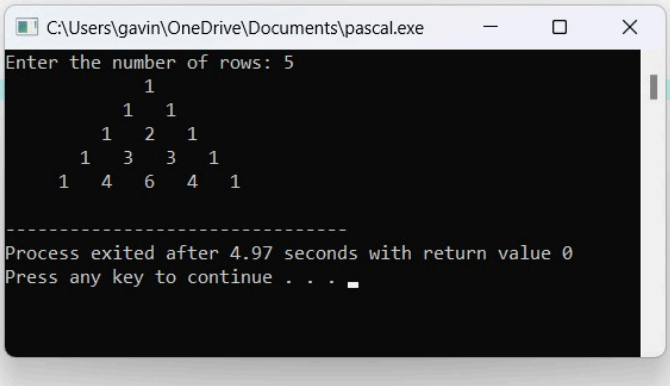
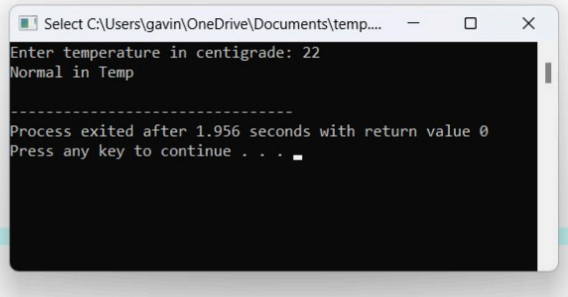
read and print.cpp | wages.cpp | roll number.cpp | temp.cpp | alphabet.cpp | [*] harmonic.cpp | pascal.cpp

```c
#include <stdio.h>

int main() {
    int rows, coef = 1, space, i, j;
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    for(i = 0; i < rows; i++) {
        for(space = 1; space <= rows - i; space++)
            printf("  ");
        for(j = 0; j <= i; j++) {
            if (j == 0 || i == 0)
                coef = 1;
            else
                coef = coef * (i - j + 1) / j;
            printf("%4d", coef);
        }
        printf("\n");
    }
    return 0;
}

```



read and print.cpp | wages.cpp | roll number.cpp | temp.cpp | [*] Untitled6

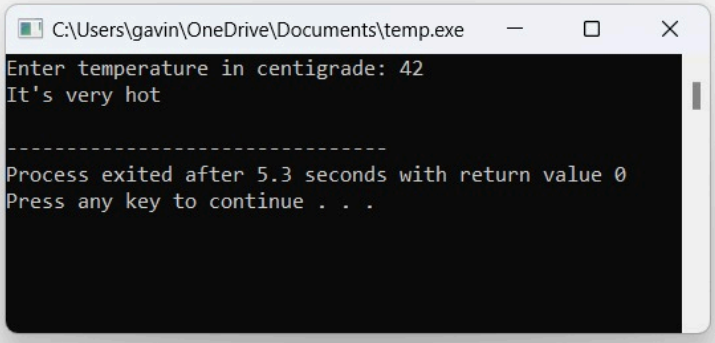
```c
#include <stdio.h>

int main() {
    float temp;

    printf("Enter temperature in centigrade: ");
    scanf("%f", &temp);

    if (temp < 0) {
        printf("Freezing weather\n");
    } else if (temp >= 0 && temp < 10) {
        printf("Very cold weather\n");
    } else if (temp >= 10 && temp < 20) {
        printf("Cold weather\n");
    } else if (temp >= 20 && temp < 30) {
        printf("Normal in Temp\n");
    } else if (temp >= 30 && temp < 40) {
        printf("It's hot\n");
    } else {
        printf("It's very hot\n");
    }

    return 0;
}
```



Compile Log | Debug | Find Results | Close

read and print.cpp | wages.cpp | roll number.cpp | temp.cpp | alphabet.cpp | [*] harmonic.cpp | [*] pascal.cpp

```c
#include <stdio.h>

int main() {
    float temp;

    printf("Enter temperature in centigrade: ");
    scanf("%f", &temp);

    if (temp < 0) {
        printf("Freezing weather\n");
    } else if (temp >= 0 && temp < 10) {
        printf("Very cold weather\n");
    } else if (temp >= 10 && temp < 20) {
        printf("Cold weather\n");
    } else if (temp >= 20 && temp < 30) {
        printf("Normal in Temp\n");
    } else if (temp >= 30 && temp < 40) {
        printf("It's hot\n");
    } else {
        printf("It's very hot\n");
    }

    return 0;
}
```



read and print.cpp | wages.cpp | roll number.cpp | temp.cpp | alphabet.cpp | [*] harmonic.cpp | [*] pascal.cpp

```c
#include <stdio.h>

int main() {
    char ch;

    printf("Enter a character: ");
    scanf("%c", &ch);

    if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {
        printf("%c is an alphabet.\n", ch);
    }
    else if (ch >= '0' && ch <= '9') {
        printf("%c is a digit.\n", ch);
    }
    else {
        printf("%c is a special character.\n", ch);
    }

    return 0;
}
```

C:\Users\gavin\OneDrive\Documents\alphabet.exe

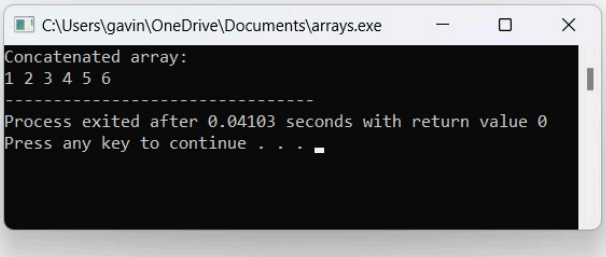
```
Enter a character: #
# is a special character.


--------------------------------
Process exited after 5.544 seconds with return value 0
Press any key to continue . . .
```

Compile Log | Debug | Find Results | Close

```
Compilation results...
--------
```

read and print.cpp | wages.cpp | roll number.cpp | temp.cpp | alphabet.cpp | [*] harmonic.cpp | pascal.cpp | arrays.cpp

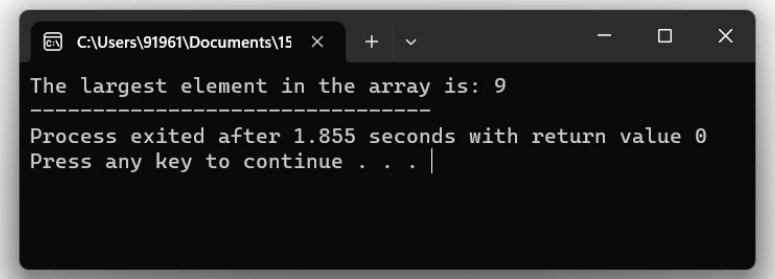
```c
#include <stdio.h>

int main() {
    int arr1[] = {1, 2, 3};
    int arr2[] = {4, 5, 6};
    int n1 = sizeof(arr1) / sizeof(int);
    int n2 = sizeof(arr2) / sizeof(int);
    int n = n1 + n2;
    int arr3[n];
    int i, j;

    for (i = 0; i < n1; i++) {
        arr3[i] = arr1[i];
    }

    for (j = 0; j < n2; j++) {
        arr3[i] = arr2[j];
        i++;
    }

    printf("Concatenated array:\n");

    for (i = 0; i < n; i++) {
        printf("%d ", arr3[i]);
    }

    return 0;
}
```



obals)

15.CLASSWORK 5.cpp

```c
#include <stdio.h>

int largestElement(int arr[], int size) {
    if (size == 1) {
        return arr[0];
    } else {
        int max = largestElement(arr, size - 1);
        if (arr[size - 1] > max) {
            return arr[size - 1];
        } else {
            return max;
        }
    }
}

int main() {
    int arr[] = {1, 3, 5, 7, 9, 2, 4, 6, 8};
    int size = sizeof(arr) / sizeof(arr[0]);

    int max = largestElement(arr, size);

    printf("The largest element in the array is: %d", max);

    return 0;
}
```



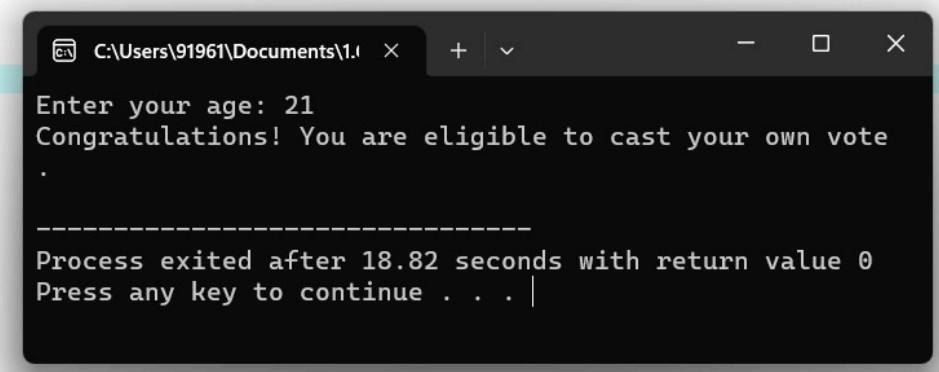
9.CLASSWORK5.cpp

```c
#include <stdio.h>

int main (void)
{
    int a[10][10];
    int i = 0, j = 0, row = 0, col = 0;

    printf ("Enter the order of the matrix (mxn):\n");
    printf ("where m = number of rows; and\n");
    printf ("      n = number of columns\n");
    scanf ("%d %d", &row, &col);

    int flag = 0;

    printf ("Enter the elements of the matrix\n");
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            scanf ("%d", &a[i][j]);
        }
    }

    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            if (i == j && a[i][j] != 1)
            {
                flag = -1;
                break;
            }
            else if (i != j && a[i][j] != 0)
            {
                flag = -1;
                break;
            }
        }
    }

    if (flag == 0)
    {
        printf ("It is a IDENTITY MATRIX\n");
    }
    else
    {
        printf ("It is NOT an identity matrix\n");
    }

    return 0;
}
```

```
C:\Users\91961\Documents\9.
Enter the order of the matrix (mxn):
where m = number of rows; and
      n = number of columns
3
3
Enter the elements of the matrix
1
0
0
0
1
0
0
0
1
It is a IDENTITY MATRIX

--------------------------------
Process exited after 112 seconds with return value 0
Press any key to continue . . .
```

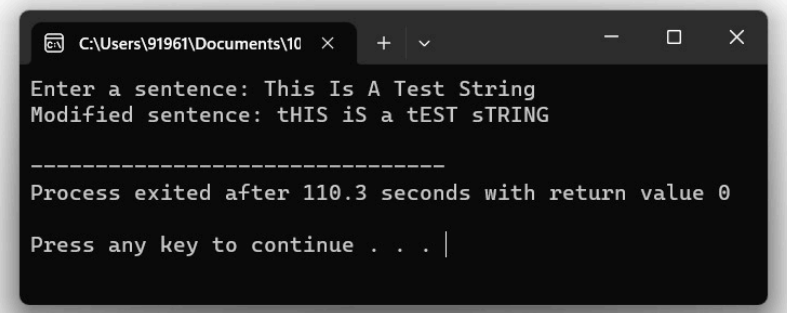
1.CLASSWORK 5.cpp

```c
#include <stdio.h>

int main() {
    int age;

    printf("Enter your age: ");
    scanf("%d", &age);

    if (age >= 18) {
        printf("Congratulations! You are eligible to cast your own vote.\n");
    } else {
        printf("Sorry, you are not eligible to cast your own vote.\n");
    }

    return 0;
}
```



10.CLASSWORK 5.cpp

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char sentence[1000];
    int i;

    printf("Enter a sentence: ");
    fgets(sentence, 1000, stdin);

    for (i = 0; sentence[i] != '\0'; i++) {
        if (islower(sentence[i]))
            sentence[i] = toupper(sentence[i]);
        else if (isupper(sentence[i]))
            sentence[i] = tolower(sentence[i]);
    }

    printf("Modified sentence: %s", sentence);

    return 0;
}
```



11.classwork 5.cpp

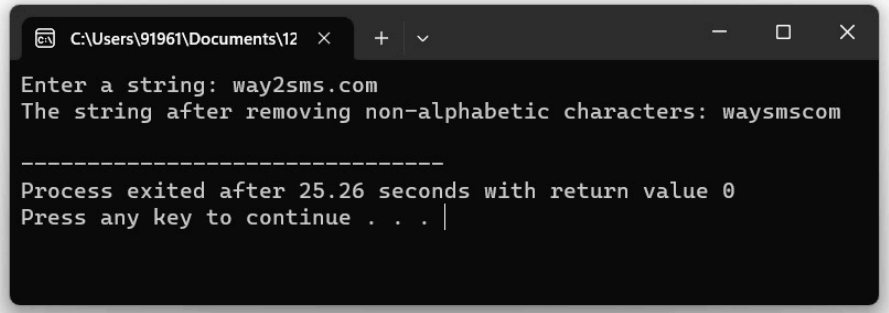
```c
#include <stdio.h>
#include <string.h>

int main() {
    char str[100];
    int count[256] = {0}; // Initialize the count array to zero

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    // Count the occurrences of each character in the string
    for (int i = 0; i < strlen(str); i++) {
        count[(int)str[i]]++;
    }

    // Print the count of each character
    printf("Character count:\n");
    for (int i = 0; i < 256; i++) {
        if (count[i] > 0) {
            printf("%c: %d\n", i, count[i]);
        }
    }

    return 0;
}
```

```
C:\Users\91961\Documents\11
Enter a string: way2sms.com
Character count:

: 1
.: 1
2: 1
a: 1
c: 1
m: 2
o: 1
s: 2
w: 1
y: 1

--------------------------------
Process exited after 31.77 seconds with return value 0
Press any key to continue . . .
```

ources | Compile Log | Debug | Find Results | Console | Close

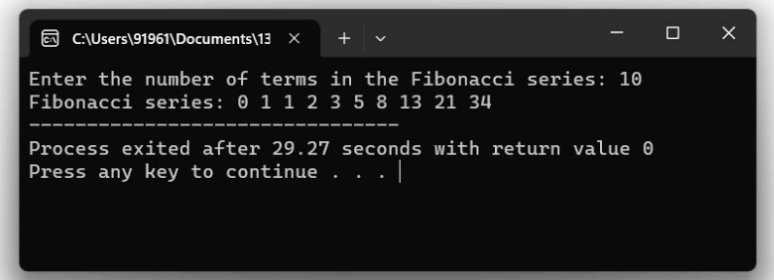
12.classwork 5.cpp

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main()
{
    char str[100];
    int i, j;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    for(i = 0, j = 0; i < strlen(str); i++)
    {
        if(isalpha(str[i]))
        {
            str[j] = str[i];
            j++;
        }
    }
    str[j] = '\0';

    printf("The string after removing non-alphabetic characters: %s", str);

    return 0;
}
```



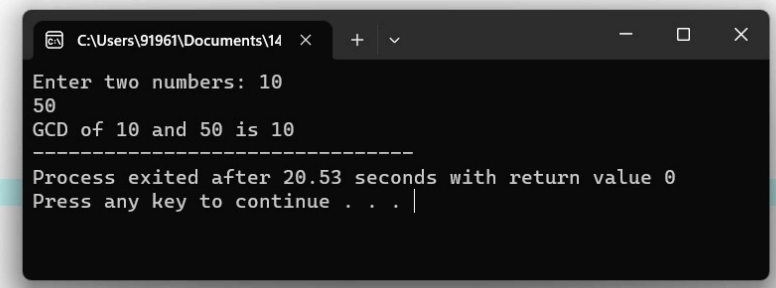
13.CLASSWORK5.cpp

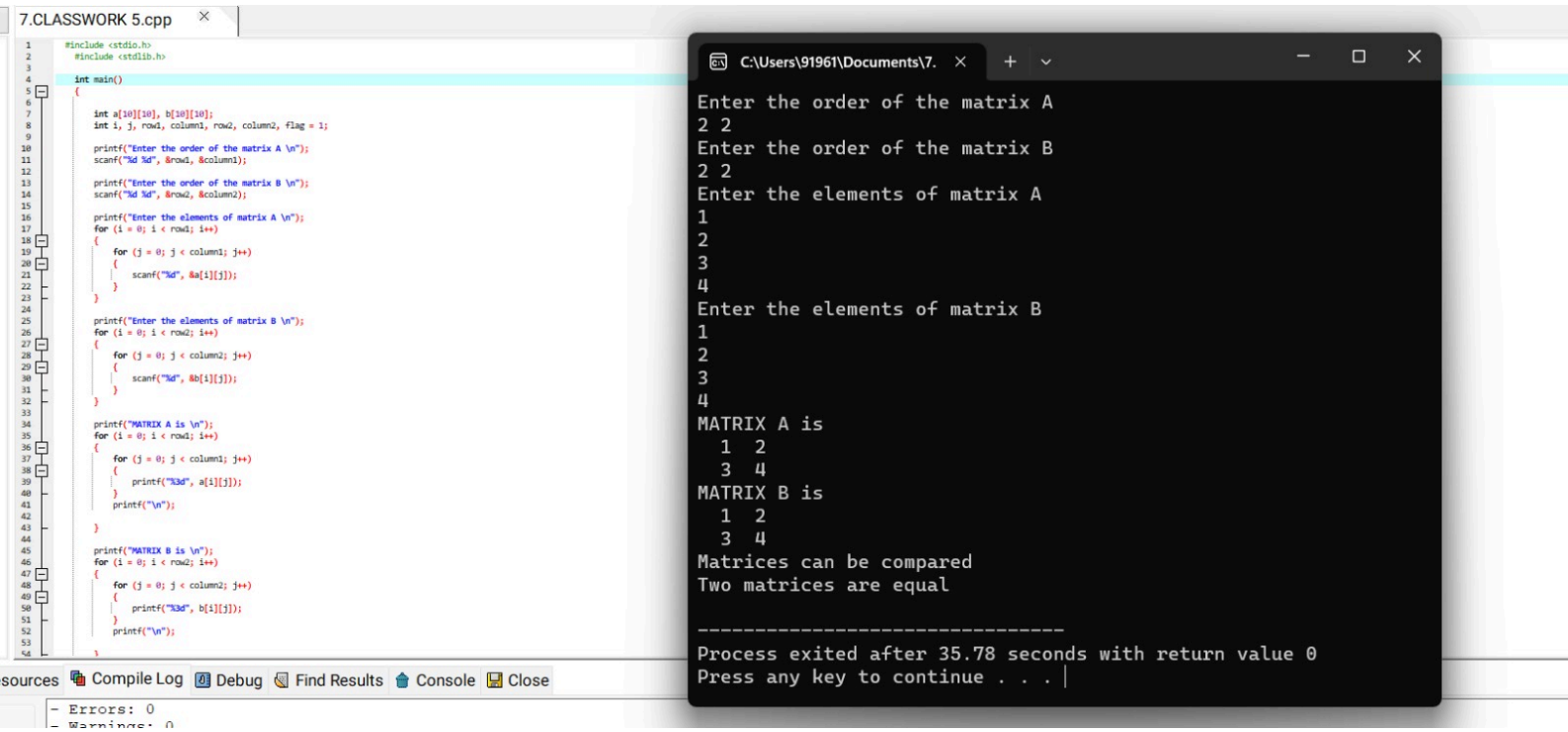
```c
#include <stdio.h>

// Function to calculate the nth Fibonacci number using recursion
int fibonacci(int n)
{
    // Base case
    if (n <= 1)
        return n;

    // Recursive case
    return fibonacci(n-1) + fibonacci(n-2);
}

int main()
{
    int n;

    // Get input from user
    printf("Enter the number of terms in the Fibonacci series: ");
    scanf("%d", &n);

    // Print the Fibonacci series
    printf("Fibonacci series: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", fibonacci(i));
    }

    return 0;
}
```

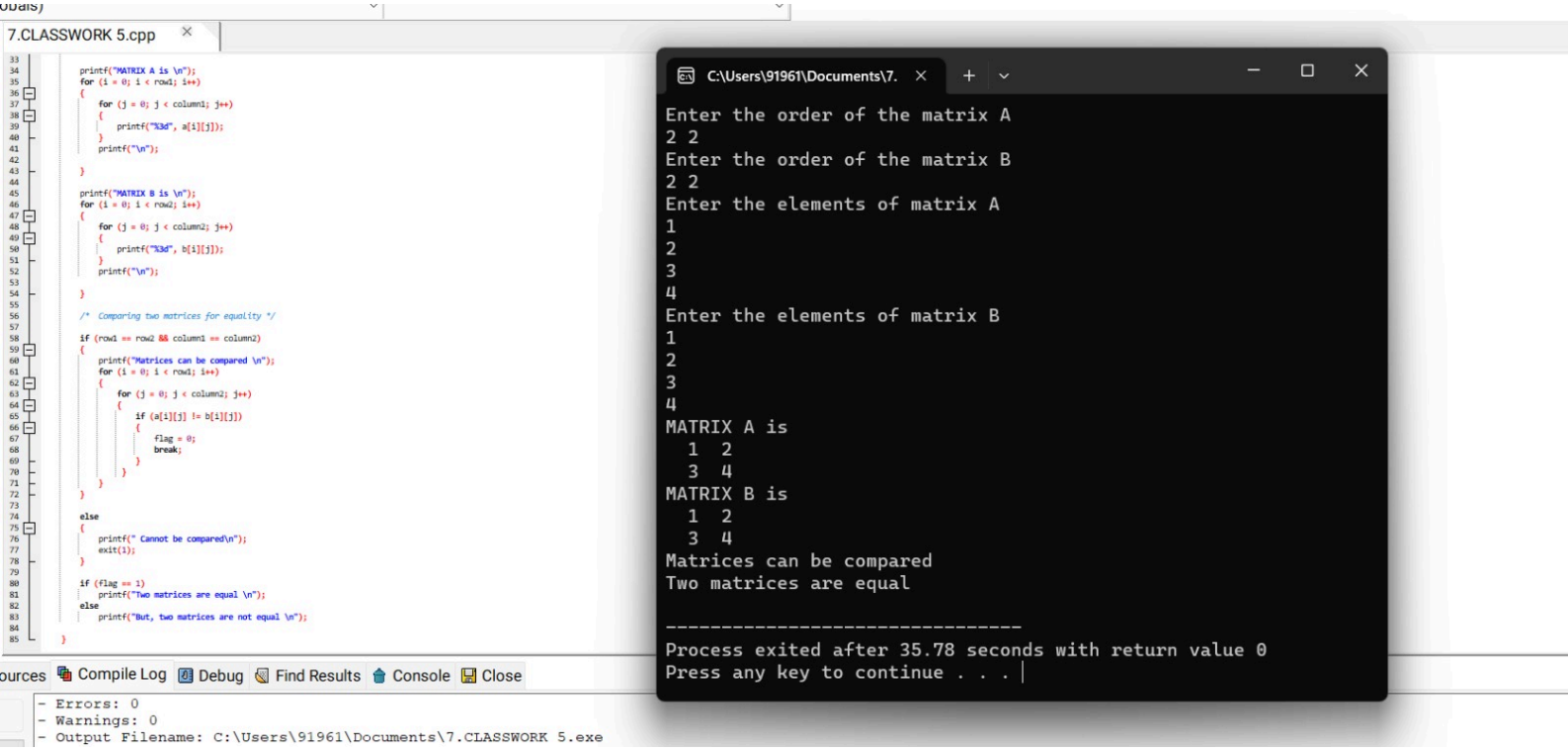


14.CLASSWORK 5.cpp

```c
#include <stdio.h>

int gcd(int a, int b);

int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    printf("GCD of %d and %d is %d", num1, num2, gcd(num1, num2));
    return 0;
}

int gcd(int a, int b) {
    if (b == 0) {
        return a;
    }
    return gcd(b, a % b);
}
```




7.CLASSWORK 5.cpp
#include <stdio.h>
#include <stdlib.h>
int main()
{
int a[10][10], b[10][10];
int i, j, row1, column1, row2, column2, flag = 1;
printf("Enter the order of the matrix A \n");
scanf("%d %d", &row1, &column1);
printf("Enter the order of the matrix B \n");
scanf("%d %d", &row2, &column2);
printf("Enter the elements of matrix A \n");
for (i = 0; i < row1; i++)
{
for (j = 0; j < column1; j++)
{
scanf("%d", &a[i][j]);
}
}
printf("Enter the elements of matrix B \n");
for (i = 0; i < row2; i++)
{
for (j = 0; j < column2; j++)
{
scanf("%d", &b[i][j]);
}
}
printf("MATRIX A is \n");
for (i = 0; i < row1; i++)
{
for (j = 0; j < column1; j++)
{
printf("%3d", a[i][j]);
}
printf("\n");
}
printf("MATRIX B is \n");
for (i = 0; i < row2; i++)
{
for (j = 0; j < column2; j++)
{
printf("%3d", b[i][j]);
}
printf("\n");
}
C:\Users\91961\Documents\7.
Enter the order of the matrix A
2 2
Enter the order of the matrix B
2 2
Enter the elements of matrix A
1
2
3
4
Enter the elements of matrix B
1
2
3
4
MATRIX A is
1 2
3 4
MATRIX B is
1 2
3 4
Matrices can be compared
Two matrices are equal
Process exited after 35.78 seconds with return value 0
Press any key to continue . . .
sources
Compile Log
Debug
Find Results
Console
Close
- Errors: 0


7.CLASSWORK 5.cpp
printf("MATRIX A is \n");
for (i = 0; i < row1; i++)
{
for (j = 0; j < column1; j++)
{
printf("%3d", a[i][j]);
}
printf("\n");
}
printf("MATRIX B is \n");
for (i = 0; i < row2; i++)
{
for (j = 0; j < column2; j++)
{
printf("%3d", b[i][j]);
}
printf("\n");
}
/* Comparing two matrices for equality */
if (row1 == row2 && column1 == column2)
{
printf("Matrices can be compared \n");
for (i = 0; i < row1; i++)
{
for (j = 0; j < column2; j++)
{
if (a[i][j] != b[i][j])
{
flag = 0;
break;
}
}
}
}
else
{
printf(" Cannot be compared\n");
exit(1);
}
if (flag == 1)
printf("Two matrices are equal \n");
else
printf("But, two matrices are not equal \n");
}
C:\Users\91961\Documents\7.
Enter the order of the matrix A
2 2
Enter the order of the matrix B
2 2
Enter the elements of matrix A
1
2
3
4
Enter the elements of matrix B
1
2
3
4
MATRIX A is
1 2
3 4
MATRIX B is
1 2
3 4
Matrices can be compared
Two matrices are equal

Process exited after 35.78 seconds with return value 0
Press any key to continue . . .
Compile Log
Debug
Find Results
Console
Close
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\91961\Documents\7.CLASSWORK 5.exe

16.CLASSWORK 5.cpp

```c
#include <stdio.h>

int isPrime(int n, int i);

int main() {
    int num;

    printf("Enter a positive integer: ");
    scanf("%d", &num);

    if (isPrime(num, num/2) == 1) {
        printf("%d is a prime number.\n", num);
    } else {
        printf("%d is not a prime number.\n", num);
    }

    return 0;
}

int isPrime(int n, int i) {
    if (i == 1) {
        return 1;
    } else {
        if (n % i == 0) {
            return 0;
        } else {
            return isPrime(n, i-1);
        }
    }
}
```