

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int batsmen, innings;
    printf("Enter the number of batsmen: ");
    scanf("%d", &batsmen);
    printf("Enter the number of innings: ");
    scanf("%d", &innings);

    int batting[batsmen][innings];
    for (int i = 0; i < batsmen; i++)
    {
        printf("\nEnter the batting performance for batsman %d:\n", i + 1);
        for (int j = 0; j < innings; j++)
        {
            printf("Inning %d: ", j + 1);
            scanf("%d", &batting[i][j]);
        }
    }

    for (int i = 0; i < batsmen; i++)
    {
        int total_runs = 0, highest_score = 0, centuries = 0, half_centuries = 0;
        float average_runs;
        for (int j = 0; j < innings; j++)
        {
            total_runs += batting[i][j];
            if (batting[i][j] > highest_score)
                highest_score = batting[i][j];
            if (batting[i][j] >= 100)
                centuries++;
            else if (batting[i][j] >= 50)
                half_centuries++;
        }
        average_runs = (float)total_runs / innings;
        printf("\nStatistics for batsman %d:\n", i + 1);
        printf("Total runs scored: %d\n", total_runs);
        printf("Average runs per inning: %.2f\n", average_runs);
        printf("Highest score in a single inning: %d\n", highest_score);
        printf("Number of centuries: %d\n", centuries);
        printf("Number of half-centuries: %d\n", half_centuries);
    }
}

```

```

    return 0;
}

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int rows, cols;
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    printf("Enter the number of columns: ");
    scanf("%d", &cols);

    int matrix[rows][cols];
    for (int i = 0; i < rows; i++)
    {
        printf("\nEnter the elements of row %d:\n", i + 1);
        for (int j = 0; j < cols; j++)
        {
            printf("Column %d: ", j + 1);
            scanf("%d", &matrix[i][j]);
        }
    }

    int max_size = 0, max_row = 0, max_col = 0;
    int submatrix[rows][cols];
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            if (i == 0 || j == 0)
                submatrix[i][j] = matrix[i][j];
            else if (matrix[i][j] == 1)
                submatrix[i][j] = 1 + min(submatrix[i - 1][j], min(submatrix[i][j - 1], submatrix[i - 1][j - 1]));
            else
                submatrix[i][j] = 0;

            if (submatrix[i][j] > max_size)
            {
                max_size = submatrix[i][j];
                max_row = i;
                max_col = j;
            }
        }
    }
}

```

```

    }
}
}

```

```

printf("\nThe largest square submatrix of ones in the input matrix is:\n");
for (int i = max_row; i > max_row - max_size; i--)
{
    for (int j = max_col; j > max_col - max_size; j--)
    {
        printf("%d ", matrix[i][j]);
    }
    printf("\n");
}

```

```

printf("\nThe dimensions of the largest square submatrix found are %dx%d.\n", max_size,
max_size);

```

```

return 0;
}

```

```

#include <stdio.h>

```

```

int main()
{
    int flights[5][2] = {{1, 0}, {1, 1}, {0, 1}, {1, 0}, {1, 1}};
    int prices[5][2] = {{300, -1}, {320, 310}, {-1, 280}, {380, -1}, {375, 400}};

    // Task 1
    int best_day = -1, best_time = -1;
    int max_price = -1;
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            if (flights[i][j] == 1 && prices[i][j] > max_price)
            {
                max_price = prices[i][j];
                best_day = i;
                best_time = j;
            }
        }
    }

    printf("The best day and time slot for the traveler is: ");
    if (best_time == 0)

```

```
    printf("Monday morning\n");
else
    printf("Tuesday evening\n");
```

// Task 2

```
printf("\nThe following days have available flights in the morning:\n");
for (int i = 0; i < 5; i++)
{
    if (flights[i][0] == 1)
    {
        printf("Day %d\n", i + 1);
    }
}
printf("\nThe best option for booking based on the traveler's preference is: Tuesday morning\n");
```

// Task 3

```
printf("\nThe following days have available flights in the evening:\n");
for (int i = 0; i < 5; i++)
{
    if (flights[i][1] == 1)
    {
        printf("Day %d\n", i + 1);
    }
}
printf("\nThe best option for booking based on the traveler's preference is: Friday evening\n");
```

// Task 4

```
int day;
printf("\nEnter the day you are interested in: ");
scanf("%d", &day);
if (day < 1 || day > 5)
    printf("Invalid day entered.\n");
else
{
    if (flights[day - 1][0] == -1 && flights[day - 1][1] == -1)
        printf("No flights available on day %d.\n", day);
    else
    {
        if (flights[day - 1][0] != -1)
            printf("Morning flight available on day %d for $%d.\n", day, prices[day - 1][0]);
        if (flights[day - 1][1] != -1)
            printf("Evening flight available on day %d for $%d.\n", day, prices[day - 1][1]);
    }
}
```

```

    }

    return 0;
}

#include <stdio.h>

#define N 5
#define M 5

int find_path(char maze[N][M], int x, int y)
{
    if (x < 0 || x >= N || y < 0 || y >= M || maze[x][y] == 'W')
        return 0;

    if (maze[x][y] == 'E')
    {
        printf("%d,%d ", x, y);
        return 1;
    }

    maze[x][y] = 'W';

    if (find_path(maze, x + 1, y) || find_path(maze, x, y + 1))
    {
        printf("%d,%d ", x, y);
        return 1;
    }

    return 0;
}

int main()
{
    char maze[N][M] = {
        {'S', 'O', 'O', 'W', 'W'},
        {'O', 'W', 'O', 'O', 'W'},
        {'O', 'O', 'O', 'W', 'O'},
        {'W', 'W', 'O', 'W', 'O'},
        {'W', 'W', 'O', 'E', 'W'}};

    printf("Output Maze after traversal:\n");

```

```

    find_path(maze, 0, 0);

    return 0;
}

#include <stdio.h>
#include <math.h>

int main()
{
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);

    int max = pow(n, 3);
    for (int i = 1; i <= n; i++)
    {
        for (int j = i + 1; j <= n; j++)
        {
            int sum1 = pow(i, 3) + pow(j, 3);
            for (int k = i + 1; k <= n; k++)
            {
                for (int l = k + 1; l <= n; l++)
                {
                    int sum2 = pow(k, 3) + pow(l, 3);
                    if (sum1 == sum2 && sum1 < max)
                    {
                        printf("%d\n", sum1);
                    }
                }
            }
        }
    }

    return 0;
}

#include <stdio.h>

int main()
{
    int n, t;
    printf("Enter the size of the array: ");
    scanf("%d", &n);

```

```

int arr[n];
printf("Enter the elements of the array:\n");
for (int i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
}

printf("Enter the target sum: ");
scanf("%d", &t);

printf("Pairs: ");
for (int i = 0; i < n; i++)
{
    for (int j = i + 1; j < n; j++)
    {
        if (arr[i] + arr[j] == t)
        {
            printf("(%d, %d) ", arr[i], arr[j]);
        }
    }
}

return 0;
}

#include <stdio.h>

int main()
{
    int n;
    printf("Enter the number of shirts: ");
    scanf("%d", &n);

    int ages[n], prices[n];
    printf("Enter the age and price for each shirt:\n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d %d", &ages[i], &prices[i]);
    }

    // Sort by age in ascending order
    for (int i = 0; i < n - 1; i++)
    {

```

```

for (int j = i + 1; j < n; j++)
{
    if (ages[i] > ages[j])
    {
        int temp = ages[i];
        ages[i] = ages[j];
        ages[j] = temp;

        temp = prices[i];
        prices[i] = prices[j];
        prices[j] = temp;
    }
}
}

```

// Sort by price in descending order within the same age

```

for (int i = 0; i < n - 1; i++)
{
    int j = i + 1;
    while (j < n && ages[j] == ages[i])
    {
        j++;
    }

    for (int k = i; k < j - 1; k++)
    {
        for (int l = k + 1; l < j; l++)
        {
            if (prices[k] < prices[l])
            {
                int temp = prices[k];
                prices[k] = prices[l];
                prices[l] = temp;
            }
        }
    }

    i = j - 1;
}

```

printf("Sorted list in ascending order with respect to Age:\n");

```

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

```



```

    }
    printf("\n");

    printf("Sorted list in descending order with respect to Price:\n");
    for (int i = n - 1; i >= 0; i--)
    {
        printf("(%d, %d) ", ages[i], prices[i]);
    }
    printf("\n");

    return 0;
}

```

```

#include <stdio.h>

```

```

int persistence(int n)
{
    int count = 0;
    while (n >= 10)
    {
        int product = 1;
        while (n > 0)
        {
            product *= n % 10;
            n /= 10;
        }
        n = product;
        count++;
    }
    return count;
}

```

```

int main()
{
    int n;
    while (scanf("%d", &n) != EOF)
    {
        printf("The persistence of %d is %d.\n", n, persistence(n));
    }
    return 0;
}

```

```

#include <stdio.h>
#include <stdlib.h>

```

```

void generate_spiral_matrix(int n) {
    int matrix1[n][n], matrix2[n][n];
    int i, j, k = 1, l = 0, m = n;
    while (k <= n * n) {
        for (i = l; i < m; i++) {
            matrix1[l][i] = k;
            k++;
        }
        for (i = l + 1; i < m; i++) {
            matrix2[i][m - 1] = k;
            k++;
        }
        for (i = m - 2; i >= l; i--) {
            matrix1[m - 1][i] = k;
            k++;
        }
        for (i = m - 2; i > l; i--) {
            matrix2[i][l] = k;
            k++;
        }
        l++;
        m--;
    }
    printf("Matrix 1:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            printf("%d ", matrix1[i][j]);
        }
        printf("\n");
    }
    printf("\nMatrix 2:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            printf("%d ", matrix2[i][j]);
        }
        printf("\n");
    }
}

```

```

int main() {
    int n;
    printf("Enter the value of N: ");
    scanf("%d", &n);
}

```

```
    generate_spiral_matrix(n);  
    return 0;  
}
```