    printf("y is : %.3lf\n",t.y);

    t.z='k';

    printf("z is : %c\n",t.z);

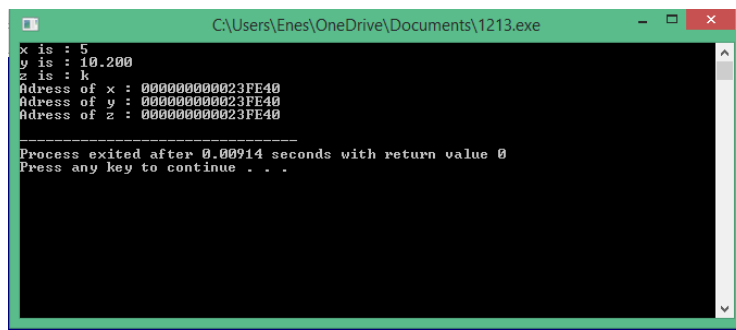
    printf("Adress of x : %p\n",&t.x);

    printf("Adress of y : %p\n",&t.y);

    printf("Adress of z : %p\n",&t.z);

}
```

Outputs:



A screenshot of a Windows command prompt window. The title bar is green and displays the file path "C:\Users\Enes\OneDrive\Documents\1213.exe". The command prompt has a black background with white text. The output shows variable declarations and memory addresses for variables x, y, and z. It also displays the process exit time and a prompt to press a key to continue.

```
x is : 5
y is : 10.200
z is : k
Address of x : 000000000023FE40
Address of y : 000000000023FE40
Address of z : 000000000023FE40

-----
Process exited after 0.00914 seconds with return value 0
Press any key to continue . . .
```