**Agnim Gupta**

**2028083**

**A-23,CSSE**

**Question 1**

#include <stdio.h>

// Creating MACRO for finding the maximum number

#define max(x, y)(((x) > (y)) ? (x) : (y))

// Creating MACRO for finding the minimum number

#define min(x, y)(((x) < (y)) ? (x) : (y))

// Function to returns the minimum number of platforms required

int findPlatform(int arr[], int dep[], int n)

{

    // plat\_needed indicates number of platforms

    // needed at a time

    int plat\_needed = 1, result = 1;

    int i = 1, j = 0;

    // run a nested  loop to find overlap

    for (int i = 0; i < n; i++) {

        // minimum platform

        plat\_needed = 1;

        for (int j = i + 1; j < n; j++) {

            // check for overlap

            if ((arr[i] >= arr[j] && arr[i] <= dep[j]) ||

           (arr[j] >= arr[i] && arr[j] <= dep[i]))

                plat\_needed++;

        }

        // update result

        result = max(result, plat\_needed);

    }

    return result;

}

// Driver Code

int main()

{

    int arr[] = { 900, 940, 950, 1100, 1500, 1800 };

    int dep[] = { 910, 1200, 1120, 1130, 1900, 2000 };

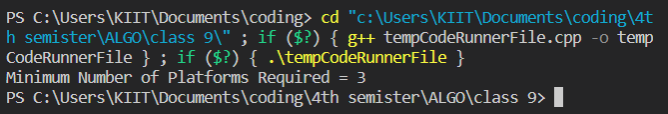
    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Minimum Number of Platforms Required = %d", findPlatform(arr, dep, n));

    return 0;

}

**Output**

****

**Question 2**

#include <stdio.h>

int chkPair(int A[], int size, int x)

{

    for (int i = 0; i < (size - 1); i++)

    {

        for (int j = (i + 1); j < size; j++)

        {

            if (A[i] + A[j] == x)

            {

                printf("Pair with a given sum %d is (%d, %d)\n", x, A[i], A[j]);

                return 1;

            }

        }

    }

    return 0;

}

int main(void)

{

    int size, x;

    printf("Enter the size of the array : ");

    scanf("%d", &size);

    int A[size];

    printf("Enter the elements of the array : \n");

    for(int i=0; i<size; i++)

    {

        scanf("%d", &A[i]);

    }

    printf("Enter the sum to be searched : ");

    scanf("%d", &x);

    if (chkPair(A, size, x))

    {

        printf("Valid pair exists\n");

    }

    else

    {

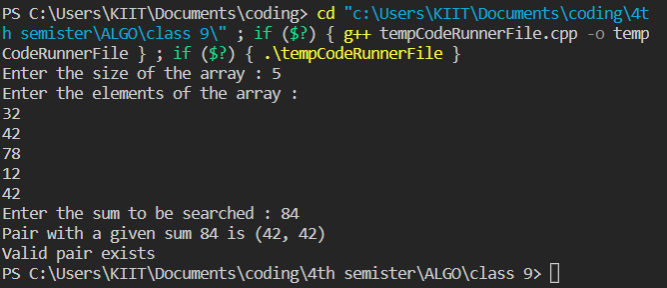
        printf("No valid pair exists for %d\n", x);

    }

    return 0;

}

**Output**

****