Student Name: ________Wajiha Nazir__________________**Roll No:** ______25K-3021__________ **Section:** BSE-1B

Question 1:

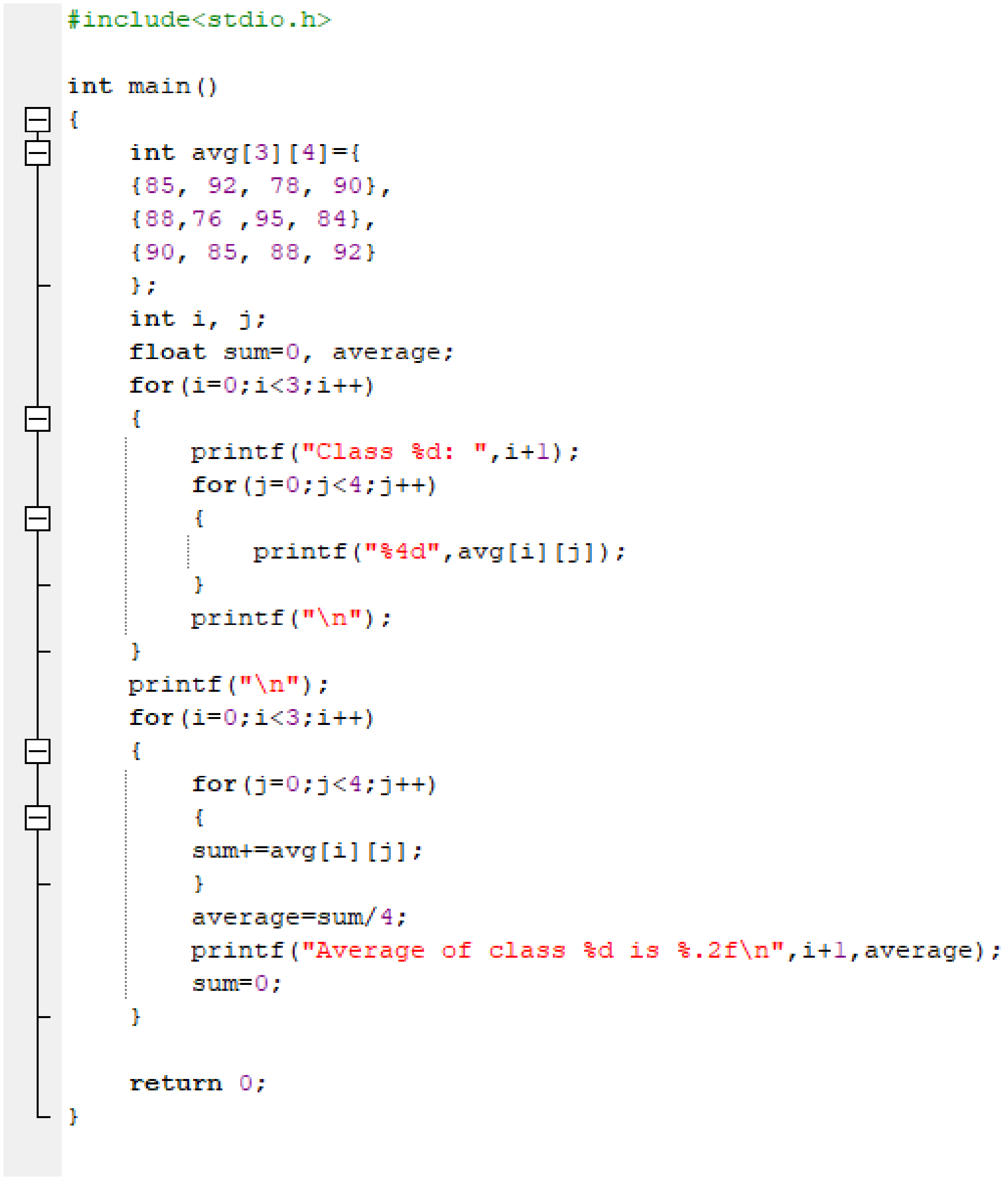
Imagine a teacher has a gradebook for 3 classes, with each class having 4 students. The teacher wants to find the average score for each class.

Class 1: 85, 92, 78, 90

Class 2: 88, 76, 95, 84

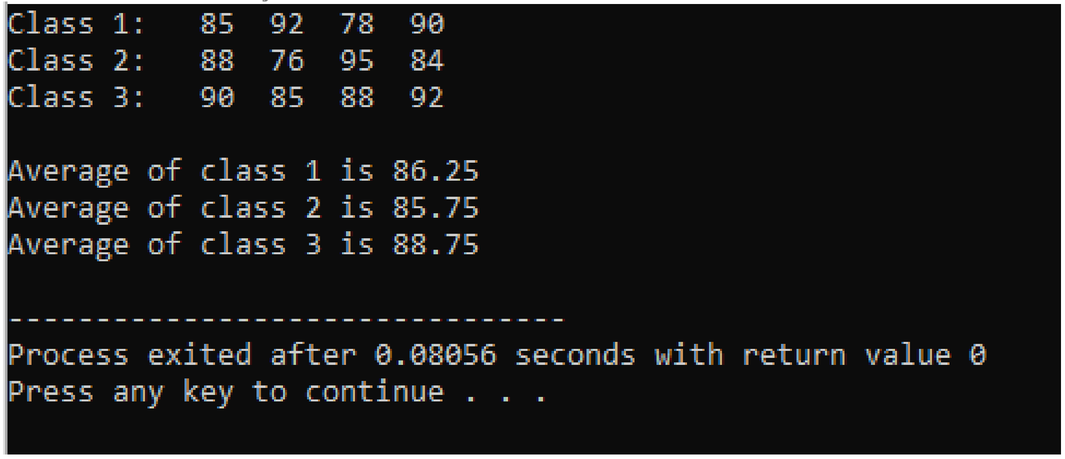
Class 3: 90, 85, 88, 92

Calculate and show the average score for each class.



```
#include<stdio.h>

int main()
{
    int avg[3][4]={
    {85, 92, 78, 90},
    {88,76 ,95, 84},
    {90, 85, 88, 92}
    };
    int i, j;
    float sum=0, average;
    for(i=0;i<3;i++)
    {
        printf("Class %d: ",i+1);
        for(j=0;j<4;j++)
        {
            printf("%4d",avg[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
        sum+=avg[i][j];
        }
        average=sum/4;
        printf("Average of class %d is %.2f\n",i+1,average);
        sum=0;
    }

    return 0;
}
```

Student Name: ________Wajiha Nazir__________________Roll No: ______25K-3021__________ Section: BSE-1B



```
Class 1:    85   92   78   90
Class 2:    88   76   95   84
Class 3:    90   85   88   92

Average of class 1 is 86.25
Average of class 2 is 85.75
Average of class 3 is 88.75

--------------------------------
Process exited after 0.08056 seconds with return value 0
Press any key to continue . . .
```

Question 2:

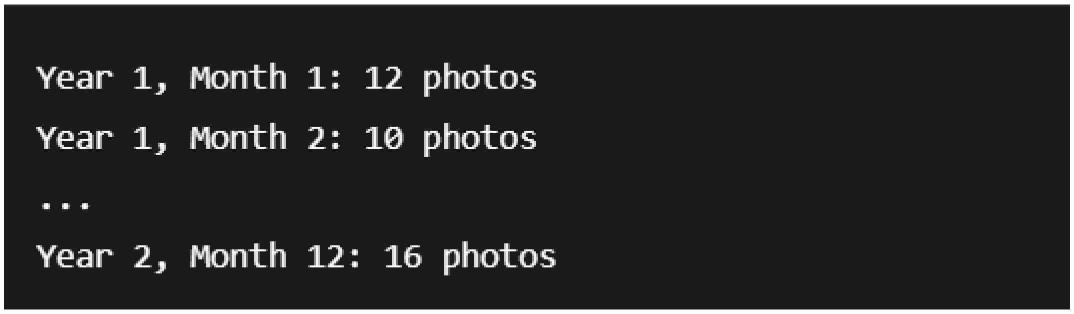
A family has a photo album organized by year and then by month. They want to list how many photos they took each month for the last 2 years.

Year 1: 12, 10, 15, 8, 5, 20, 25, 30, 10, 5, 8, 15

Year 2: 10, 12, 18, 9, 6, 22, 28, 35, 12, 7, 9, 16

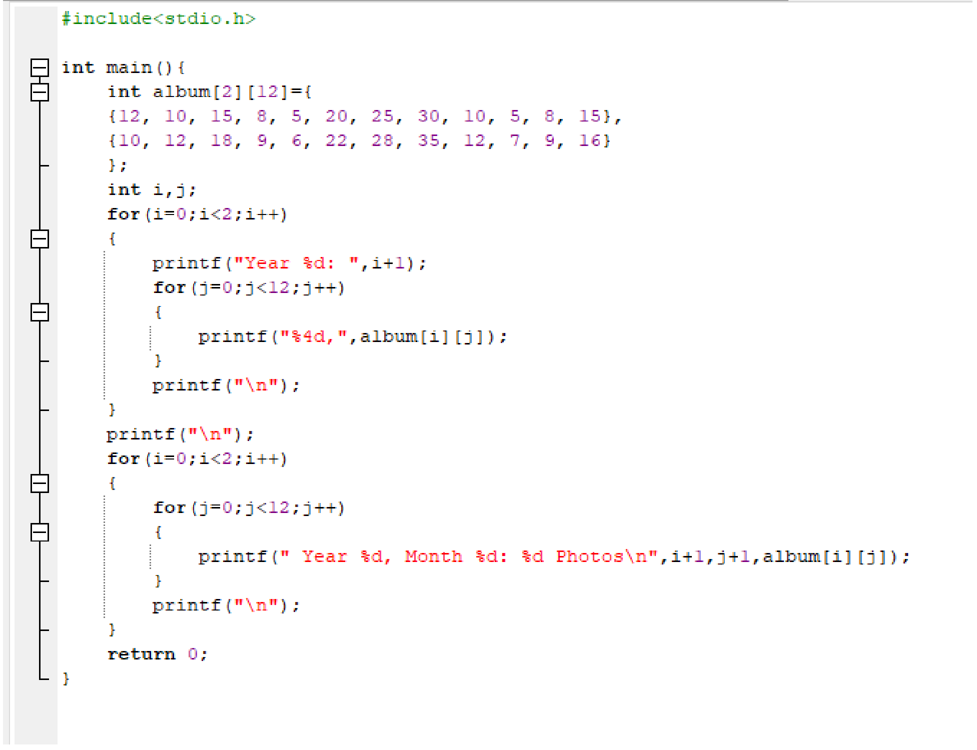
Print the number of photos for each month of each year.

Output:

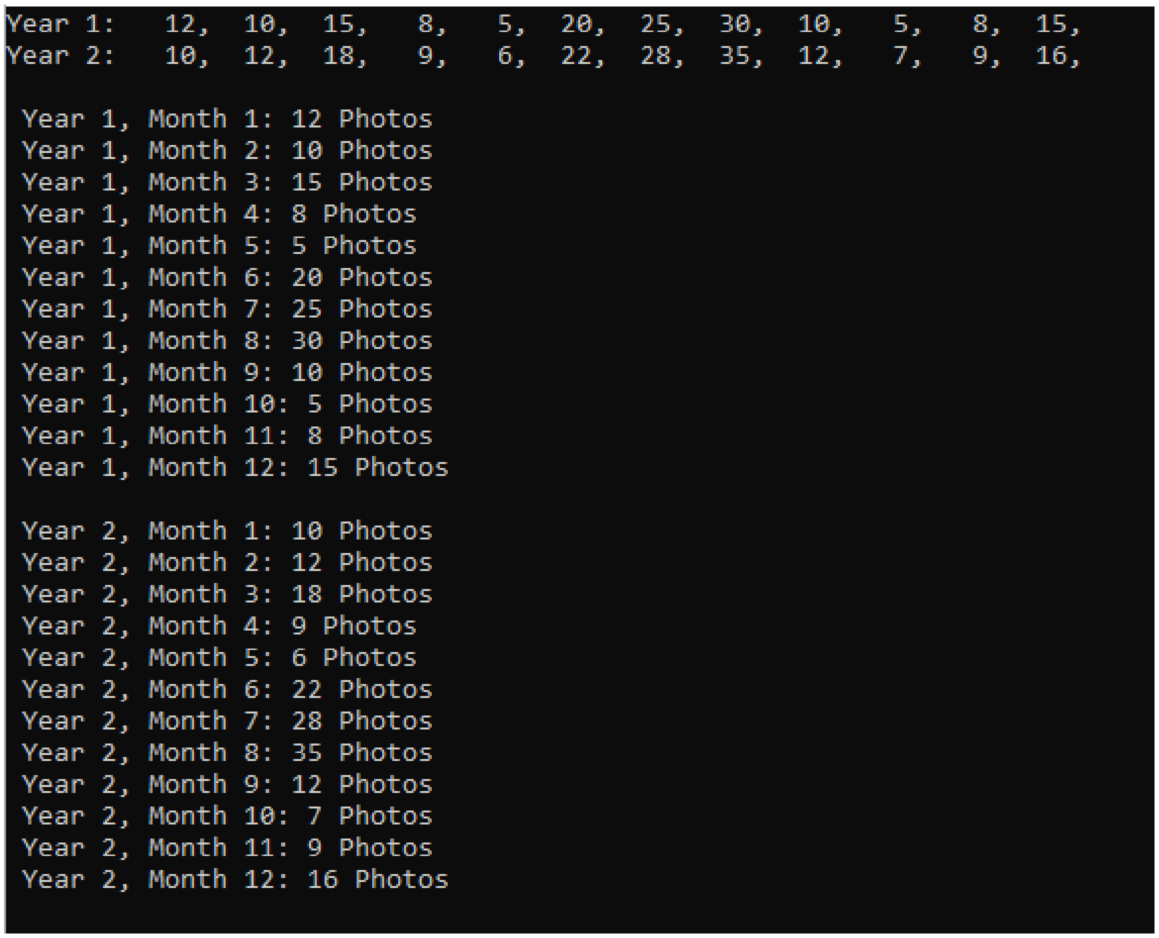


```
Year 1, Month 1: 12 photos
Year 1, Month 2: 10 photos
...
Year 2, Month 12: 16 photos
```

Student Name: ________Wajiha Nazir__________________**Roll No:** ______25K-3021__________ **Section: BSE-1B**



```c
#include<stdio.h>

int main(){
    int album[2][12]={
    {12, 10, 15, 8, 5, 20, 25, 30, 10, 5, 8, 15},
    {10, 12, 18, 9, 6, 22, 28, 35, 12, 7, 9, 16}
    };
    int i,j;
    for(i=0;i<2;i++)
    {
        printf("Year %d: ",i+1);
        for(j=0;j<12;j++)
        {
            printf("%4d,",album[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<12;j++)
        {
            printf(" Year %d, Month %d: %d Photos\n",i+1,j+1,album[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```



```
Year 1:   12,  10,  15,   8,   5,  20,  25,  30,  10,   5,   8,  15,
Year 2:   10,  12,  18,   9,   6,  22,  28,  35,  12,   7,   9,  16,

 Year 1, Month 1: 12 Photos
 Year 1, Month 2: 10 Photos
 Year 1, Month 3: 15 Photos
 Year 1, Month 4: 8 Photos
 Year 1, Month 5: 5 Photos
 Year 1, Month 6: 20 Photos
 Year 1, Month 7: 25 Photos
 Year 1, Month 8: 30 Photos
 Year 1, Month 9: 10 Photos
 Year 1, Month 10: 5 Photos
 Year 1, Month 11: 8 Photos
 Year 1, Month 12: 15 Photos

 Year 2, Month 1: 10 Photos
 Year 2, Month 2: 12 Photos
 Year 2, Month 3: 18 Photos
 Year 2, Month 4: 9 Photos
 Year 2, Month 5: 6 Photos
 Year 2, Month 6: 22 Photos
 Year 2, Month 7: 28 Photos
 Year 2, Month 8: 35 Photos
 Year 2, Month 9: 12 Photos
 Year 2, Month 10: 7 Photos
 Year 2, Month 11: 9 Photos
 Year 2, Month 12: 16 Photos
```

Student Name: ________Wajiha Nazir____________________Roll No: ______25K-3021__________ Section: BSE-1B

Question 3:

Create a program that works with a small 4x4 black and white image. The program should:

Create an original image where 1 represents white pixels and 0 represents black pixels

Display the original image

Create an inverted version (negative) of the image where white becomes black and black becomes white

Display both images side by side

Count how many white pixels are in the original image

Original Image 4x4:

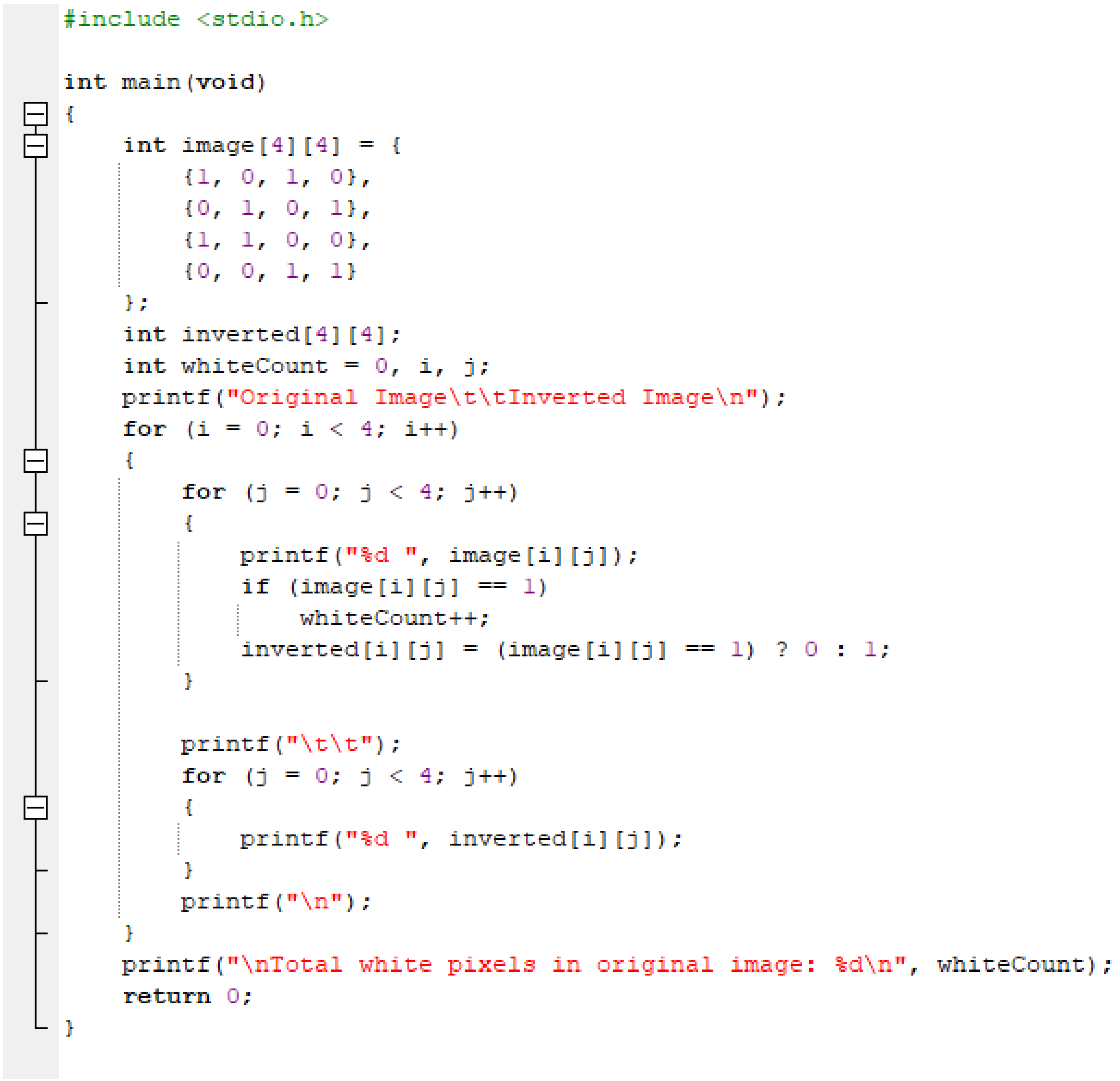
{1, 0, 1, 0},

 {0, 1, 0, 1},

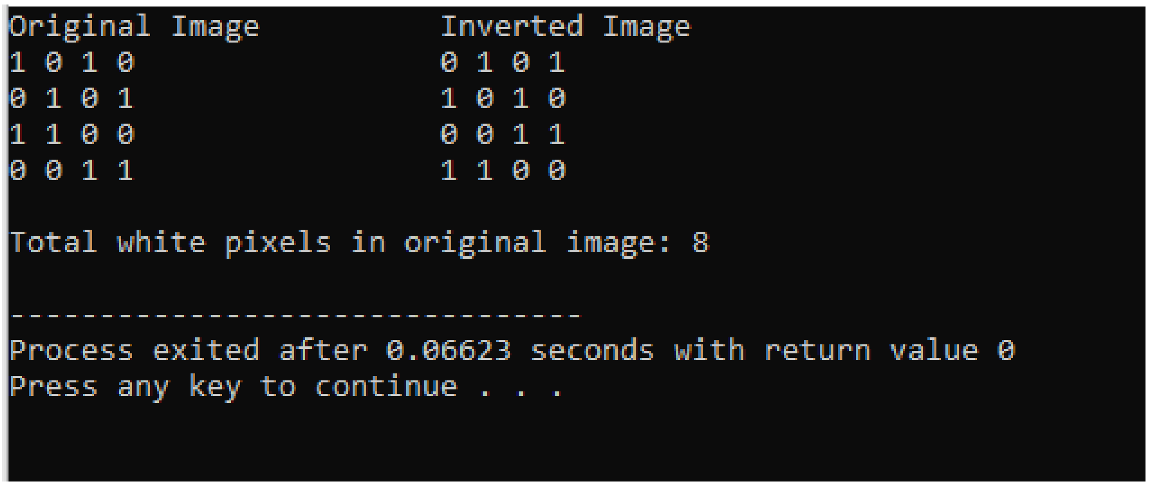
 {1, 1, 0, 0},

 {0, 0, 1, 1}

Student Name: ________Wajiha Nazir__________________Roll No: ______25K-3021__________ Section: BSE-1B



```c
#include <stdio.h>

int main(void)
{
    int image[4][4] = {
        {1, 0, 1, 0},
        {0, 1, 0, 1},
        {1, 1, 0, 0},
        {0, 0, 1, 1}
    };
    int inverted[4][4];
    int whiteCount = 0, i, j;
    printf("Original Image\t\tInverted Image\n");
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            printf("%d ", image[i][j]);
            if (image[i][j] == 1)
                whiteCount++;
            inverted[i][j] = (image[i][j] == 1) ? 0 : 1;
        }

        printf("\t\t");
        for (j = 0; j < 4; j++)
        {
            printf("%d ", inverted[i][j]);
        }
        printf("\n");
    }
    printf("\nTotal white pixels in original image: %d\n", whiteCount);
    return 0;
}
```



```
Original Image          Inverted Image
1 0 1 0                 0 1 0 1
0 1 0 1                 1 0 1 0
1 1 0 0                 0 0 1 1
0 0 1 1                 1 1 0 0

Total white pixels in original image: 8

--------------------------------
Process exited after 0.06623 seconds with return value 0
Press any key to continue . . .
```

Student Name: ________Wajiha Nazir____________________Roll No: ______25K-3021__________ Section: BSE-1B

Question # 4

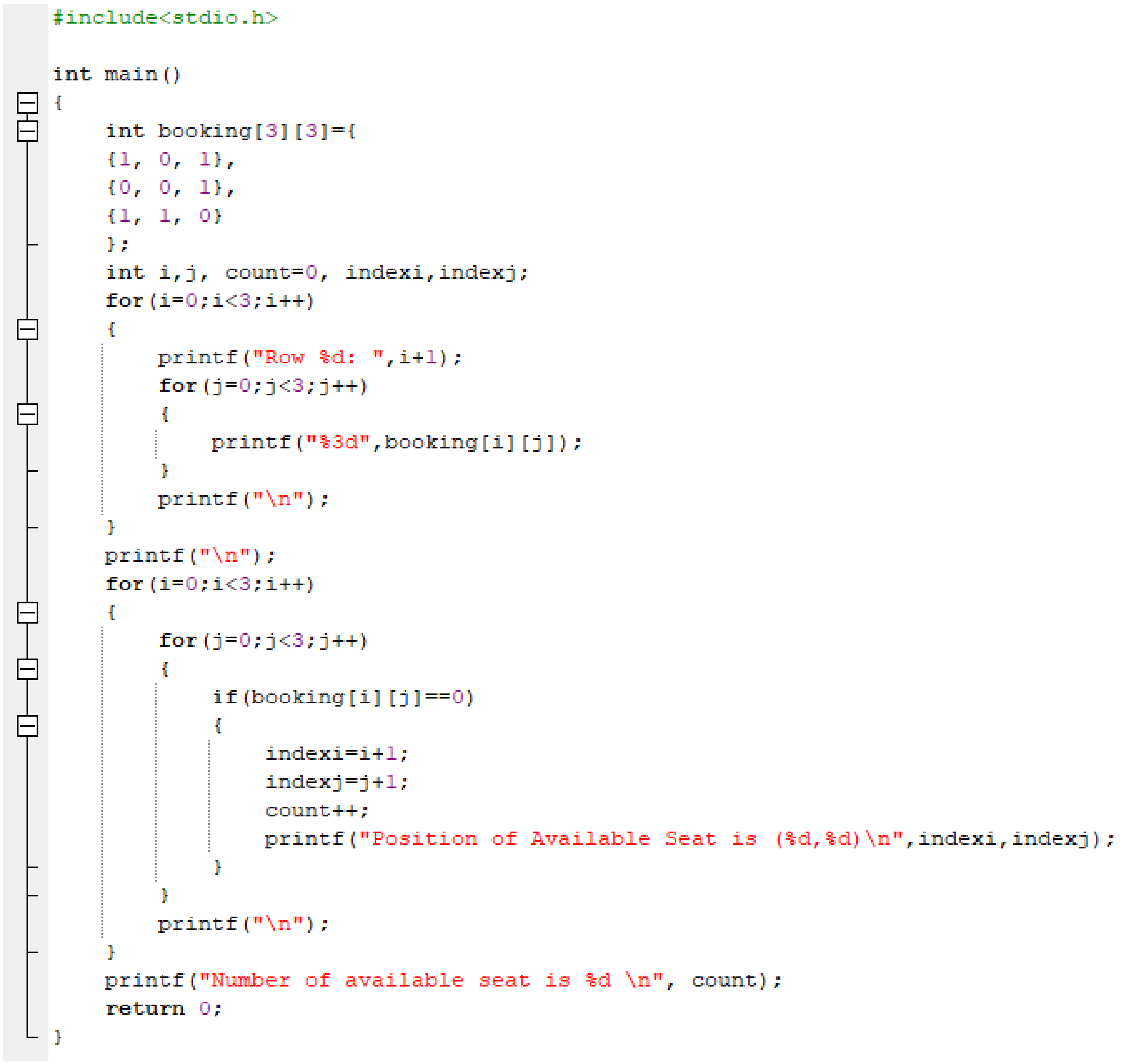
A small cinema has 3 rows with 3 seats in each row. The booking system marks a seat as 1 if it's booked and 0 if it's available. Find the total number of available seats and list their positions.

Row 1: 1, 0, 1

Row 2: 0, 0, 1

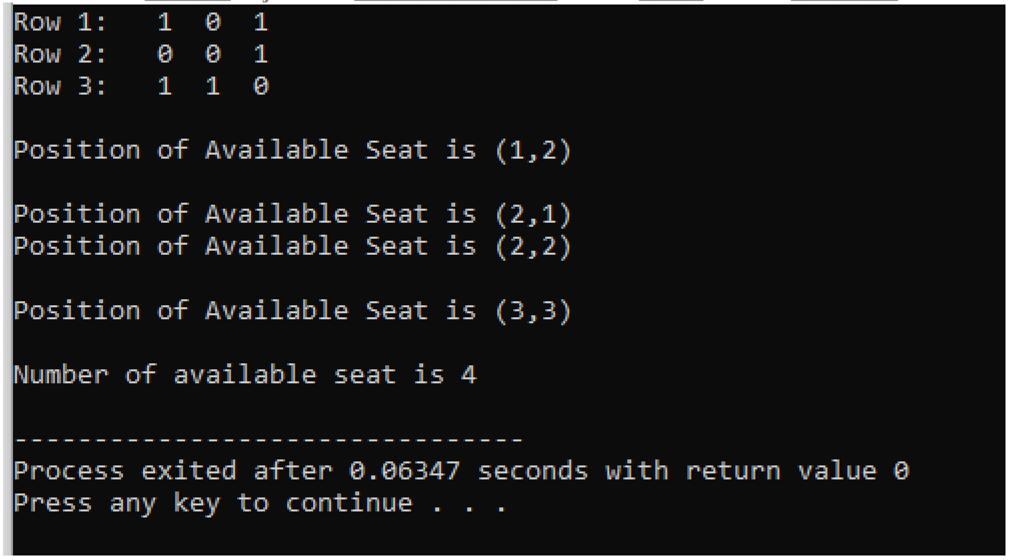
Row 3: 1, 1, 0

Count all available seats and print their row and seat number.



```c
#include<stdio.h>

int main()
{
    int booking[3][3]={
    {1, 0, 1},
    {0, 0, 1},
    {1, 1, 0}
    };
    int i,j, count=0, indexi,indexj;
    for(i=0;i<3;i++)
    {
        printf("Row %d: ",i+1);
        for(j=0;j<3;j++)
        {
            printf("%3d",booking[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            if(booking[i][j]==0)
            {
                indexi=i+1;
                indexj=j+1;
                count++;
                printf("Position of Available Seat is (%d,%d)\n",indexi,indexj);
            }
        }
        printf("\n");
    }
    printf("Number of available seat is %d \n", count);
    return 0;
}
```

Student Name: ________Wajiha Nazir__________________Roll No: ______25K-3021__________ Section: BSE-1B



```
Row 1:   1  0  1
Row 2:   0  0  1
Row 3:   1  1  0

Position of Available Seat is (1,2)

Position of Available Seat is (2,1)
Position of Available Seat is (2,2)

Position of Available Seat is (3,3)

Number of available seat is 4

--------------------------------
Process exited after 0.06347 seconds with return value 0
Press any key to continue . . .
```

Question # 5

A research team has placed sensors in a mountain valley arranged in a grid. Each sensor records the daily temperature. They need to find all "cold spots" - sensors where the temperature is lower than all of its immediate neighbors (to the north, south, east, and west). Sensors at the edge of the grid have fewer neighbors.

Grid of Temperatures (°C):

12, 15, 10, 9

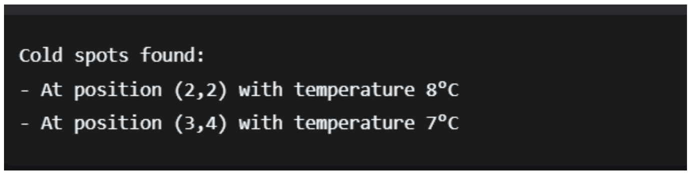
11, 8, 12, 13

14, 13, 9, 7

16, 11, 10, 8

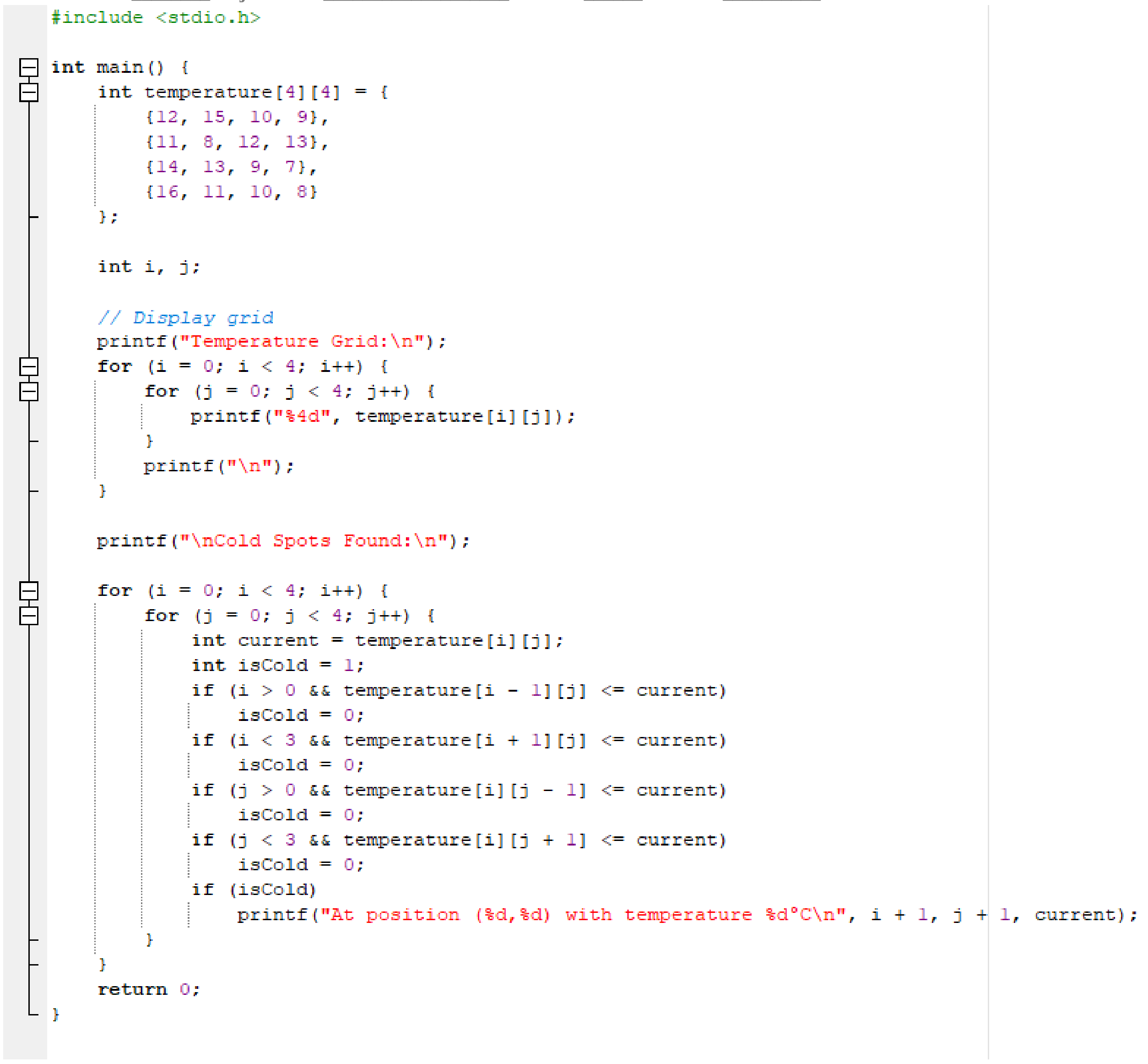
Find and report the location (row, column) and temperature of all cold spots. A cold spot must be colder than all its existing adjacent neighbors.

Output:

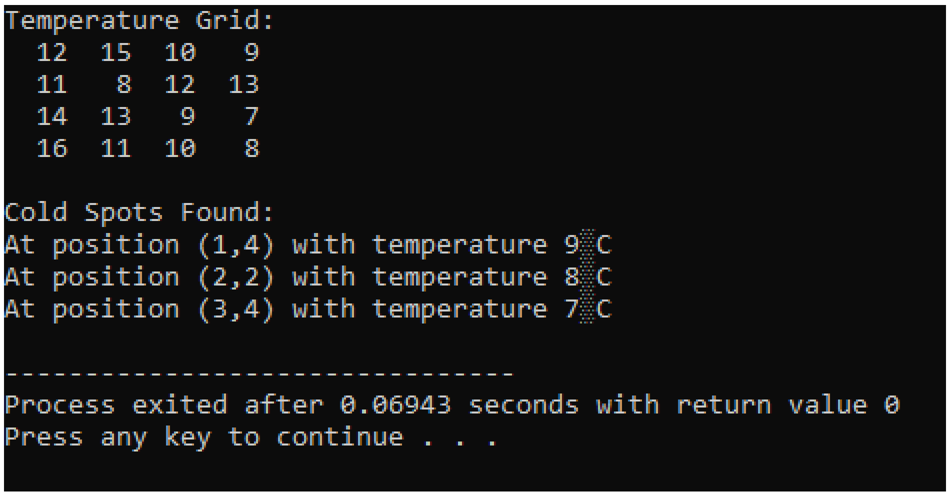


```
Cold spots found:
- At position (2,2) with temperature 8°C
- At position (3,4) with temperature 7°C
```

Student Name: ________Wajiha Nazir__________________**Roll No:** ______25K-3021__________ **Section:** BSE-1B



```c
#include <stdio.h>

int main() {
    int temperature[4][4] = {
        {12, 15, 10, 9},
        {11, 8, 12, 13},
        {14, 13, 9, 7},
        {16, 11, 10, 8}
    };

    int i, j;

    // Display grid
    printf("Temperature Grid:\n");
    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            printf("%4d", temperature[i][j]);
        }
        printf("\n");
    }

    printf("\nCold Spots Found:\n");

    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            int current = temperature[i][j];
            int isCold = 1;
            if (i > 0 && temperature[i - 1][j] <= current)
                isCold = 0;
            if (i < 3 && temperature[i + 1][j] <= current)
                isCold = 0;
            if (j > 0 && temperature[i][j - 1] <= current)
                isCold = 0;
            if (j < 3 && temperature[i][j + 1] <= current)
                isCold = 0;
            if (isCold)
                printf("At position (%d,%d) with temperature %d°C\n", i + 1, j + 1, current);
        }
    }
    return 0;
}
```

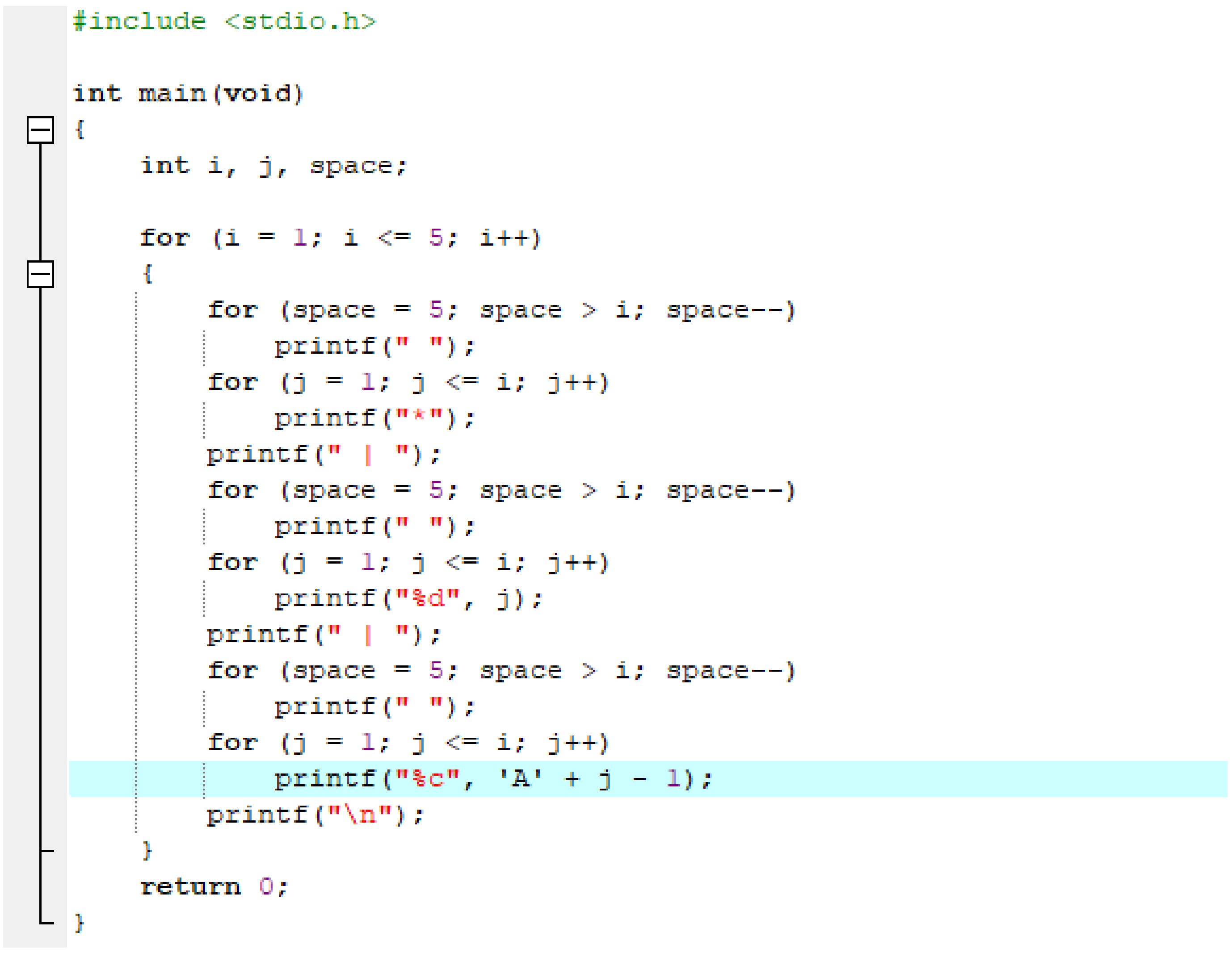


```
Temperature Grid:
  12  15  10   9
  11   8  12  13
  14  13   9   7
  16  11  10   8

Cold Spots Found:
At position (1,4) with temperature 9▒C
At position (2,2) with temperature 8▒C
At position (3,4) with temperature 7▒C

--------------------------------
Process exited after 0.06943 seconds with return value 0
Press any key to continue . . .
```

Student Name: ________Wajiha Nazir___________________Roll No: ______25K-3021__________ Section: BSE-1B

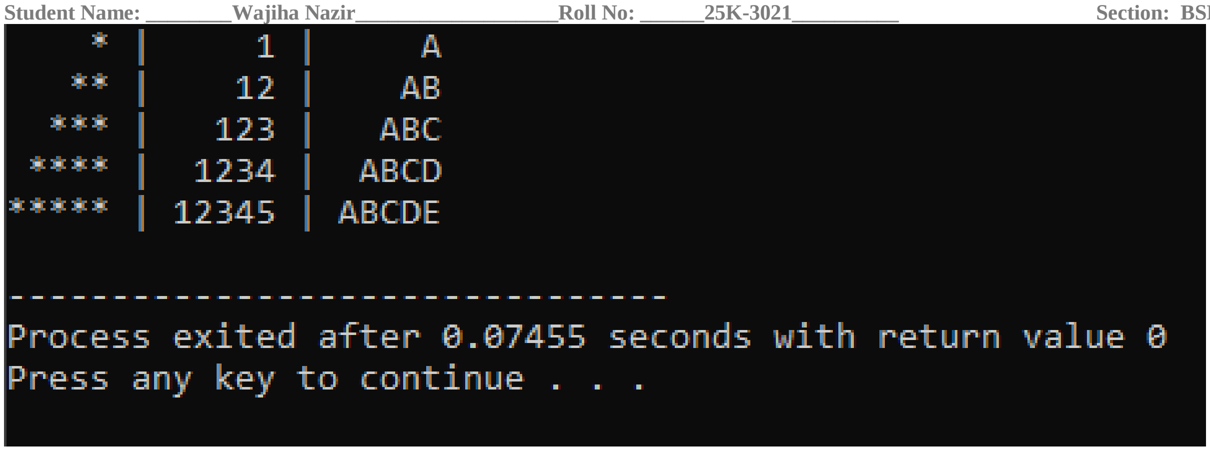
<u>Question # 6</u>

```
        *    |         1    |          A
      * *    |       1 2    |        A B
    * * *    |     1 2 3    |      A B C
  * * * *    |   1 2 3 4    |    A B C D
* * * * *    | 1 2 3 4 5    |  A B C D E
```



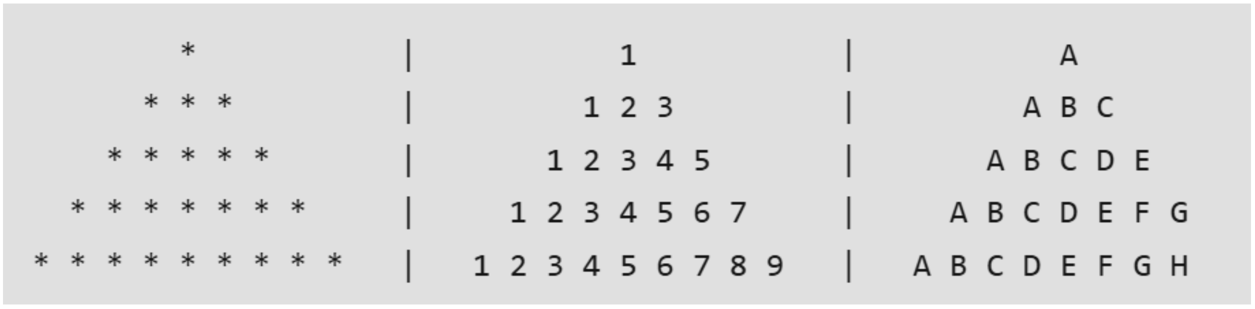
```
#include <stdio.h>

int main(void)
{
    int i, j, space;

    for (i = 1; i <= 5; i++)
    {
        for (space = 5; space > i; space--)
            printf(" ");
        for (j = 1; j <= i; j++)
            printf("*");
        printf("  |  ");
        for (space = 5; space > i; space--)
            printf(" ");
        for (j = 1; j <= i; j++)
            printf("%d", j);
        printf("  |  ");
        for (space = 5; space > i; space--)
            printf(" ");
        for (j = 1; j <= i; j++)
            printf("%c", 'A' + j - 1);
        printf("\n");
    }
    return 0;
}
```

Student Name: ________Wajiha Nazir__________________Roll No: ______25K-3021__________ Section: BSE-1B

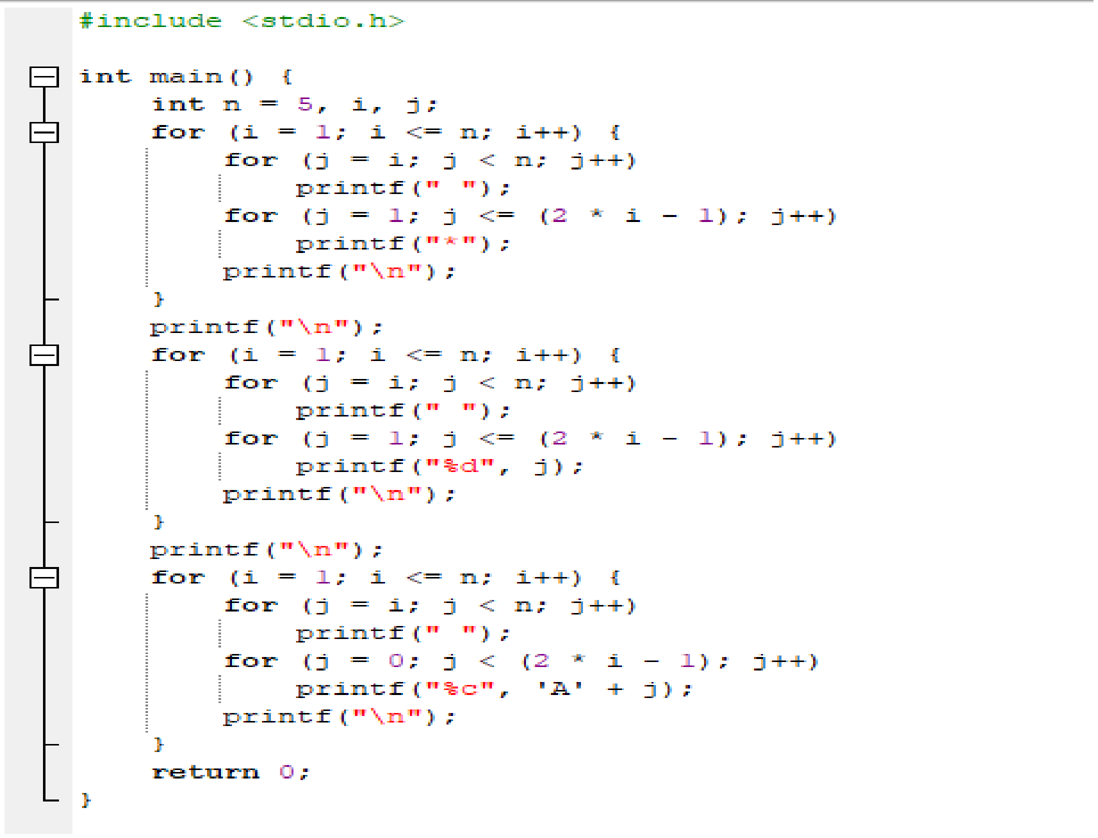


```
    * |     1 |     A
   ** |    12 |    AB
  *** |   123 |   ABC
 **** |  1234 |  ABCD
***** | 12345 | ABCDE

--------------------------------
Process exited after 0.07455 seconds with return value 0
Press any key to continue . . .
```

Question # 7

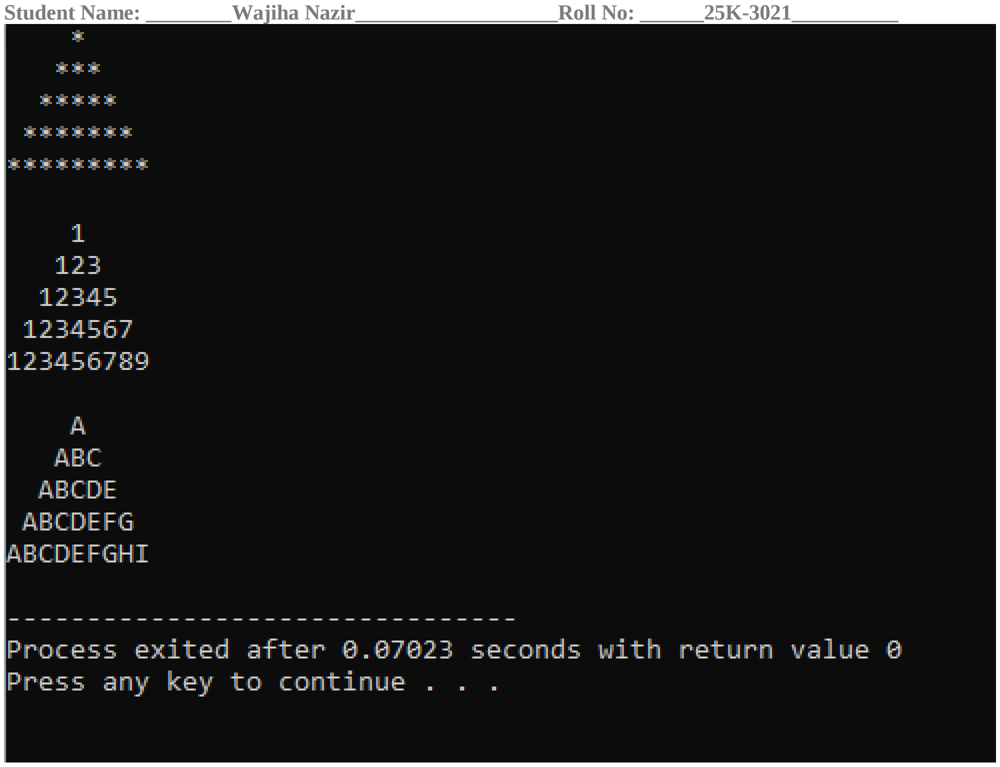


```
        *          |         1          |        A
      * * *        |       1 2 3        |      A B C
    * * * * *      |     1 2 3 4 5      |    A B C D E
  * * * * * * *    |   1 2 3 4 5 6 7    |  A B C D E F G
* * * * * * * * *  | 1 2 3 4 5 6 7 8 9  | A B C D E F G H
```



```c
#include <stdio.h>

int main() {
    int n = 5, i, j;
    for (i = 1; i <= n; i++) {
        for (j = i; j < n; j++)
            printf(" ");
        for (j = 1; j <= (2 * i - 1); j++)
            printf("*");
        printf("\n");
    }
    printf("\n");
    for (i = 1; i <= n; i++) {
        for (j = i; j < n; j++)
            printf(" ");
        for (j = 1; j <= (2 * i - 1); j++)
            printf("%d", j);
        printf("\n");
    }
    printf("\n");
    for (i = 1; i <= n; i++) {
        for (j = i; j < n; j++)
            printf(" ");
        for (j = 0; j < (2 * i - 1); j++)
            printf("%c", 'A' + j);
        printf("\n");
    }
    return 0;
}
```

Student Name: ________Wajiha Nazir___________________Roll No: ______25K-3021__________ Section: BSE-1B



```
    *
   ***
  *****
 *******
*********


    1
   123
  12345
 1234567
123456789

    A
   ABC
  ABCDE
 ABCDEFG
ABCDEFGHI

--------------------------------
Process exited after 0.07023 seconds with return value 0
Press any key to continue . . .
```

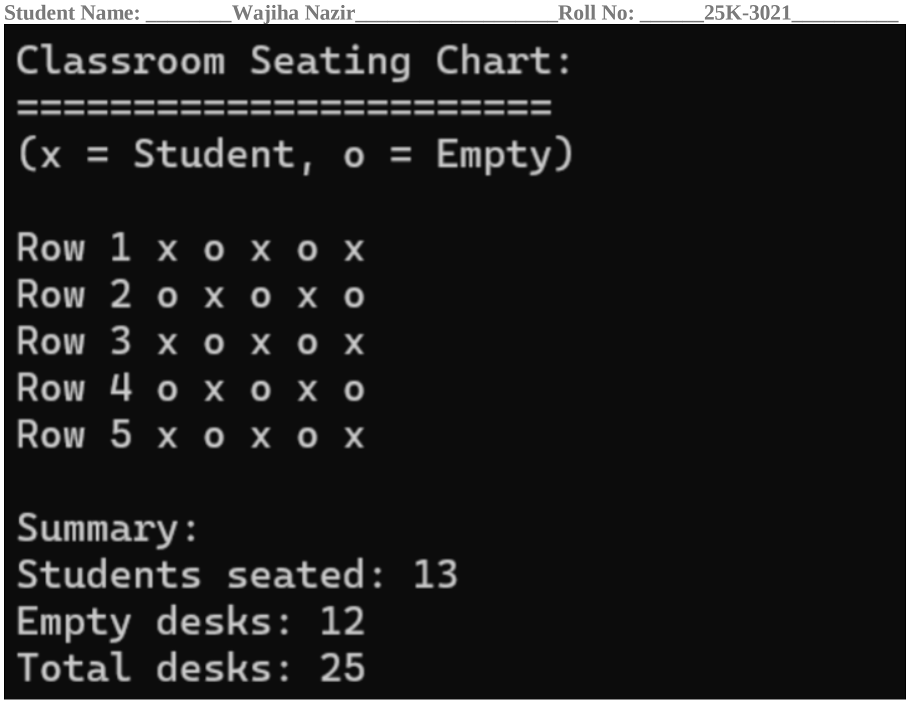
Question # 8

A teacher needs to organize seating for students in a classroom that has 5 rows with 5 desks in each row. Create a program that shows which desks are occupied and which are empty.

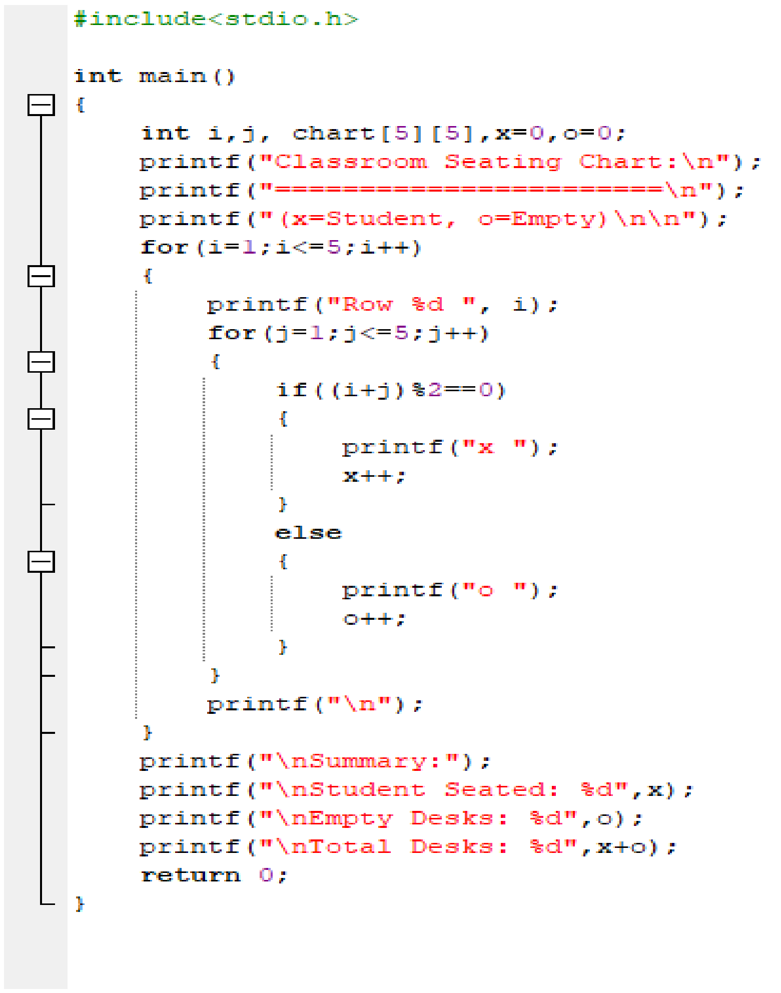
The seating should follow this pattern: students sit in every other desk, creating a checkerboard-style arrangement where occupied desks are separated by empty ones.

The program should display the final seating chart showing exactly where students are sitting.

Student Name: ________Wajiha Nazir___________________Roll No: ______25K-3021__________ Section: BSE-1B

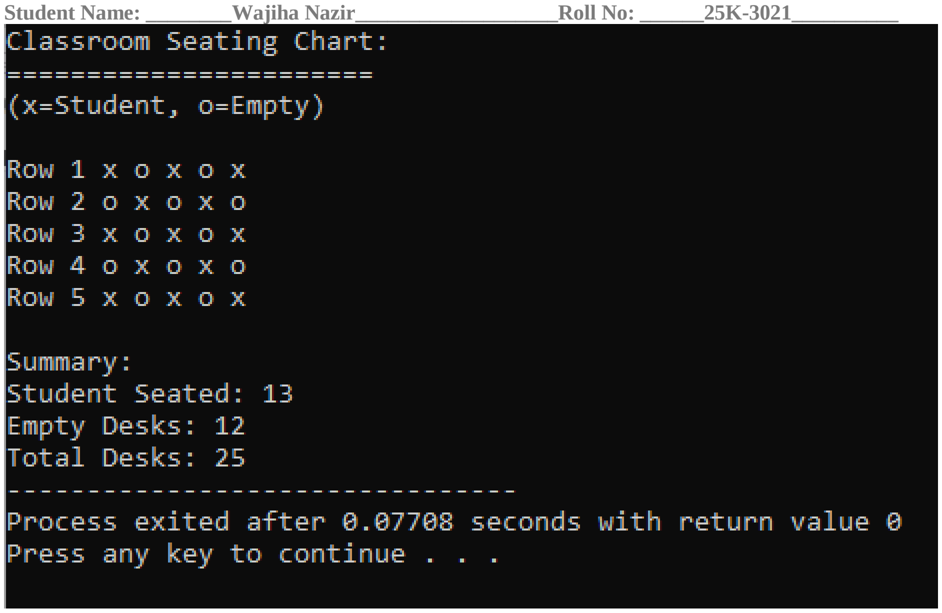


```
Classroom Seating Chart:
========================
(x = Student, o = Empty)

Row 1 x o x o x
Row 2 o x o x o
Row 3 x o x o x
Row 4 o x o x o
Row 5 x o x o x

Summary:
Students seated: 13
Empty desks: 12
Total desks: 25
```



```c
#include<stdio.h>

int main()
{
    int i,j, chart[5][5],x=0,o=0;
    printf("Classroom Seating Chart:\n");
    printf("========================\n");
    printf("(x=Student, o=Empty)\n\n");
    for(i=1;i<=5;i++)
    {
        printf("Row %d ", i);
        for(j=1;j<=5;j++)
        {
            if((i+j)%2==0)
            {
                printf("x ");
                x++;
            }
            else
            {
                printf("o ");
                o++;
            }
        }
        printf("\n");
    }
    printf("\nSummary:");
    printf("\nStudent Seated: %d",x);
    printf("\nEmpty Desks: %d",o);
    printf("\nTotal Desks: %d",x+o);
    return 0;
}
```

Student Name: ________Wajiha Nazir__________________Roll No: ______25K-3021__________ Section: BSE-1B



```
Classroom Seating Chart:
=======================
(x=Student, o=Empty)

Row 1 x o x o x
Row 2 o x o x o
Row 3 x o x o x
Row 4 o x o x o
Row 5 x o x o x

Summary:
Student Seated: 13
Empty Desks: 12
Total Desks: 25
--------------------------------
Process exited after 0.07708 seconds with return value 0
Press any key to continue . . .
```