IE2072 [2023/JUL]-Foundations of Algorithms

IT21826368 – Nanayakkara Y.D.T.D

Assignment (30% of total assessments)

Contents

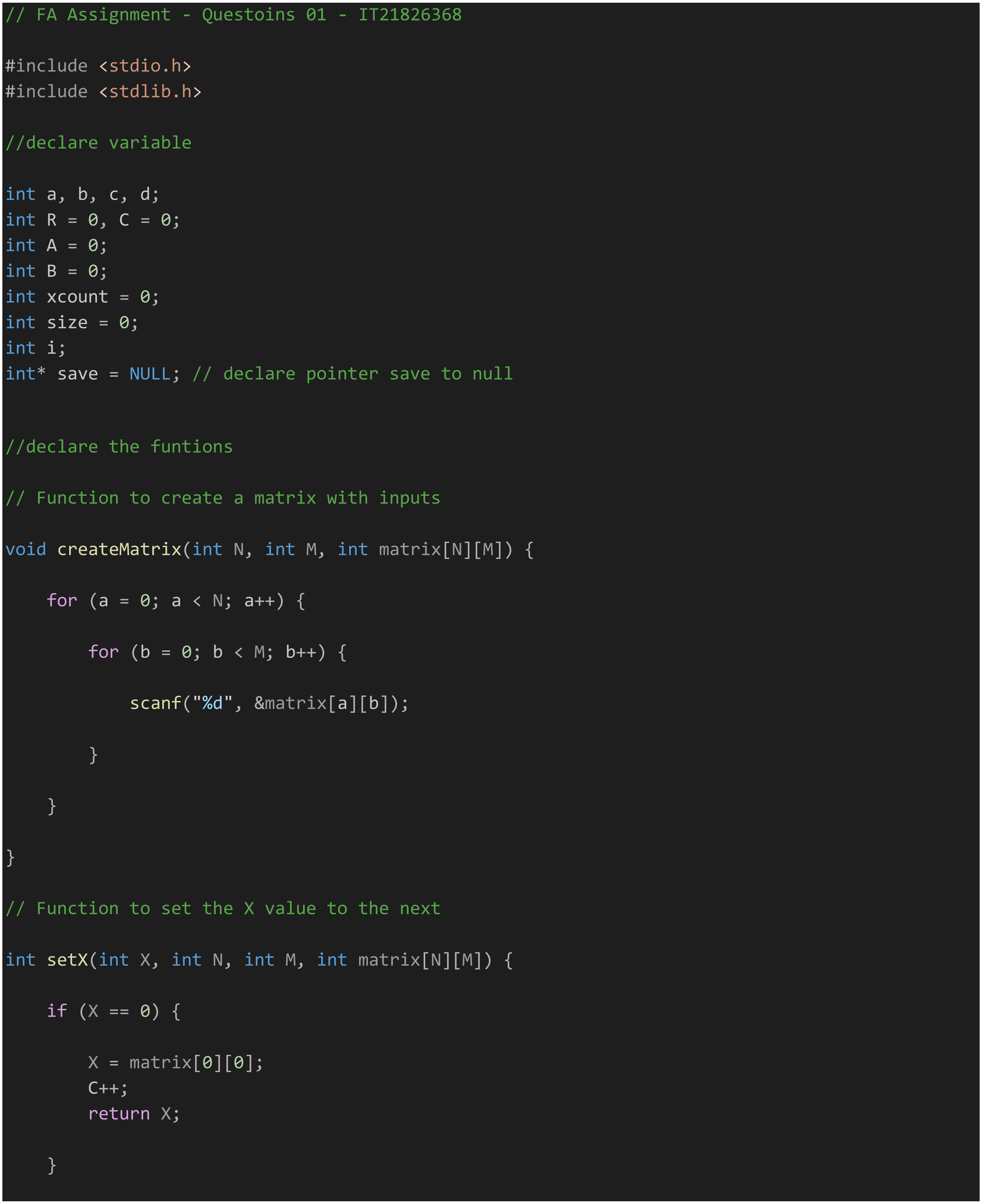
Question 1 (10 marks)

1. Source code (2 files)

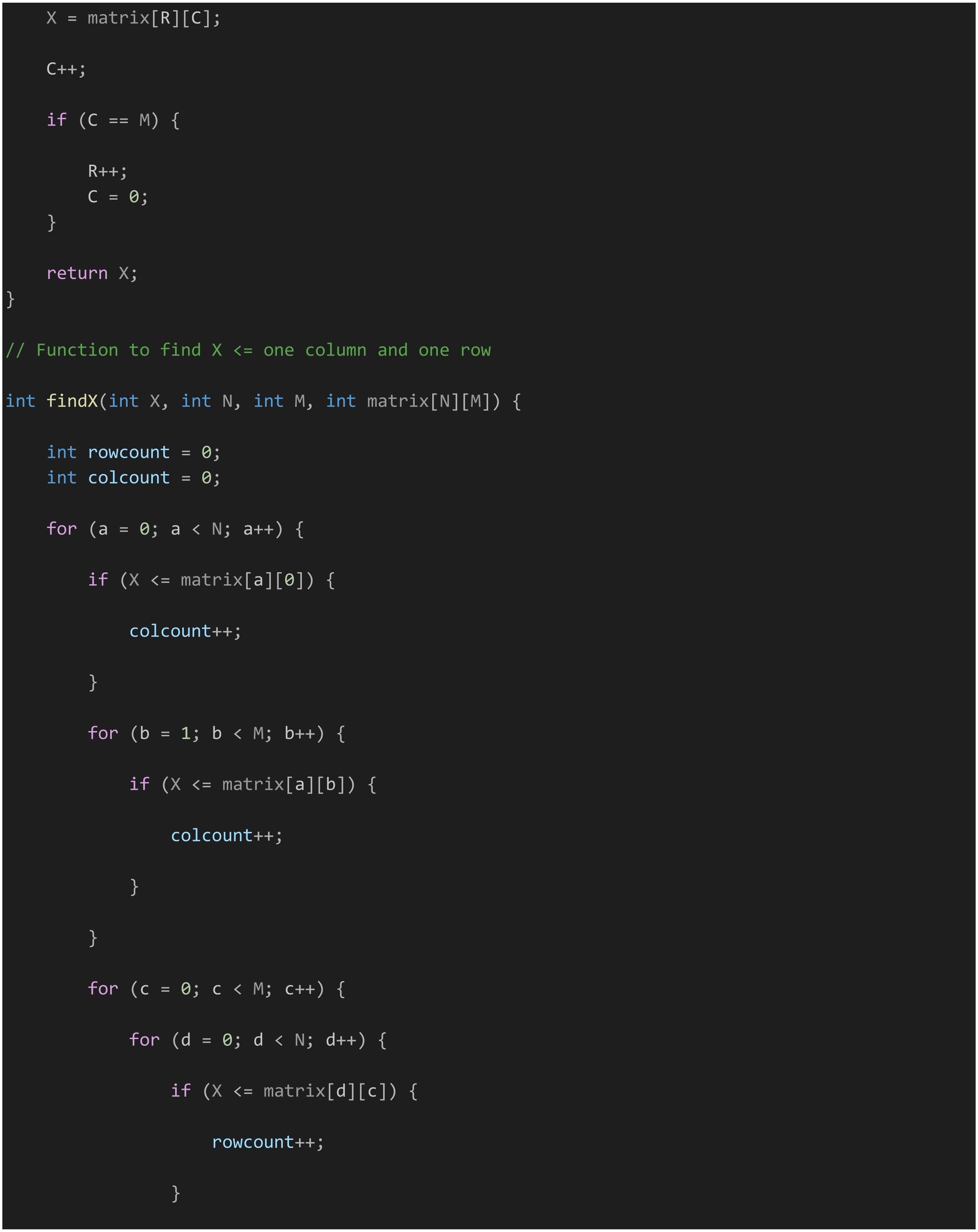
Main

```
// FA Assignment - Questions 01 - IT21826368

#include "Q01.h"

int main(){

    int N, M; //row and column for matrix.
    int X = 0;

    printf(" Enter N and M : ");

    scanf("%d %d", &N, &M);

    int matrix[N][M]; // Matrix


    //Calling the funtions.

    printf("  (if N and M are 2 : 1 2 3 4)\n Enter Matrix values : ");

    createMatrix(N, M, matrix);

    int ttl = (N * M);
    int count = 0;

    while(count != ttl) {

        X = setX(X, N, M, matrix);

        findX(X, N, M, matrix);

        count++;

    }

    size = findX(X, N, M, matrix);

    maxX(save, size);

    return 0;

}
```

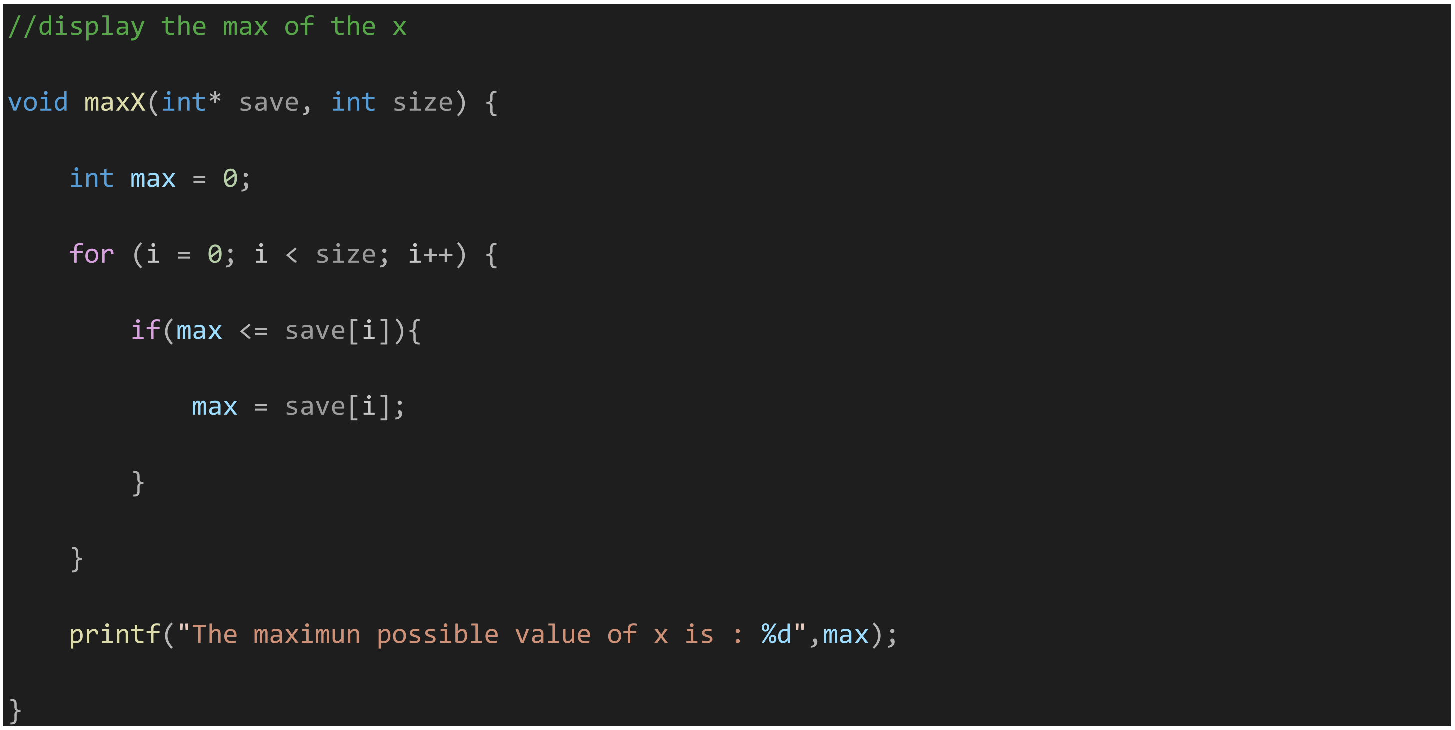
h file



```
// FA Assignment - Questoins 01 - IT21826368

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

//declare variable

int a, b, c, d;
int R = 0, C = 0;
int A = 0;
int B = 0;
int xcount = 0;
int size = 0;
int i;
int* save = NULL; // declare pointer save to null


//declare the funtions

// Function to create a matrix with inputs

void createMatrix(int N, int M, int matrix[N][M]) {

    for (a = 0; a < N; a++) {

        for (b = 0; b < M; b++) {

            scanf("%d", &matrix[a][b]);

        }

    }

}

// Function to set the X value to the next

int setX(int X, int N, int M, int matrix[N][M]) {

    if (X == 0) {

        X = matrix[0][0];
        C++;
        return X;

    }
```

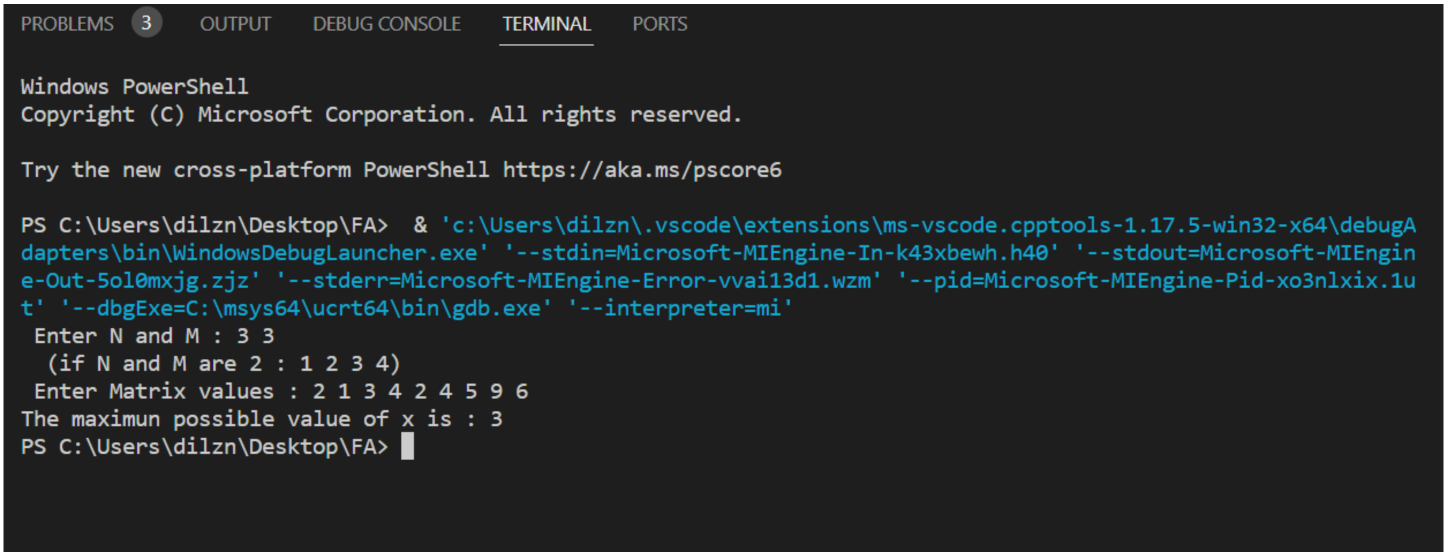


```
    X = matrix[R][C];

    C++;

    if (C == M) {

        R++;
        C = 0;
    }

    return X;
}

// Function to find X <= one column and one row

int findX(int X, int N, int M, int matrix[N][M]) {

    int rowcount = 0;
    int colcount = 0;

    for (a = 0; a < N; a++) {

        if (X <= matrix[a][0]) {

            colcount++;

        }

        for (b = 1; b < M; b++) {

            if (X <= matrix[a][b]) {

                colcount++;

            }

        }

        for (c = 0; c < M; c++) {

            for (d = 0; d < N; d++) {

                if (X <= matrix[d][c]) {

                    rowcount++;

                }
            }
            if (rowcount == N && colcount == M) {
                xcount++;
            }
            rowcount = 0;
        }
        colcount = 0;
    }
    if (xcount != 0) {
        saveX(X, &save, &size);
    }
    xcount = 0;
    return size;
}
//save the x to array
void saveX(int X, int** save, int* size) {
    *save = realloc(*save, (*size + 1) * sizeof(int));
    if (*save == NULL) {
        fprintf(stderr, "Memory allocation failed\n");
        exit(1);
    }
    (*save)[*size] = X; // Save the value into the updated array
    (*size)++;
}
```



```c
//display the max of the x

void maxX(int* save, int size) {

    int max = 0;

    for (i = 0; i < size; i++) {

        if(max <= save[i]){

            max = save[i];

        }

    }

    printf("The maximun possible value of x is : %d",max);

}
```

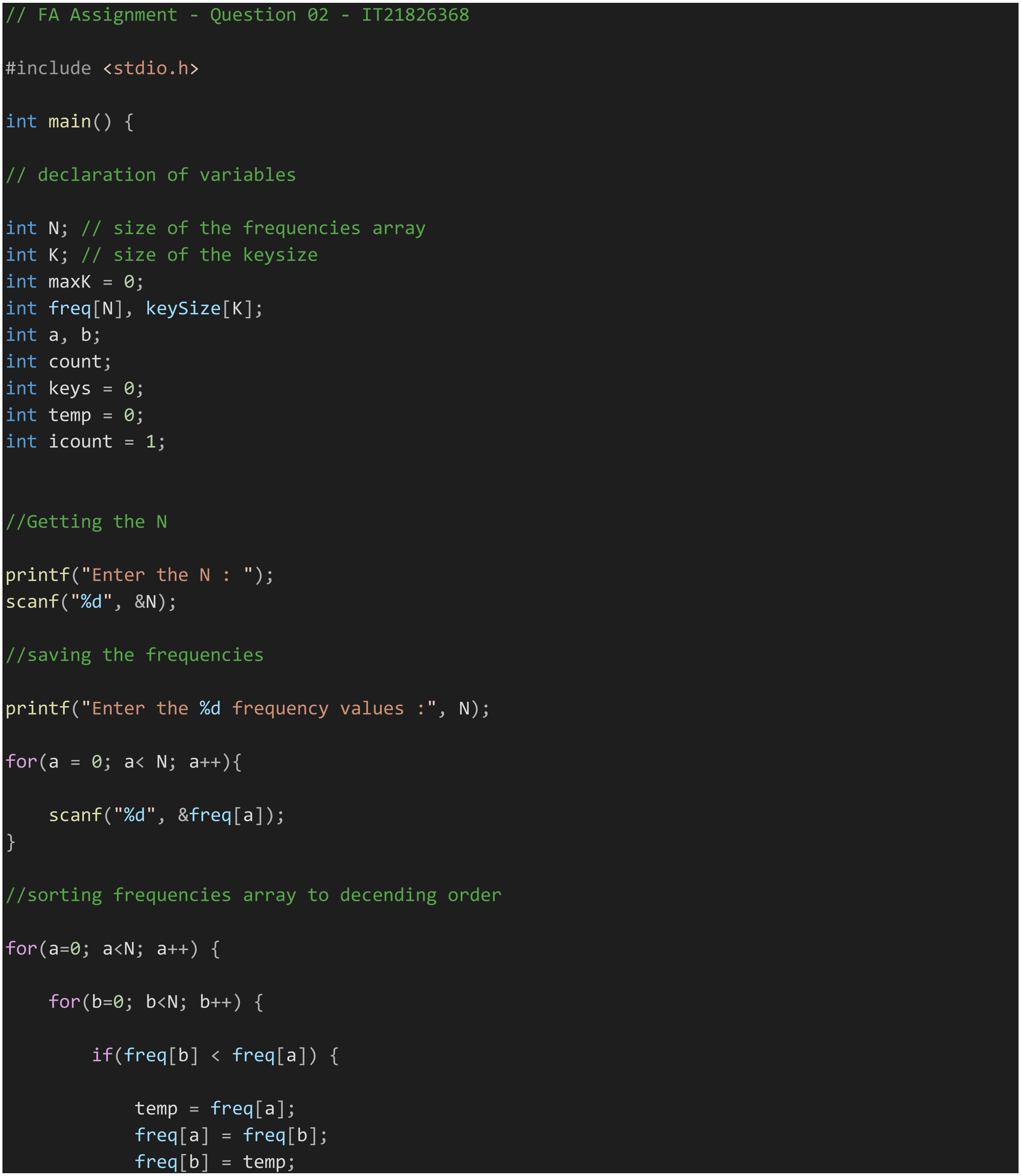
2. Output



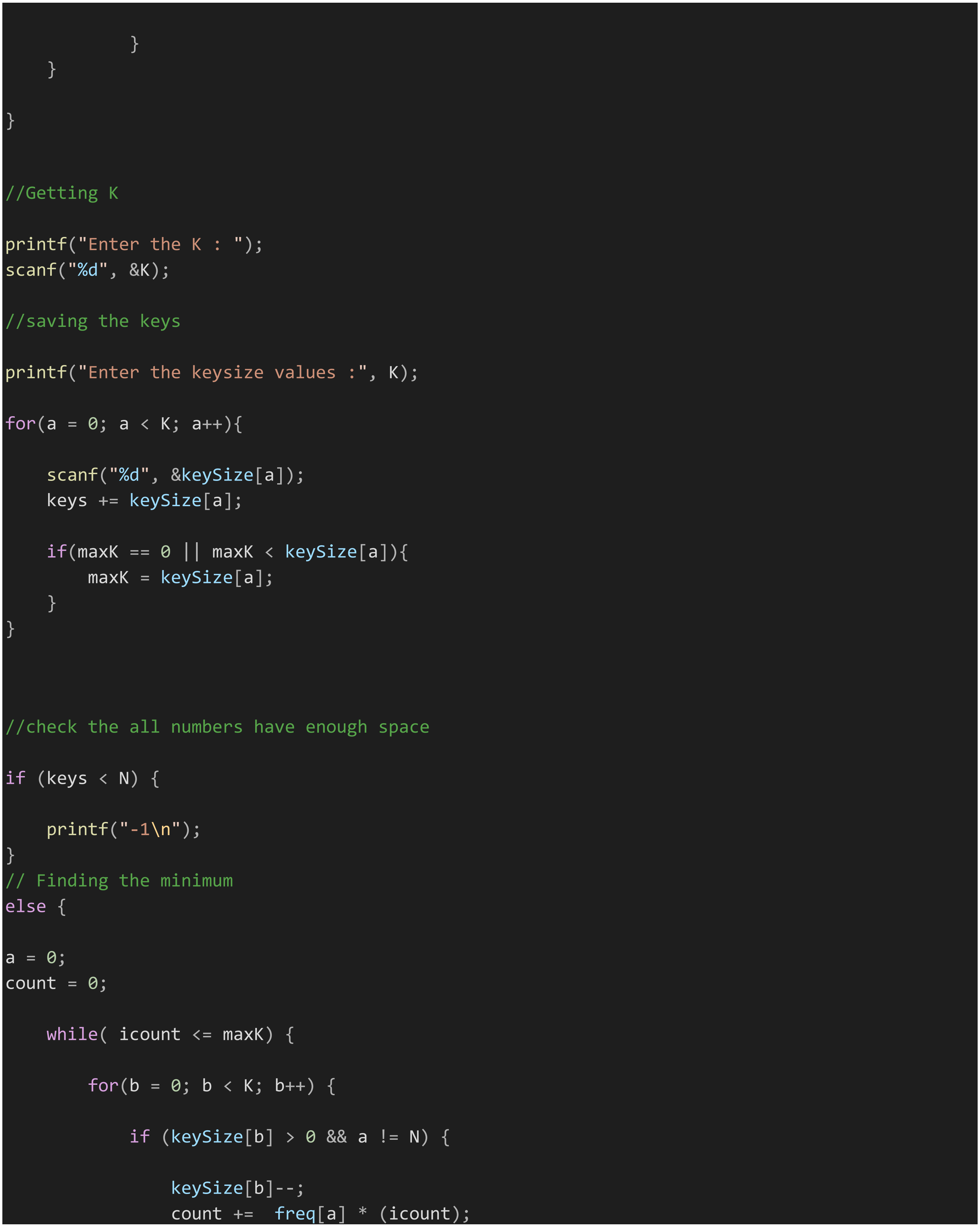
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\dilzn\Desktop\FA>  & 'c:\Users\dilzn\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-k43xbewh.h40' '--stdout=Microsoft-MIEngine-Out-5ol0mxjg.zjz' '--stderr=Microsoft-MIEngine-Error-vvai13d1.wzm' '--pid=Microsoft-MIEngine-Pid-xo3nlxix.1ut' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
 Enter N and M : 3 3
  (if N and M are 2 : 1 2 3 4)
 Enter Matrix values : 2 1 3 4 2 4 5 9 6
The maximun possible value of x is : 3
PS C:\Users\dilzn\Desktop\FA> 
```

2. Question 2 (20 marks)

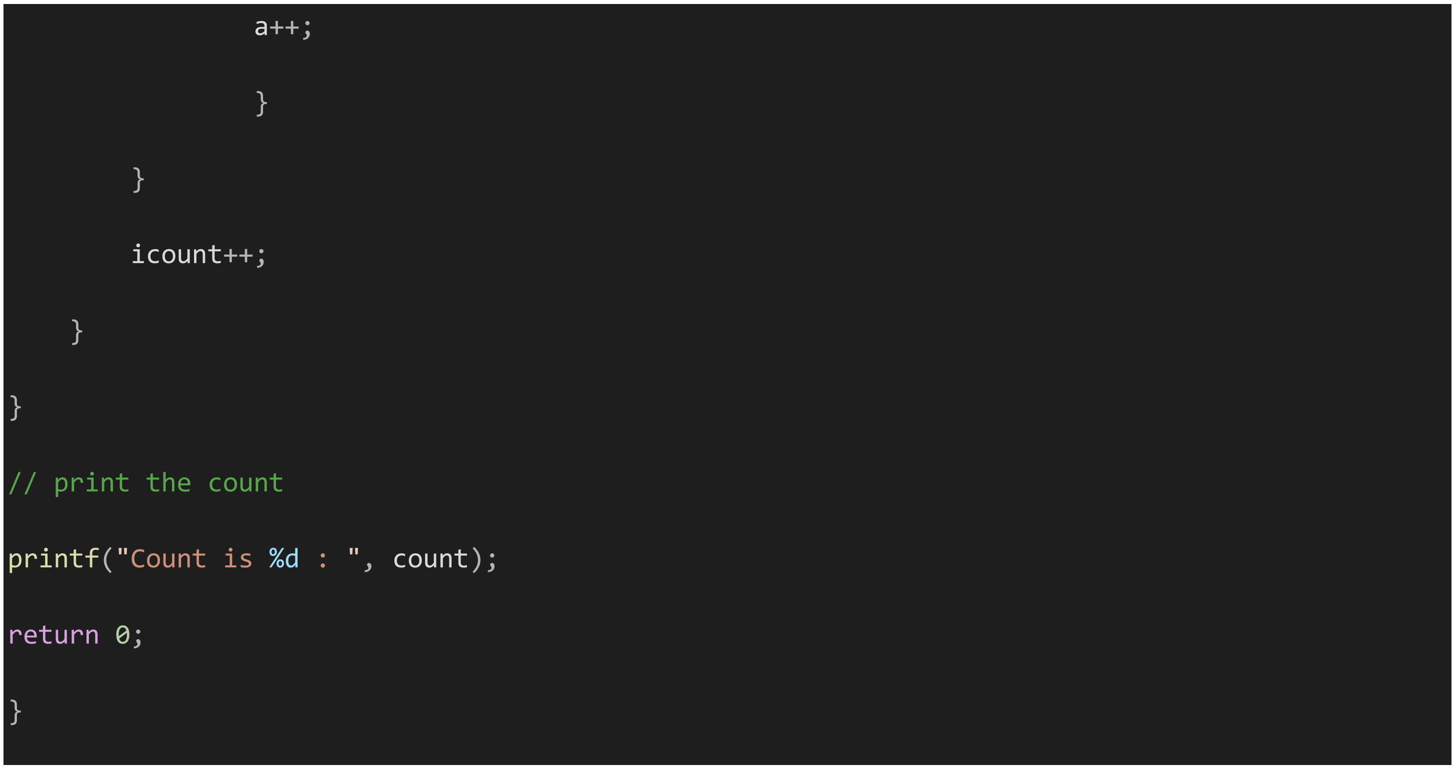
1. Source code



```
// FA Assignment - Question 02 - IT21826368

#include <stdio.h>

int main() {

// declaration of variables

int N; // size of the frequencies array
int K; // size of the keysize
int maxK = 0;
int freq[N], keySize[K];
int a, b;
int count;
int keys = 0;
int temp = 0;
int icount = 1;


//Getting the N

printf("Enter the N : ");
scanf("%d", &N);

//saving the frequencies

printf("Enter the %d frequency values :", N);

for(a = 0; a< N; a++){

    scanf("%d", &freq[a]);
}

//sorting frequencies array to decending order

for(a=0; a<N; a++) {

    for(b=0; b<N; b++) {

        if(freq[b] < freq[a]) {

            temp = freq[a];
            freq[a] = freq[b];
            freq[b] = temp;
```

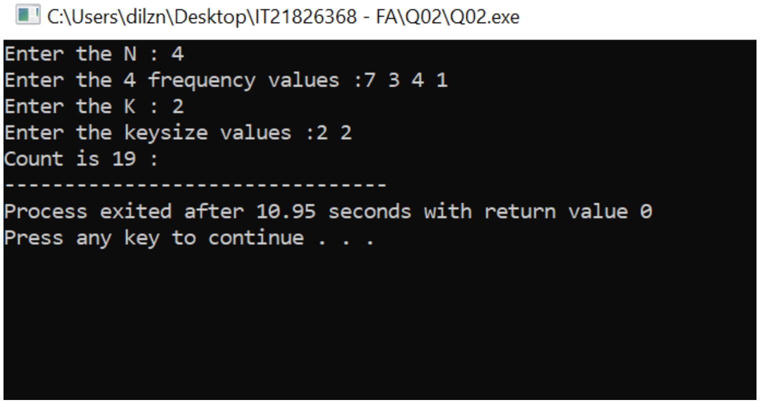


```
            }
    }

}


//Getting K

printf("Enter the K : ");
scanf("%d", &K);

//saving the keys

printf("Enter the keysize values :", K);

for(a = 0; a < K; a++){

    scanf("%d", &keySize[a]);
    keys += keySize[a];

    if(maxK == 0 || maxK < keySize[a]){
        maxK = keySize[a];
    }
}



//check the all numbers have enough space

if (keys < N) {

    printf("-1\n");
}
// Finding the minimum
else {

a = 0;
count = 0;

    while( icount <= maxK) {

        for(b = 0; b < K; b++) {

            if (keySize[b] > 0 && a != N) {

                keySize[b]--;
                count +=  freq[a] * (icount);
```



```
                a++;

                }

        }

        icount++;

    }

}

// print the count

printf("Count is %d : ", count);

return 0;

}
```

2. Output



C:\Users\dilzn\Desktop\IT21826368 - FA\Q02\Q02.exe

```
Enter the N : 4
Enter the 4 frequency values :7 3 4 1
Enter the K : 2
Enter the keysize values :2 2
Count is 19 :
--------------------------------
Process exited after 10.95 seconds with return value 0
Press any key to continue . . .
```