Student Name: _____Emaan Binte Shams_____Roll No: _____25K-3082_____ 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.

lab8Q1.c

```c
#include <stdio.h>
main(){
    int Marks[3][4] = {{85, 92, 78, 90},{88, 76, 95, 84},{90, 85, 88, 92}};
    float sum;
    int i,j;
    for(i=0;i<3;i++){
        sum = 0.00;
        for(j=0;j<4;j++){
            sum += Marks[i][j];
        }
        printf("Average for class %d is: %f\n",i+1,(sum/4.00));
    }
}
```

```
Average for class 1 is: 86.250000
Average for class 2 is: 85.750000
Average for class 3 is: 88.750000
```

Student Name: _____Emaan Binte Shams_____Roll No: _____25K-3082_____ Section: BSE-1B

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.

lab8Q1.c lab8Q2.c

```c
#include<stdio.h>
main(){
    int Photos[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 year,month;
    for(year=0;year<2;year++){
        for(month=0;month<12;month++){
            printf("Year %d,Month %d: %d photos\n", year + 1, month + 1,Photos[year][month]);
        }
        printf("\n");
    }
}
```

```
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: _____Emaan Binte Shams_____Roll No: _____25K-3082_____ 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

lab8Q1.c lab8Q2.c lab8Q3.c

```c
#include<stdio.h>
main(){
    int i,j,whitecount = 0;
    int OriginalImage[4][4] = {{1, 0, 1, 0},{0, 1, 0, 1},{1, 1, 0, 0},{0, 0, 1, 1}};
    int InvertedImage[4][4];
    for(i=0;i<4;i++){
        for(j=0;j<4;j++){
            if(OriginalImage[i][j] == 1){
                whitecount += 1;
            }
        }
    }
    for(i=0;i<4;i++){
        for(j=0;j<4;j++){
            if(OriginalImage[i][j] == 0){
                InvertedImage[i][j] = 1;
            }
            else{
                InvertedImage[i][j] = 0;
            }
        }
    }
    printf("Original Image:\t\tInverted Image:\n");
    for(i=0;i<4;i++){
        for(j=0;j<4;j++){
            printf("%2d ", OriginalImage[i][j]);
        }
        printf("\t\t");
        for(j=0;j<4;j++){
            printf("%2d ", InvertedImage[i][j]);
        }
        printf("\n");
    }
    printf("\n\nnumber of white pixels in the original image: %d", whitecount);
}
```

```
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


number of white pixels in the original image: 8
--------------------------------
```

Student Name: _____Emaan Binte Shams_____Roll No: _____25K-3082_____ 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 cinema[3][3] = {{1, 0, 1}, {0, 0, 1}, {1, 1, 0}};
    int availableCount = 0;
    printf("Available seats:\n");
    for (int row = 0; row < 3; row++){
        for (int seat = 0; seat < 3; seat++) {
            if (cinema[row][seat] == 0) {
                availableCount++;
                printf("Row %d, Seat %d\n", row + 1, seat + 1);
            }
        }
    }
    printf("\nTotal available seats: %d\n", availableCount);
    return 0;
}
```

```
Available seats:
Row 1, Seat 2
Row 2, Seat 1
Row 2, Seat 2
Row 3, Seat 3

Total available seats: 4
```

Student Name: _____Emaan Binte Shams_____Roll No: _____25K-3082_____ Section: BSE-1B

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.

[*] lab8q5.cpp

```
#include <stdio.h>

int main() {
    int temp[4][4] = {{12, 15, 10,  9},{11,  8, 12, 13},{14, 13,  9,  7},{16, 11, 10,  8}};
    int i, j;
    printf("Cold spots found:\n");

    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            int current = temp[i][j];
            int isCold = 1;

            if (i > 0 && temp[i - 1][j] <= current) isCold = 0;
            if (i < 3 && temp[i + 1][j] <= current) isCold = 0;
            if (j > 0 && temp[i][j - 1] <= current) isCold = 0;
            if (j < 3 && temp[i][j + 1] <= current) isCold = 0;
            if (isCold)
            {
                printf("Row %d, Column %d: %d°C\n", i + 1, j + 1, current);
            }
        }
    }
    return 0;
}
```

```
Cold spots found:
Row 1, Column 4: 9°C
Row 2, Column 2: 8°C
Row 3, Column 4: 7°C
```

Student Name: _____Emaan Binte Shams_____Roll No: _____25K-3082_____ Section: BSE-1B

Question # 6

```
        *     |          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
```

```c
#include <stdio.h>
int main() {
    int i, j;
    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= i; j++) {
            printf("*");
        }
        printf("\t");
        for (j = 1; j <= i; j++) {
            printf("%d", j);
        }
        printf("\t");
        for (j = 1; j <= i; j++) {
            printf("%c", 'A' + j - 1);
        }
        printf("\n");
    }

    return 0;
}
```

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
```

Student Name: _____Emaan Binte Shams_____Roll No: _____25K-3082_____ Section: BSE-1B

```c
#include <stdio.h>
int main() {
    int i, j;
    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= i; j++) {
            printf("*");
        }
        printf("\t");
        for (j = 1; j <= (2 * i - 1); j++) {
            printf("%d", j);
        }
        printf("\t");
        for (j = 1; j <= (2 * i - 1); j++) {
            printf("%c", 'A' + j - 1);
        }
        printf("\n");
    }
    return 0;
}
```

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.

```c
#include <stdio.h>
int main() {
    int i, j, rows = 5, cols = 5;
    printf("Seating Chart (O = Occupied, E = Empty):\n\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            if ((i + j) % 2 == 0) {
                printf(" O ");
            } else {
                printf(" E ");
            }
        }
        printf("\n");
    }
    return 0;
}
```

```
Seating Chart (O = Occupied, E = Empty):

 O  E  O  E  O
 E  O  E  O  E
 O  E  O  E  O
 E  O  E  O  E
 O  E  O  E  O
```