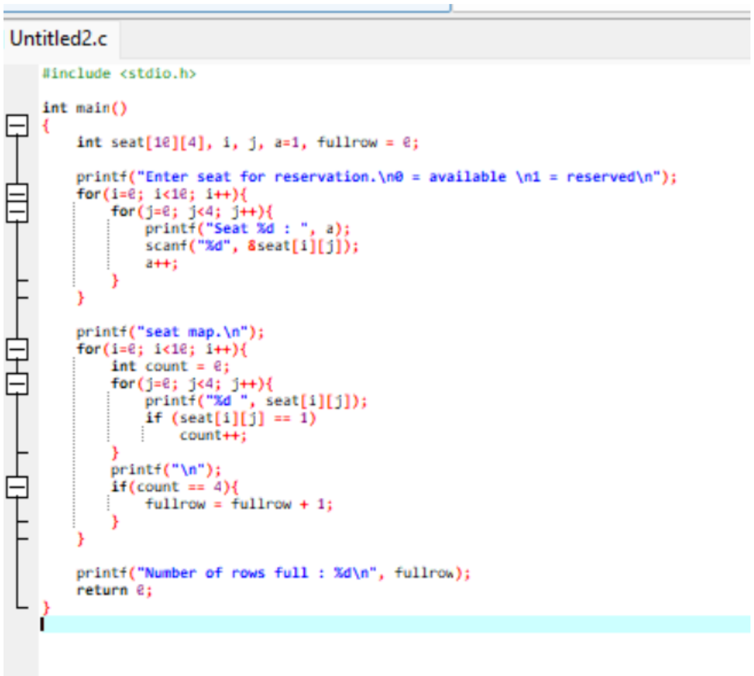
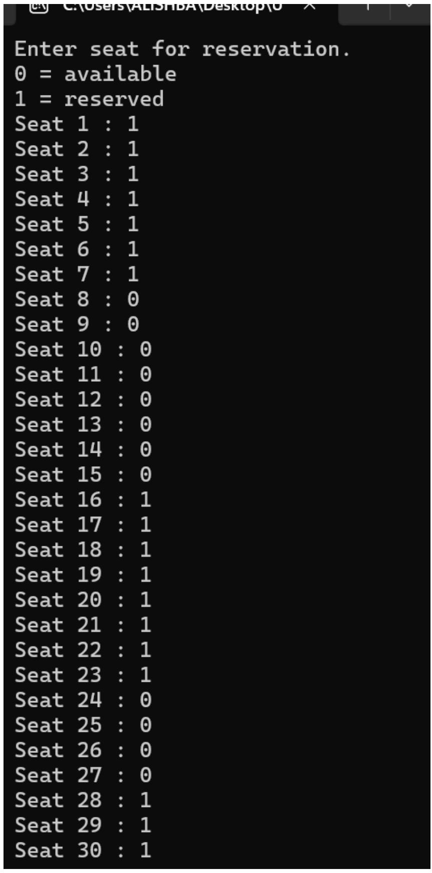
Lab 8 Asharib Faisal 25K-2013

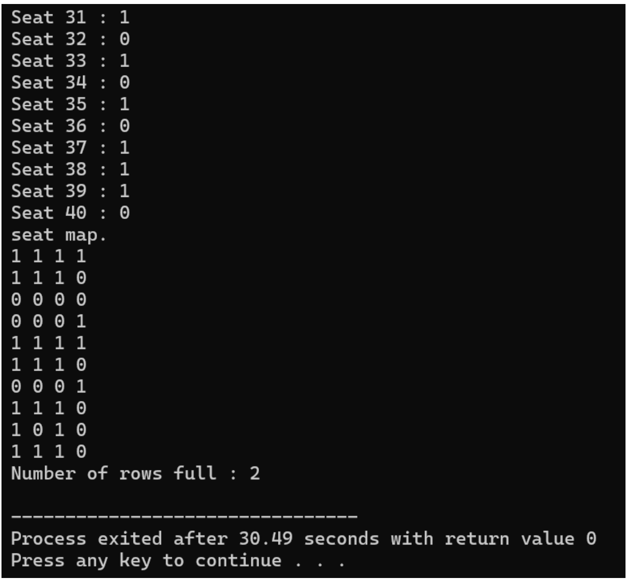
Q1



```c
#include <stdio.h>

int main()
{
    int seat[10][4], i, j, a=1, fullrow = 0;

    printf("Enter seat for reservation.\n0 = available \n1 = reserved\n");
    for(i=0; i<10; i++){
        for(j=0; j<4; j++){
            printf("Seat %d : ", a);
            scanf("%d", &seat[i][j]);
            a++;
        }
    }

    printf("seat map.\n");
    for(i=0; i<10; i++){
        int count = 0;
        for(j=0; j<4; j++){
            printf("%d ", seat[i][j]);
            if (seat[i][j] == 1)
                count++;
        }
        printf("\n");
        if(count == 4){
            fullrow = fullrow + 1;
        }
    }

    printf("Number of rows full : %d\n", fullrow);
    return 0;
}
```

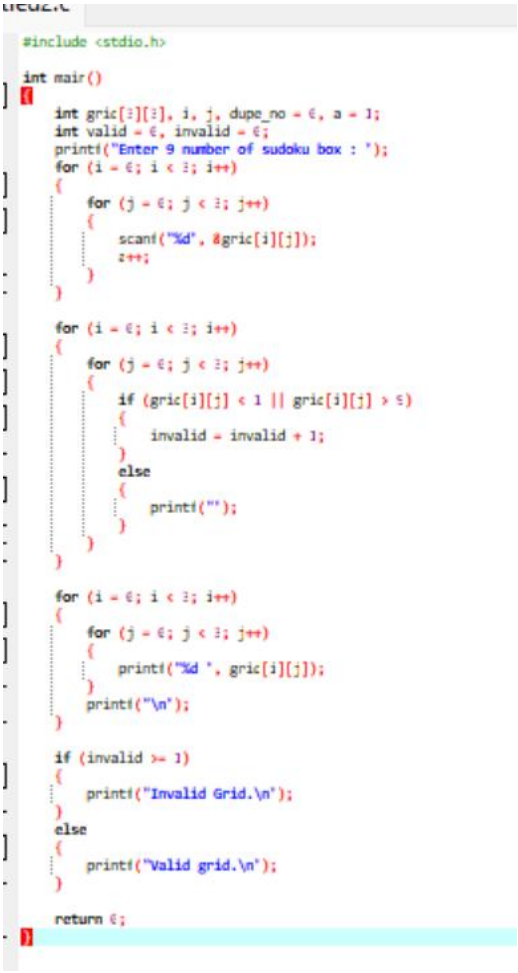


```
Enter seat for reservation.
0 = available
1 = reserved
Seat 1 : 1
Seat 2 : 1
Seat 3 : 1
Seat 4 : 1
Seat 5 : 1
Seat 6 : 1
Seat 7 : 1
Seat 8 : 0
Seat 9 : 0
Seat 10 : 0
Seat 11 : 0
Seat 12 : 0
Seat 13 : 0
Seat 14 : 0
Seat 15 : 0
Seat 16 : 1
Seat 17 : 1
Seat 18 : 1
Seat 19 : 1
Seat 20 : 1
Seat 21 : 1
Seat 22 : 1
Seat 23 : 1
Seat 24 : 0
Seat 25 : 0
Seat 26 : 0
Seat 27 : 0
Seat 28 : 1
Seat 29 : 1
Seat 30 : 1
```

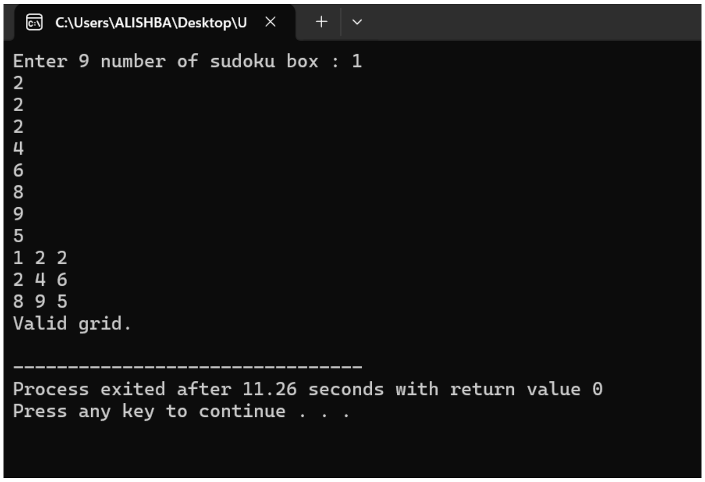


```
Seat 31 : 1
Seat 32 : 0
Seat 33 : 1
Seat 34 : 0
Seat 35 : 1
Seat 36 : 0
Seat 37 : 1
Seat 38 : 1
Seat 39 : 1
Seat 40 : 0
seat map.
1 1 1 1
1 1 1 0
0 0 0 0
0 0 0 1
1 1 1 1
1 1 1 0
0 0 0 1
1 1 1 0
1 0 1 0
1 1 1 0
Number of rows full : 2

--------------------------------
Process exited after 30.49 seconds with return value 0
Press any key to continue . . .
```

Q2

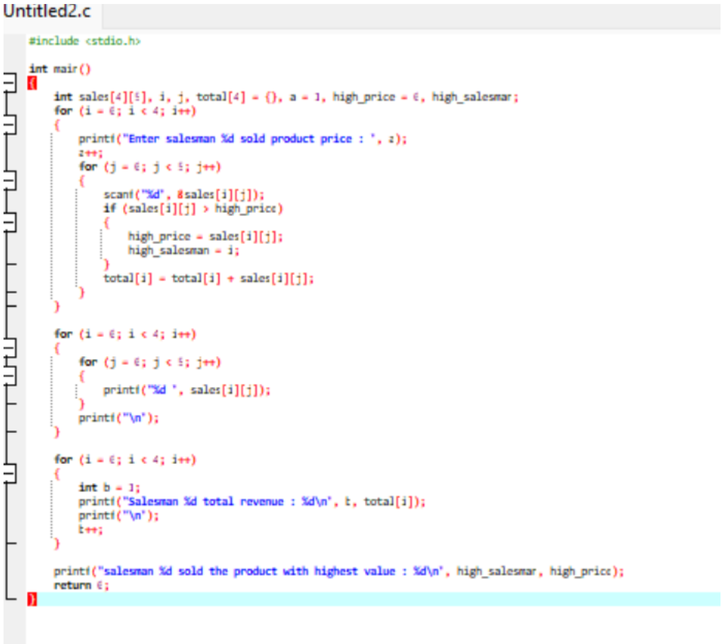


```c
#include <stdio.h>

int main()
{
    int grid[3][3], i, j, dupe_no = 0, a = 1;
    int valid = 0, invalid = 0;
    printf("Enter 9 number of sudoku box : ");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &grid[i][j]);
            a++;
        }
    }

    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            if (grid[i][j] < 1 || grid[i][j] > 9)
            {
                invalid = invalid + 1;
            }
            else
            {
                printf("");
            }
        }
    }

    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d ", grid[i][j]);
        }
        printf("\n");
    }

    if (invalid >= 1)
    {
        printf("Invalid Grid.\n");
    }
    else
    {
        printf("Valid grid.\n");
    }

    return 0;
}
```

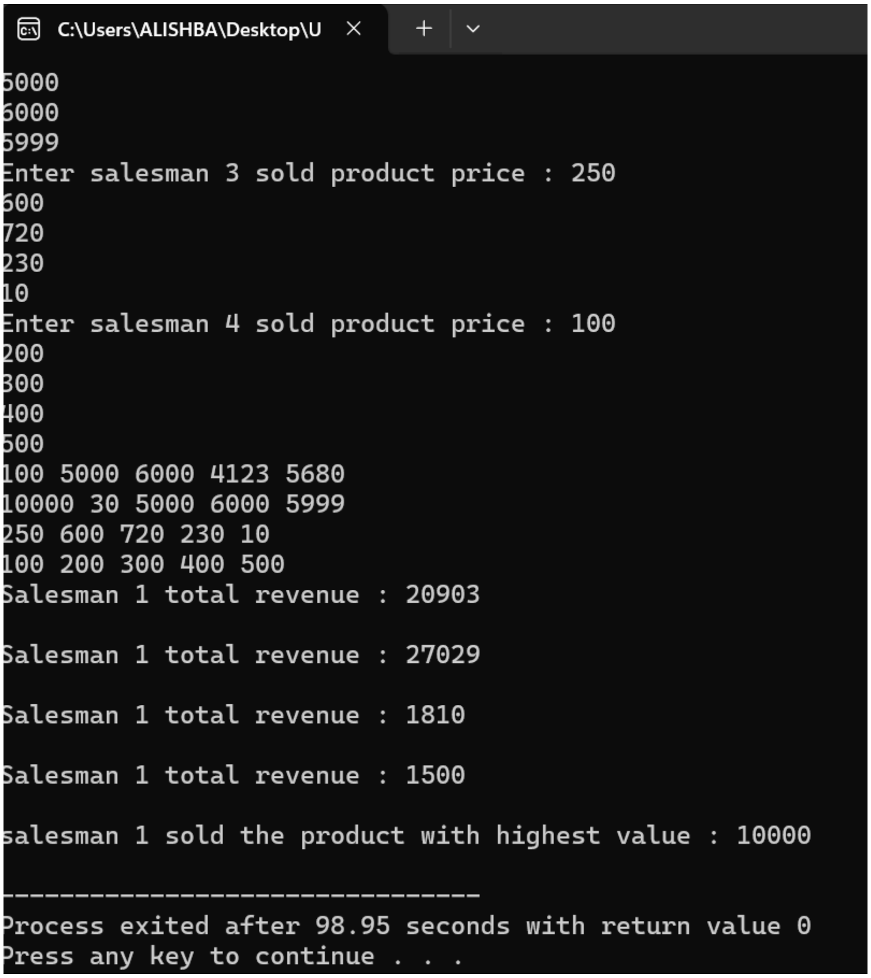


```
Enter 9 number of sudoku box : 1
2
2
2
4
6
8
9
5
1 2 2
2 4 6
8 9 5
Valid grid.

--------------------------------
Process exited after 11.26 seconds with return value 0
Press any key to continue . . .
```

Q3

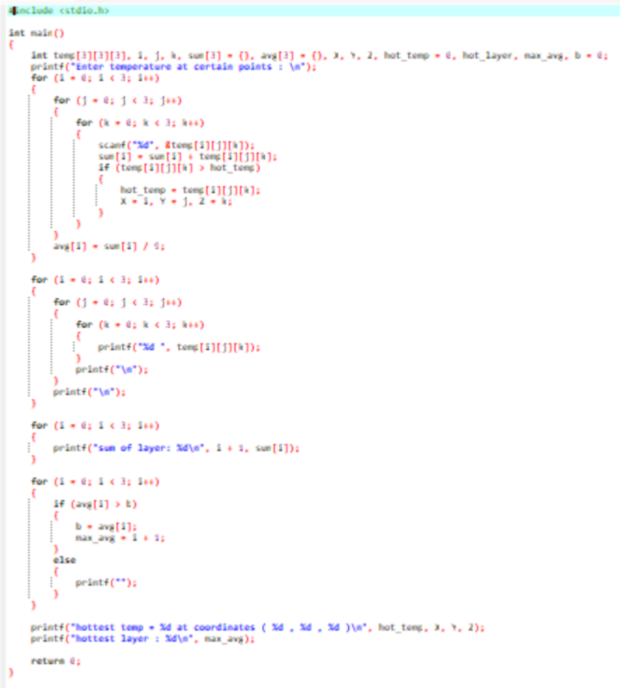

Untitled2.c

```
#include <stdio.h>

int main()
{
    int sales[4][5], i, j, total[4] = {0}, a = 1, high_price = 0, high_salesman;
    for (i = 0; i < 4; i++)
    {
        printf("Enter salesman %d sold product price : ", a);
        a++;
        for (j = 0; j < 5; j++)
        {
            scanf("%d", &sales[i][j]);
            if (sales[i][j] > high_price)
            {
                high_price = sales[i][j];
                high_salesman = i;
            }
            total[i] = total[i] + sales[i][j];
        }
    }

    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 5; j++)
        {
            printf("%d ", sales[i][j]);
        }
        printf("\n");
    }

    for (i = 0; i < 4; i++)
    {
        int b = 1;
        printf("Salesman %d total revenue : %d\n", b, total[i]);
        printf("\n");
        b++;
    }

    printf("salesman %d sold the product with highest value : %d\n", high_salesman, high_price);
    return 0;
}
```

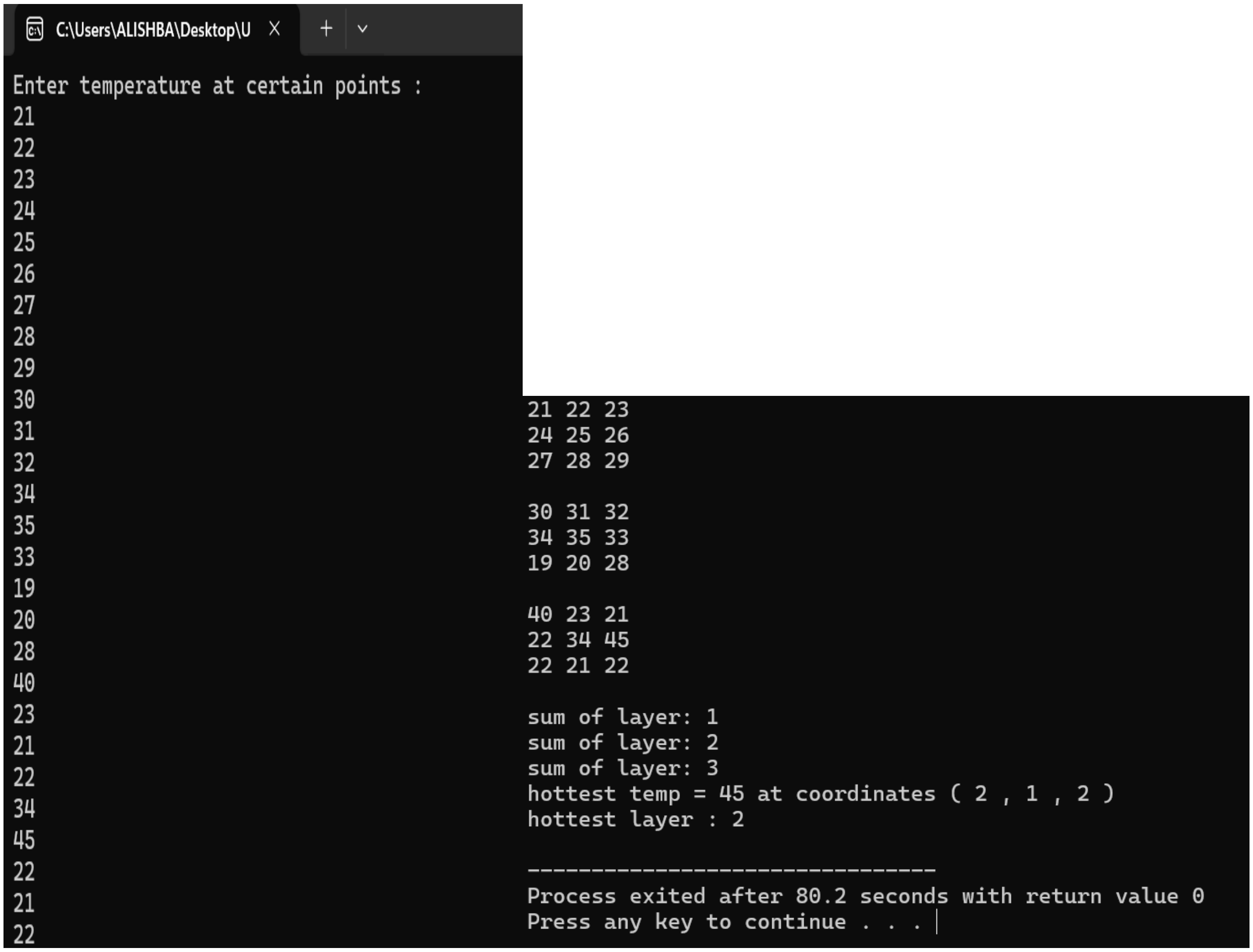

C:\Users\ALISHBA\Desktop\U

```
5000
6000
5999
Enter salesman 3 sold product price : 250
600
720
230
10
Enter salesman 4 sold product price : 100
200
300
400
500
100 5000 6000 4123 5680
10000 30 5000 6000 5999
250 600 720 230 10
100 200 300 400 500
Salesman 1 total revenue : 20903

Salesman 1 total revenue : 27029

Salesman 1 total revenue : 1810

Salesman 1 total revenue : 1500

salesman 1 sold the product with highest value : 10000

--------------------------------
Process exited after 98.95 seconds with return value 0
Press any key to continue . . .
```

Q4

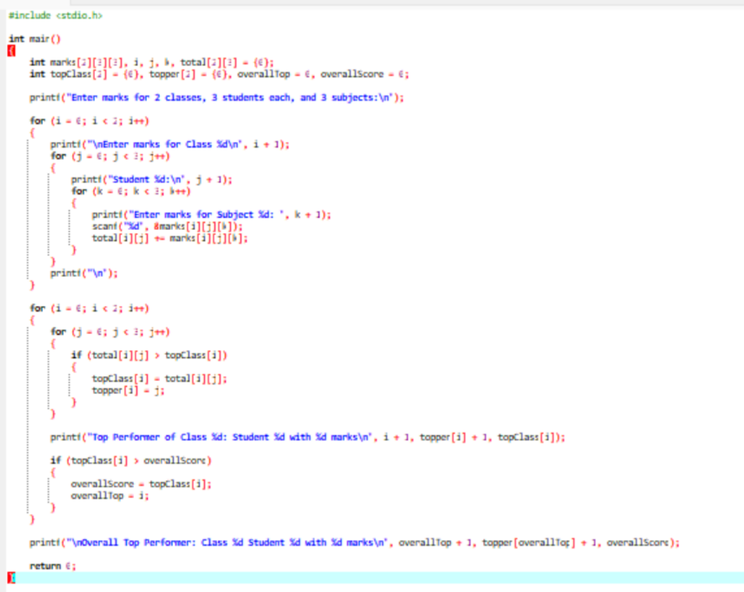


```c
#include <stdio.h>

int main()
{
    int temp[3][3][3], i, j, k, sum[3] = {}, avg[3] = {}, X, Y, Z, hot_temp = 0, hot_layer, max_avg, b = 0;
    printf("Enter temperature at certain points : \n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            for (k = 0; k < 3; k++)
            {
                scanf("%d", &temp[i][j][k]);
                sum[i] = sum[i] + temp[i][j][k];
                if (temp[i][j][k] > hot_temp)
                {
                    hot_temp = temp[i][j][k];
                    X = i, Y = j, Z = k;
                }
            }
        }
        avg[i] = sum[i] / 9;
    }

    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            for (k = 0; k < 3; k++)
            {
                printf("%d ", temp[i][j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }

    for (i = 0; i < 3; i++)
    {
        printf("sum of layer: %d\n", i + 1, sum[i]);
    }

    for (i = 0; i < 3; i++)
    {
        if (avg[i] > b)
        {
            b = avg[i];
            max_avg = i + 1;
        }
        else
        {
            printf("");
        }
    }

    printf("hottest temp = %d at coordinates ( %d , %d , %d )\n", hot_temp, X, Y, Z);
    printf("hottest layer : %d\n", max_avg);

    return 0;
}
```

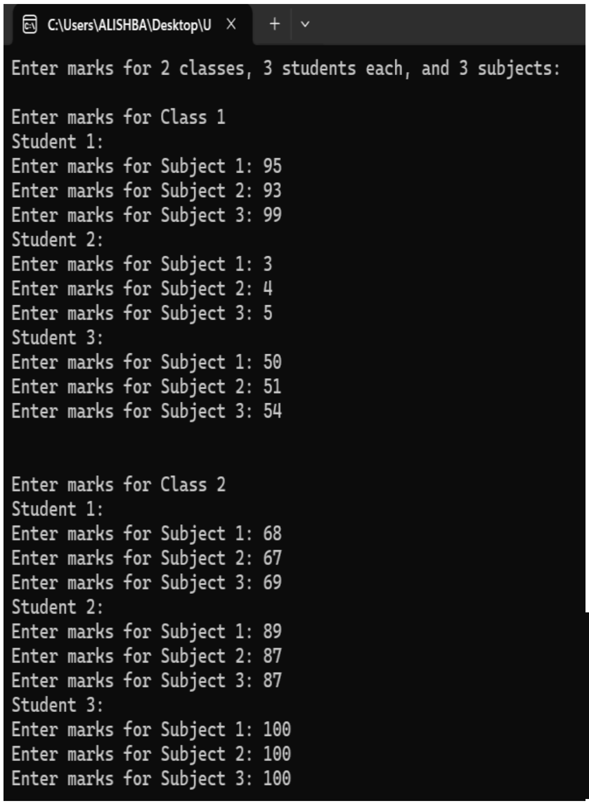


```
Enter temperature at certain points :
21
22
23
24
25
26
27
28
29
30
31
32
34
35
33
19
20
28
40
23
21
22
34
45
22
21
22
21 22 23
24 25 26
27 28 29

30 31 32
34 35 33
19 20 28

40 23 21
22 34 45
22 21 22

sum of layer: 1
sum of layer: 2
sum of layer: 3
hottest temp = 45 at coordinates ( 2 , 1 , 2 )
hottest layer : 2

--------------------------------
Process exited after 80.2 seconds with return value 0
Press any key to continue . . .
```

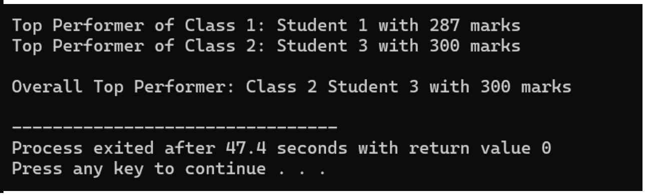
Q5



```c
#include <stdio.h>

int main()
{
    int marks[2][3][3], i, j, k, total[2][3] = {0};
    int topClass[2] = {0}, topper[2] = {0}, overallTop = 0, overallScore = 0;

    printf("Enter marks for 2 classes, 3 students each, and 3 subjects:\n");

    for (i = 0; i < 2; i++)
    {
        printf("\nEnter marks for Class %d\n", i + 1);
        for (j = 0; j < 3; j++)
        {
            printf("Student %d:\n", j + 1);
            for (k = 0; k < 3; k++)
            {
                printf("Enter marks for Subject %d: ", k + 1);
                scanf("%d", &marks[i][j][k]);
                total[i][j] += marks[i][j][k];
            }
        }
        printf("\n");
    }

    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            if (total[i][j] > topClass[i])
            {
                topClass[i] = total[i][j];
                topper[i] = j;
            }
        }

        printf("Top Performer of Class %d: Student %d with %d marks\n", i + 1, topper[i] + 1, topClass[i]);

        if (topClass[i] > overallScore)
        {
            overallScore = topClass[i];
            overallTop = i;
        }
    }

    printf("\nOverall Top Performer: Class %d Student %d with %d marks\n", overallTop + 1, topper[overallTop] + 1, overallScore);

    return 0;
}
```

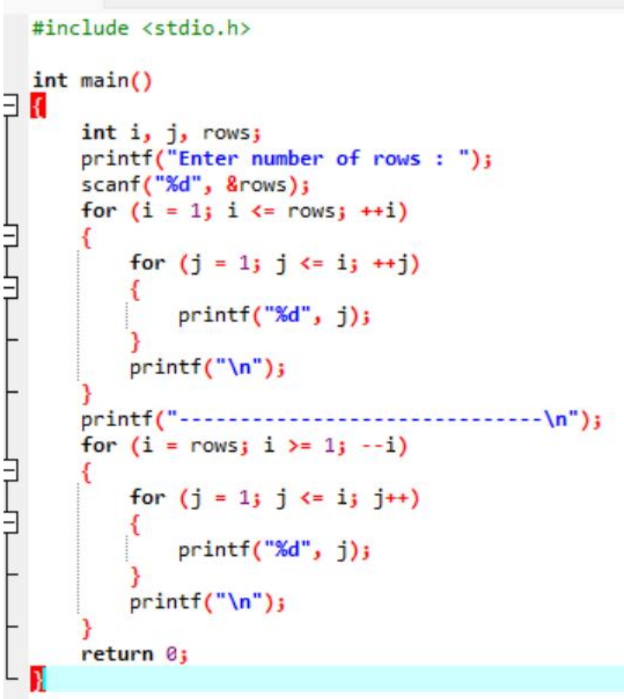


```
Enter marks for 2 classes, 3 students each, and 3 subjects:

Enter marks for Class 1
Student 1:
Enter marks for Subject 1: 95
Enter marks for Subject 2: 93
Enter marks for Subject 3: 99
Student 2:
Enter marks for Subject 1: 3
Enter marks for Subject 2: 4
Enter marks for Subject 3: 5
Student 3:
Enter marks for Subject 1: 50
Enter marks for Subject 2: 51
Enter marks for Subject 3: 54


Enter marks for Class 2
Student 1:
Enter marks for Subject 1: 68
Enter marks for Subject 2: 67
Enter marks for Subject 3: 69
Student 2:
Enter marks for Subject 1: 89
Enter marks for Subject 2: 87
Enter marks for Subject 3: 87
Student 3:
Enter marks for Subject 1: 100
Enter marks for Subject 2: 100
Enter marks for Subject 3: 100
```

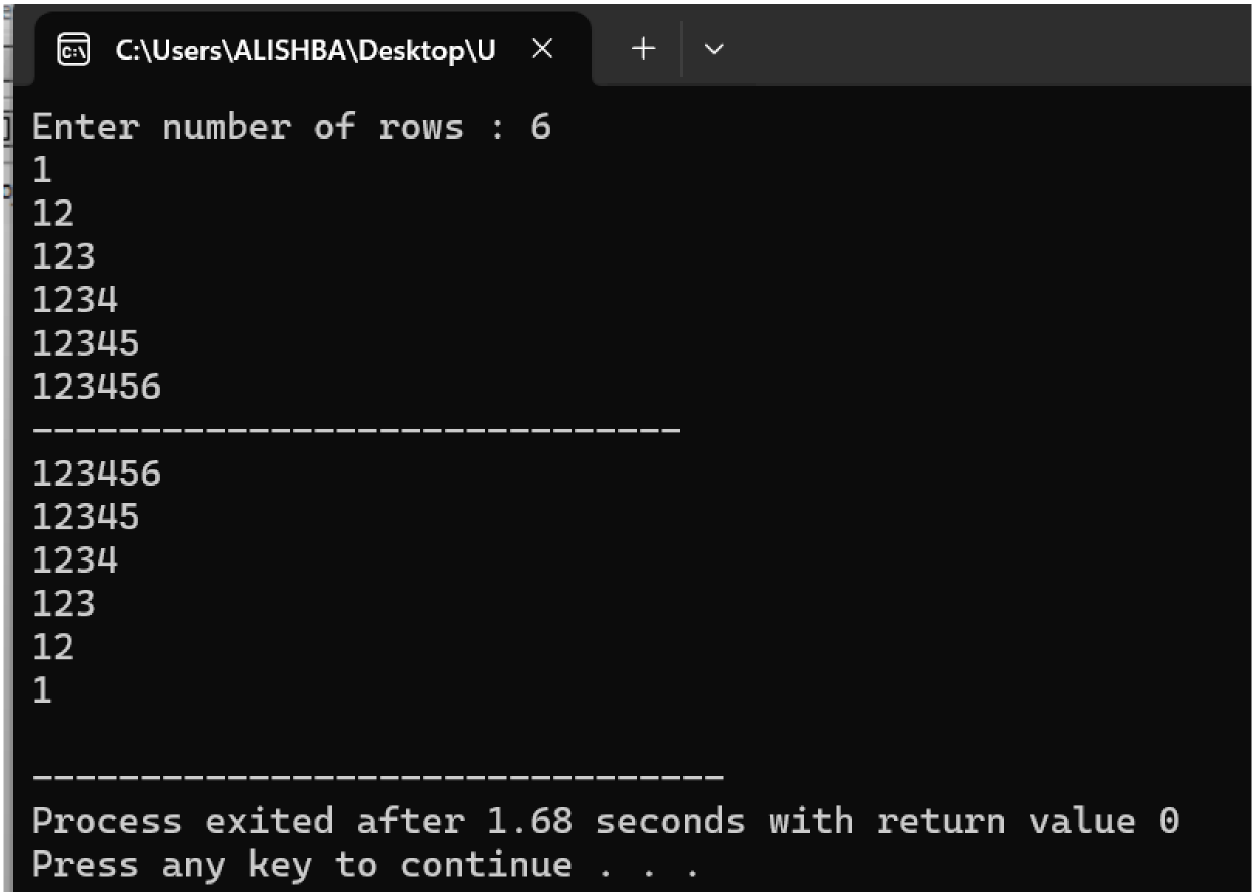


```
Top Performer of Class 1: Student 1 with 287 marks
Top Performer of Class 2: Student 3 with 300 marks

Overall Top Performer: Class 2 Student 3 with 300 marks

--------------------------------
Process exited after 47.4 seconds with return value 0
Press any key to continue . . .
```

Q6

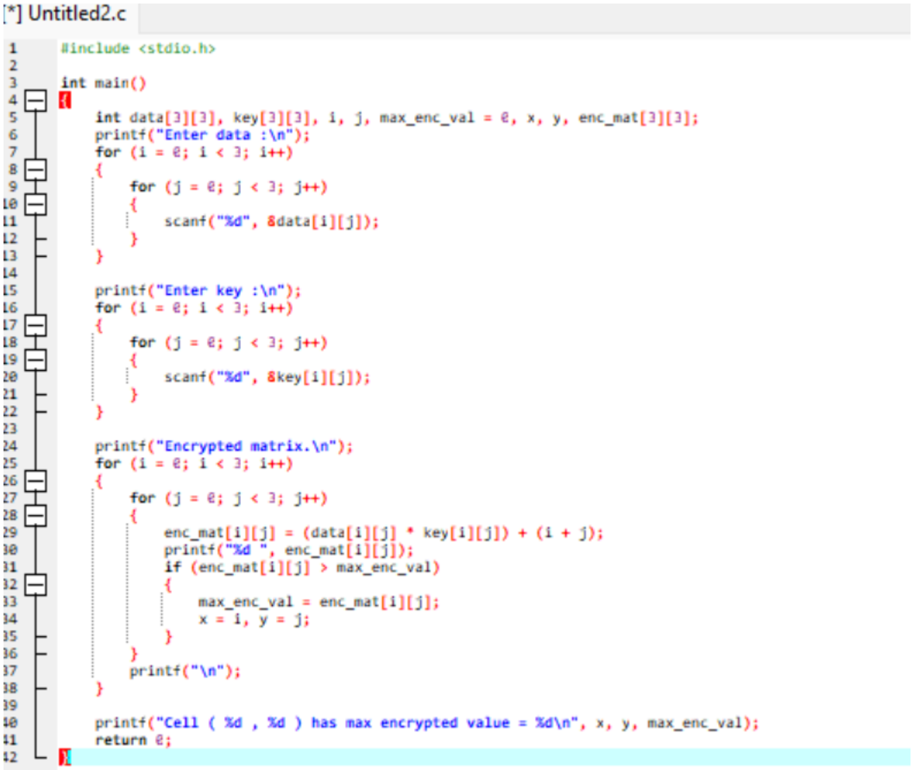


```c
#include <stdio.h>

int main()
{
    int i, j, rows;
    printf("Enter number of rows : ");
    scanf("%d", &rows);
    for (i = 1; i <= rows; ++i)
    {
        for (j = 1; j <= i; ++j)
        {
            printf("%d", j);
        }
        printf("\n");
    }
    printf("-----------------------------\n");
    for (i = rows; i >= 1; --i)
    {
        for (j = 1; j <= i; j++)
        {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}
```



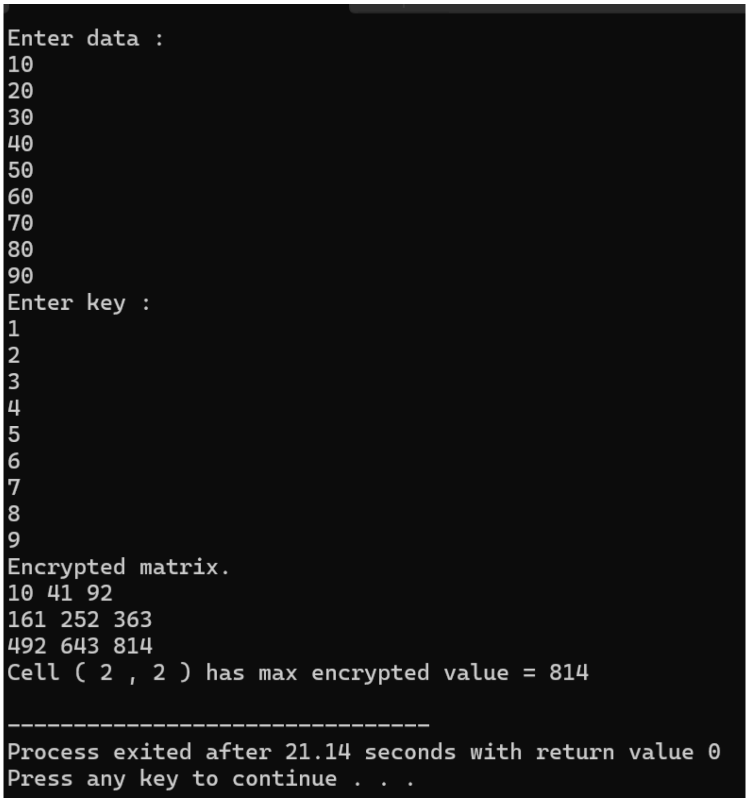
```
Enter number of rows : 6
1
12
123
1234
12345
123456
-----------------------------
123456
12345
1234
123
12
1


--------------------------------
Process exited after 1.68 seconds with return value 0
Press any key to continue . . .
```

Q7



[*] Untitled2.c

```c
#include <stdio.h>

int main()
{
    int data[3][3], key[3][3], i, j, max_enc_val = 0, x, y, enc_mat[3][3];
    printf("Enter data :\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &data[i][j]);
        }
    }

    printf("Enter key :\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &key[i][j]);
        }
    }

    printf("Encrypted matrix.\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            enc_mat[i][j] = (data[i][j] * key[i][j]) + (i + j);
            printf("%d ", enc_mat[i][j]);
            if (enc_mat[i][j] > max_enc_val)
            {
                max_enc_val = enc_mat[i][j];
                x = i, y = j;
            }
        }
        printf("\n");
    }

    printf("Cell ( %d , %d ) has max encrypted value = %d\n", x, y, max_enc_val);
    return 0;
}
```



```
Enter data :
10
20
30
40
50
60
70
80
90
Enter key :
1
2
3
4
5
6
7
8
9
Encrypted matrix.
10 41 92
161 252 363
492 643 814
Cell ( 2 , 2 ) has max encrypted value = 814

--------------------------------
Process exited after 21.14 seconds with return value 0
Press any key to continue . . .
```