Module Title

Principles of Programming

Weekly Assignment - Practical 4

Year

2025

Student Name: NIRVIK K.C.

UWE ID: 25024649

Assignment Submission Date: January 9, 2026

This assignment consists of the programming questions from practical exercises related to the topics from week 4.

Questions

1. **One-dimensional arrays:**
   The problem: identifying a repeated offender (using 10 gene chromosomes) In a criminal investigation, DNA profiling was used to match the DNA of a suspect who has been arrested for suspicion of committing a burglary against the DNA profiling of several criminals who have been convicted of committing similar crimes. To identify a repeated offender correctly 10 particular gene chromosomes of the suspect's DNA profile were chosen which must exactly match for the corresponding ones of a convicted criminal.

   For example, the suspect's DNA profile consists of 10 real numbers as follows:
   2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
   and the DNA profile of a criminal is as follows:
   2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

Given code:

Add the following code to the program:

```
#include <stdio.h>
#include <stdbool.h>
int main()
{
int size = 10;
float suspect[size]; // declaring suspect array
float criminal[size]; // declaring criminal array
// read 10 input values into suspect array from keyboard
printf("Enter the 10 chromosomes of the suspect separated by spaces: \n");
for (int i = 0; i < size; i++)
scanf(" %f", &suspect[i]);
// read 10 input values into criminal array from keyboard
printf("Enter the 10 chromosomes of the criminal separated by spaces: \n");
for (int i = 0; i < size; i++)
scanf(" %f", &criminal[i]);
return 0;
}
```

Extend the program to match the two profiles in the two arrays. Add the following code to the program:

```
// match two profiles
bool match = true;
```

```
for (int i = 0; i < size; i++)
if (suspect[i] != criminal[i])
  match = false; // display matching result
if (match)
  printf("The two profiles match! \n");
else
   printf("The two profiles don't match! \n");
```

Now run and test your program with the following suspect profile:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

and the following criminal profile:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

Run and test your program again with the same suspect profile and another criminal profile:

1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9

Answer:

Step 1: Create a folder(practical4) to store the program (MatchingProfilesA.c) in this practical.

Step 2: Add the given code to read the two profiles from the keyboard.

Step 3: Extend the program to match the two profiles in the two arrays by adding the remaining given code to the program.

Step 4: Run and test your program with the following suspect profile:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

and the following criminal profile:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

Run and test your program again with the same suspect profile and another criminal profile:

1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9

Following code for input:

```
/* read values from input file
Practical 4 -  One-dimensional arrays, Part 1
@Nirvik K.C. */
#include <stdio.h>
#include <stdbool.h>
int main()
{
   int size = 10;
```

```
float suspect[size];  // declaring suspect array
float criminal[size]; // declaring criminal array
// read 10 input values into suspect array from keyboard
printf("Enter the 10 chromosomes of the suspect separated by spaces: \n");
for (int i = 0; i < size; i++)
{
   scanf(" %f", &suspect[i]);
}
// read 10 input values into criminal array from keyboard
printf("Enter the 10 chromosomes of the criminal separated by spaces: \n");
for (int i = 0; i < size; i++) {
   scanf(" %f", &criminal[i]);
}
// match two profiles
bool match = true;
for (int i = 0; i < size; i++)
{
   if (suspect[i] != criminal[i])
   {
      match = false;
   }
}
```

```
    // display matching result
    if (match)
    {
        printf("The two profiles match! \n");
    }
    else
    {
        printf("The two profiles don't match! \n");
    }
    return 0;
}

/* Output:
Test Case 1:
Enter the 10 chromosomes of the suspect separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the criminal separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
```

Test Case 2:

Enter the 10 chromosomes of the suspect separated by spaces:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

Enter the 10 chromosomes of the criminal separated by spaces:

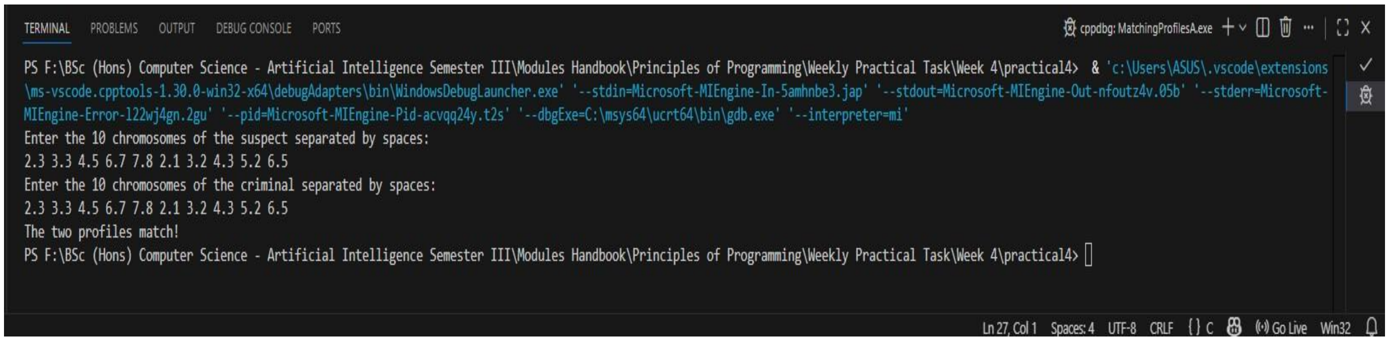
1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9

The two profiles don't match!

*/

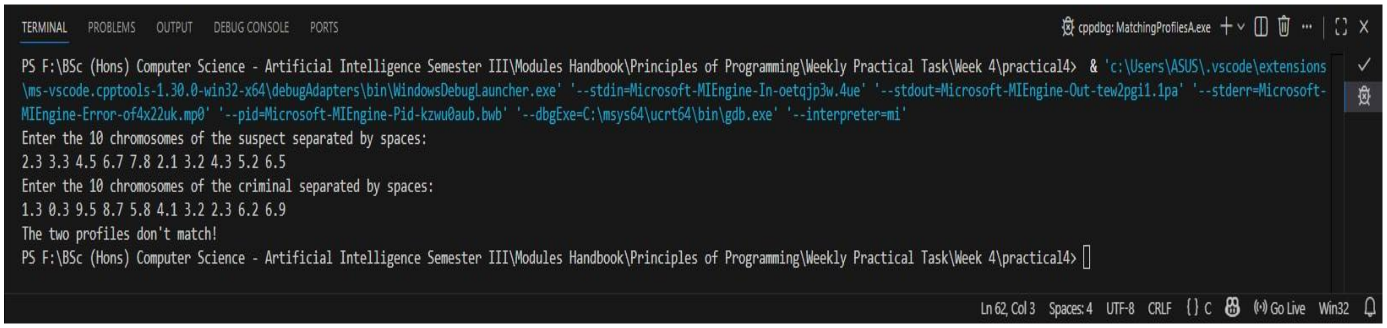
Output obtained in execution:

Test Case 1:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-5amhnbe3.jap' '--stdout=Microsoft-MIEngine-Out-nfoutz4v.05b' '--stderr=Microsoft-MIEngine-Error-122wj4gn.2gu' '--pid=Microsoft-MIEngine-Pid-acvqq24y.t2s' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter the 10 chromosomes of the suspect separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the criminal separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
The two profiles match!
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>
```

Test Case 2:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-oetqjp3w.4ue' '--stdout=Microsoft-MIEngine-Out-tew2pgi1.1pa' '--stderr=Microsoft-MIEngine-Error-of4x22uk.mp0' '--pid=Microsoft-MIEngine-Pid-kzwu0aub.bwb' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter the 10 chromosomes of the suspect separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the criminal separated by spaces:
1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9
The two profiles don't match!
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>
```

2. **Two-dimensional arrays**

Use a two-dimensional array to represent more convicted criminals. We have now been given more than one convicted criminal so we need a two-dimensional array to represent them using multiple rows in the array. Replace the following code for the one-dimensional array in the program,

Given code:

float criminal[size]; // declaring criminal array

with the following code for the two-dimensional array:

int sizeR = 3;

int sizeC = 10;

float criminals[sizeR][sizeC]; // declaring criminals array

Replace the code for reading 10 input values into the one-dimensional criminal [ ] array with the code for reading 3 sets of 10 input values into the two-dimensional criminals[ ] [ ] array:

// read multiple profiles of 10 values into criminals array from the keyboard

for (int i = 0; i < sizeR; i++)

{

printf("Enter the 10 chromosomes of the %dth criminal: \n", i+1);

// read 10 input values of a criminal into criminals array from the keyboard

```
for (int j = 0; j < sizeC; j++)
{
   scanf(" %f", &criminals[i][j]);
}
}
```

Change the code for matching two profiles into the code for matching the suspect with each of the 3 criminals.

Change the code for displaying the matching result of the suspect and a criminal into the code for displaying the matching result of the suspect with each of the criminals.

Run and test your program

Now run and test your program with the following suspect profile:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

and the following three criminal profiles:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9

6.3 9.3 4.3 6.4 7.5 2.9 3.0 4.1 5.3 6.5

Answer:

Following code for input:

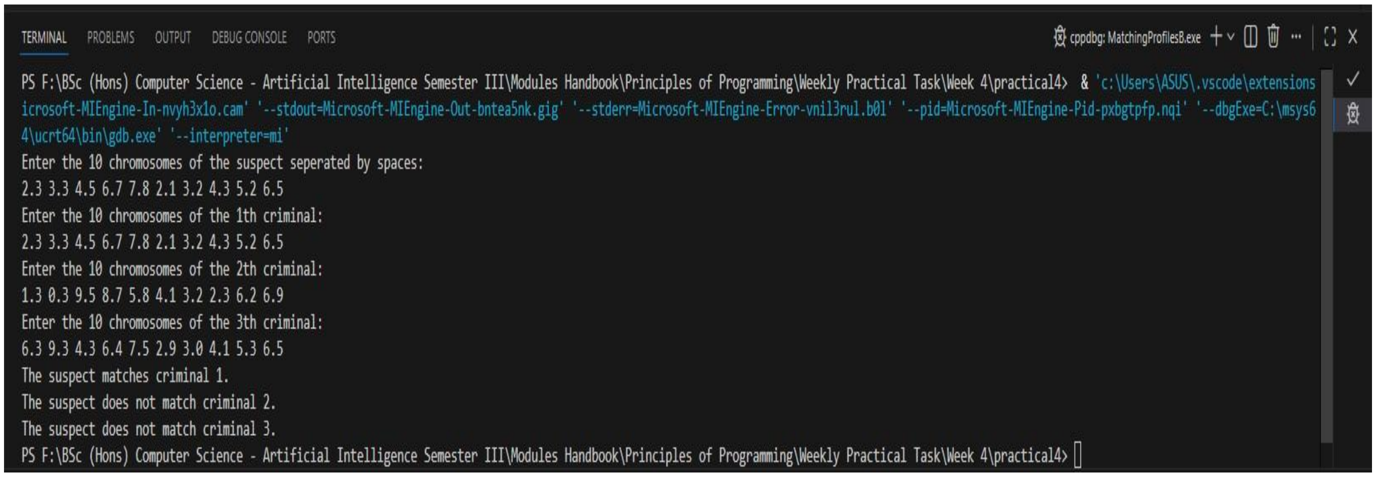
```
/* read values from input file
Practical 4 -  Two-dimensional arrays, Part 2
@Nirvik K.C. */


#include <stdio.h>
#include <stdbool.h>
int main()
{
   int size = 10;  // number of chromosomes
   int sizeR = 3;  // number of criminals
   int sizeC = 10; // number of chromosomes per criminal
   float suspect[size]; // declaring suspect array
   float criminals[sizeR][sizeC]; // declaring criminals two-dimensional array
   // read 10 input values into suspect array from keyboard
   printf("Enter the 10 chromosomes of the suspect separated by spaces: \n");
   for (int i = 0; i < size; i++)
   {
      scanf("%f", &suspect[i]);
   }
```

```
// read 3 criminals' profiles
 for (int i = 0; i < sizeR; i++)
 {
    printf("Enter the 10 chromosomes of the %dth criminal:\n", i + 1);
    for (int j = 0; j < sizeC; j++)
    {
       scanf("%f", &criminals[i][j]);
    }
 }

 // match suspect profile with each criminal profile
 for (int i = 0; i < sizeR; i++)
 {
    bool match = true;

    for (int j = 0; j < sizeC; j++)
    {
       if (suspect[j] != criminals[i][j])
       {
          match = false;
       }
    }
```

```
        // display matching result
        if (match)
        {
            printf("The suspect matches criminal %d - repeated offender!\n", i + 1);
        }
        else
        {
            printf("The suspect does not match criminal %d.\n", i + 1);
        }
    }
    return 0;
}
```

```
/*Output:
Enter the 10 chromosomes of the suspect separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the 1th criminal:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the 2th criminal:
1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9
Enter the 10 chromosomes of the 3th criminal:
6.3 9.3 4.3 6.4 7.5 2.9 3.0 4.1 5.3 6.5
The suspect matches criminal 1.
The suspect does not match criminal 2.
The suspect does not match criminal 3.
*/
```

Output obtained in execution:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions
icrosoft-MIEngine-In-nvyh3x1o.cam' '--stdout=Microsoft-MIEngine-Out-bntea5nk.gig' '--stderr=Microsoft-MIEngine-Error-vnil3rul.b0l' '--pid=Microsoft-MIEngine-Pid-pxbgtpfp.nqi' '--dbgExe=C:\msys6
4\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter the 10 chromosomes of the suspect seperated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the 1th criminal:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the 2th criminal:
1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9
Enter the 10 chromosomes of the 3th criminal:
6.3 9.3 4.3 6.4 7.5 2.9 3.0 4.1 5.3 6.5
The suspect matches criminal 1.
The suspect does not match criminal 2.
The suspect does not match criminal 3.
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4> 
```

3. **Part 3. Arrays**

In this exercise, you are required: first, to design, write, run and test a C program called repeatedOffenderA.c in the practical4 folder that you have already created, which will use two arrays, process them, and display the results of the processing on screen. The problem: identifying a repeated offender (using 10 gene chromosomes) In a criminal investigation, DNA profiling was used to match the DNA of a suspect who has been arrested for suspicion of committing a burglary against the DNA profiling of several criminals who have been convicted of committing similar crimes and therefore identified as a repeated offender. To identify a repeated offender correctly 10 particular gene chromosomes of the suspect's DNA profile were chosen which must exactly match for the corresponding ones of a convicted criminal.

The suspect's DNA profile consists of 10 real numbers as follows:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

There are 5 convicted criminals whose DNA profiles are used to match with the DNA profile of the suspect, each simply being identified as criminal 0, 1, 2, 3 or 4. The DNA profiles of the 5 convicted criminals, each of which consists of 10 real numbers, are as follows:

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9

6.3 9.3 4.3 6.4 7.5 2.9 3.0 4.1 5.3 6.5

6.1 9.4 4.5 6.6 7.4 2.8 3.2 4.4 5.0 6.0

2.3 3.3 4.5 6.6 7.8 2.2 3.2 4.3 5.2 6.5

An input file typically contains the data for the DNA of the suspect followed on the next line by an integer specifying the number of the DNA profiles for the criminals, and then followed by the profiles of the 5 criminals. For example, the file contains the following data:

```
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
5
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9
6.3 9.3 4.3 6.4 7.5 2.9 3.0 4.1 5.3 6.5
6.1 9.4 4.5 6.6 7.4 2.8 3.2 4.4 5.0 6.0
2.3 3.3 4.5 6.6 7.8 2.2 3.2 4.3 5.2 6.5
```

Assume the data has been validated so your program would not need to validate it. Each of these sets of 10 values contains the data for DNA of a convicted criminal.

You are required to write a program, which will take input from the input file, match the DNA profile of the suspect with each of the convicted criminals, and display a report listing the reference number of the DNA profile which matches the DNA profile of the suspect who therefore can be identified as a repeated offender.

Your program is required to declare and initialise a one-dimensional array, called suspect, to hold the DNA profile of the suspect from the input file and a two-dimensional array, called criminals, to hold the DNA profiles of the 5 convicted criminals from the input file.

Answer:

Following program outline:

1. Declare variables and arrays (10 chromosomes, 5 criminals).
2. Open the input file "dna_input.txt" for reading the input data.
3. Read the 10 chromosome values for the suspect
4. Read the integer number of criminals (5).
5. Read the DNA profiles of criminals.
6. Close the file.
7. Compare the suspect's profile with each criminal's profile.
8. Display the result with list for each criminal whether match or no match. Also declare the conclusion with the suspect is a repeated offender.

Following pseudocode:

```
Start program

Declare file pointer

Declare array suspect with size 10

Declare two-dimensional array criminals with size 5 by 10

Declare the file dna_input.txt in read mode

If file cannot be opened

  Display error message

  End program
```

```
Read 10 DNA values from the file into suspect array
Read number of criminals from the file
For each criminal from 0 to number of criminals minus 1
  For each criminal from 0 to 9
    Read DNA value into criminals array
Close the file
Display DNA matching report heading
Set found match flag to false
For each criminal from 0 to number of criminals minus 1
  Set match flag to true
  For each chromosome from 0 to 9
     If suspect DNA value is not equal to criminal DNA value
        Set match flag to false
        Exit inner loop
  If match flag is true
      Display match found message with criminal number
      Set found match flag to true
    Else
      Display no match message for that criminal
If found match flag is true
   Display conclusion that suspect is a repeated offender
Else
   Display conclusion that suspect is not a repeated offender
End program
```

Following code for input:

/* read values from input file

Practical 4 - Arrays (exercise) , Part 3

@Nirvik K.C. */

```
#include <stdio.h>

#include <stdbool.h>

int main()

{

   FILE *fp;

   float suspect[10];

   float criminals[5][10];

   int numCriminals;

   int i, j;

   int match;

   int foundAnyMatch = 0;

    // Open the input file

   fp = fopen(“dna_input.txt”, “r”);

   if (fp == NULL)

   {

      printf(“Error opening the file!\n”);

      return 1;

   }
```

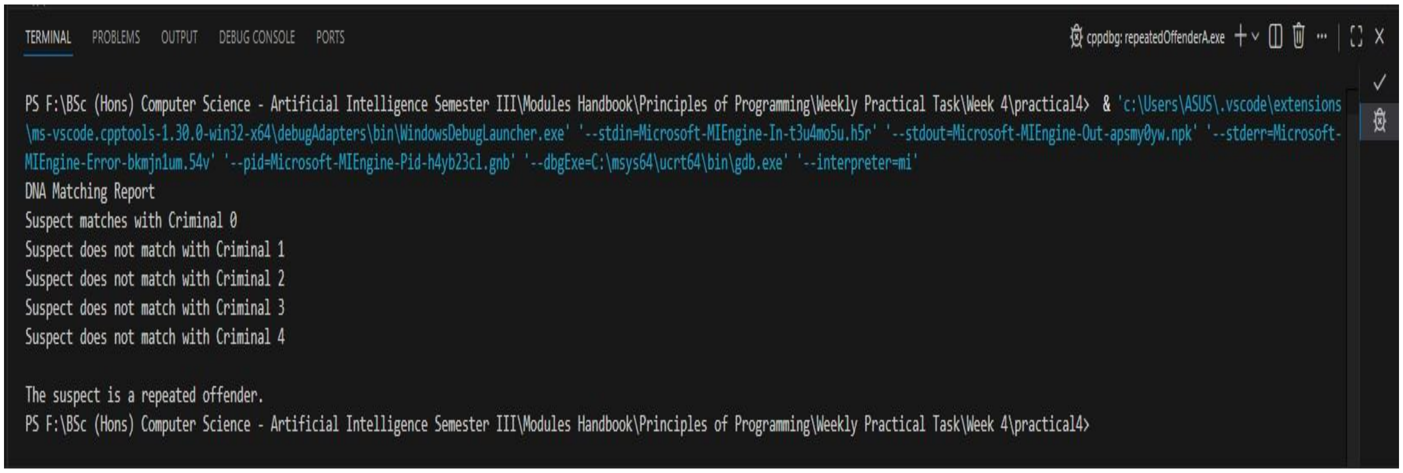
```
 // Read suspect profile
 for (i = 0; i < 10; i++)
{
   fscanf(fp, "%f", &suspect[i]);
}
// Read the number of criminals
fscanf(fp, "%d", &numCriminals);
// Read the profiles of the criminals
for (i = 0; i < numCriminals; i++)
{
   for (j = 0; j < 10; j++)
   {
      fscanf(fp, "%f", &criminals[i][j]);
   }
}
fclose(fp);
printf("DNA Matching Report\n");
foundAnyMatch = 0;
```

```
for (int i = 0; i < numCriminals; i++)
  {
      match = 1;

      for (j = 0; j < 10; j++)
      {
        if (suspect[j] != criminals[i][j])
        {
            match = 0;
            break;
        }
      }
       if (match)
       {
          printf("Suspect matches with Criminal %d\n", i);
          foundAnyMatch = 1;
       }
```

```
        else
        {
            printf("Suspect does not match with Criminal %d\n", i);
        }
    }
        printf("\n");
    if (foundAnyMatch)
    {
        printf("The suspect is a repeated offender.\n");
    }
    else
    {
        printf("The suspect is not a repeated offender.\n");
    }
    return 0;
}
```

```
/*Output:

DNA Matching Report

Suspect matches with Criminal 0

Suspect does not match with Criminal 1

Suspect does not match with Criminal 2

Suspect does not match with Criminal 3

Suspect does not match with Criminal 4


The suspect is a repeated offender.

*/
```

Output obtained in execution:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-t3u4mo5u.h5r' '--stdout=Microsoft-MIEngine-Out-apsmy0yw.npk' '--stderr=Microsoft-MIEngine-Error-bkmjn1um.54v' '--pid=Microsoft-MIEngine-Pid-h4yb23cl.gnb' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
DNA Matching Report
Suspect matches with Criminal 0
Suspect does not match with Criminal 1
Suspect does not match with Criminal 2
Suspect does not match with Criminal 3
Suspect does not match with Criminal 4

The suspect is a repeated offender.
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>
```

4. (i) **(Print distinct numbers)**

Write a program that reads in ten numbers and displays the number of distinct numbers and the distinct numbers separated by exactly one space (i.e., if a number appears multiple times, it is displayed only once). (Hint: Read a number and store it to an array if it is new. If the number is already in the array, ignore it.) After the input, the array contains the distinct numbers. Here is the sample run of the program:

```
Enter ten numbers: 1 2 3 2 1 6 3 4 5 2
The number of distinct numbers is 6
The distinct numbers are: 1 2 3 6 4 5
```

Answer:

Following code for input:

```
/* read values from input file
Practical 4 - More Programming exercises, Part 4 - 4.1 (Print distinct numbers)
@Nirvik K.C. */
include <stdio.h>
int main()
{
```

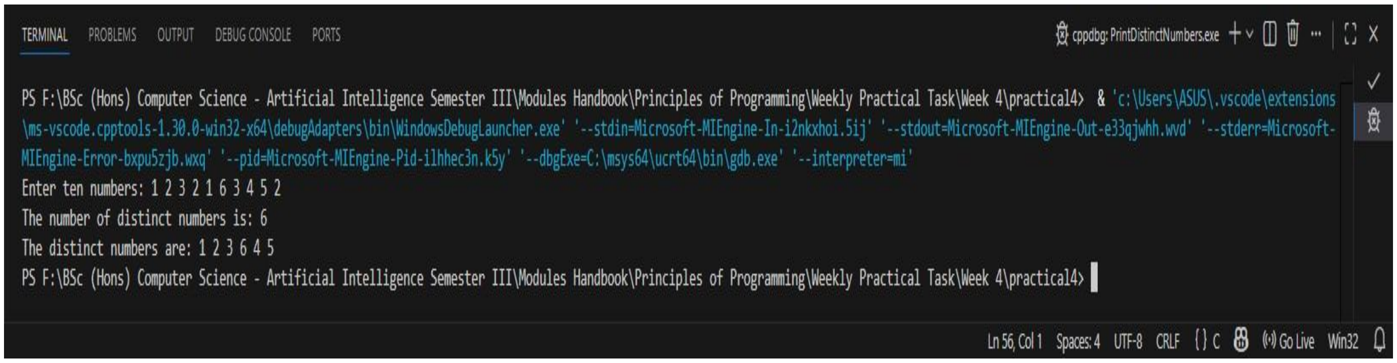
```
int numbers[10]; // declaring an array to hold 10 numbers
int count = 0;
int num;
int i, j;
printf("Enter ten numbers: ");
for (i = 0; i < 10; i++)
{
   scanf("%d", &num);

   // Check if already exists
   for (j = 0; j < count; j++)
   {
      if (numbers[j] == num)
      {
         break;
      }
   }
```

```
    // If the number wasn't found in the list, add the new number
    if (j == count)
    {
        numbers[count] = num;
        count++;
    }
}
printf("The number of distinct numbers is: %d\n", count);
printf("The distinct numbers are: ");
for (i = 0; i < count; i++)
{
    printf("%d", numbers[i]);


    if (i < count - 1)
    {
        printf(" ");
    }
}
```

```
    printf("\n");

    return 0;

}


/*Output:

Enter ten numbers: 1 2 3 2 1 6 3 4 5 2

The number of distinct numbers is: 6

The distinct numbers are: 1 2 3 6 4 5


Enter ten numbers: 4 1 4 9 1 4 2 9 4 1

The number of distinct numbers is: 4

The distinct numbers are: 4 1 9 2

*/
```

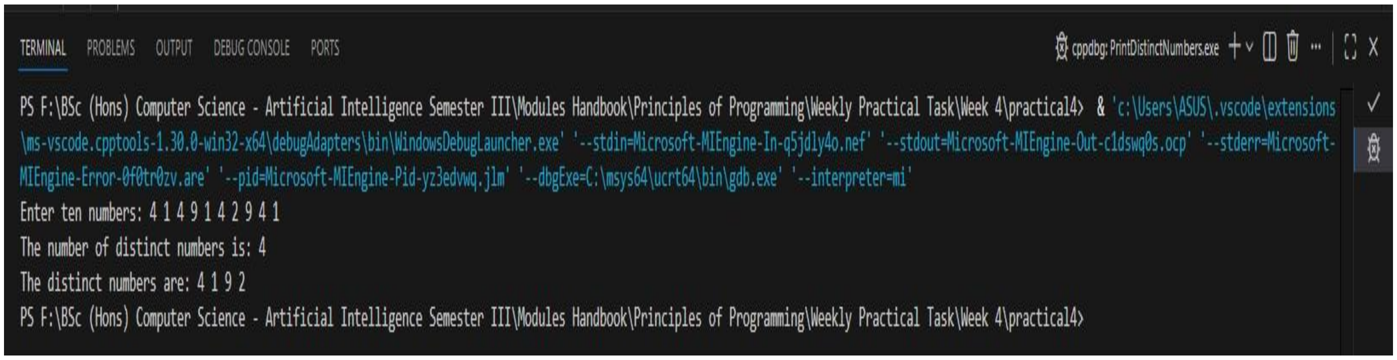
Output obtained in execution:

Example 1:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions
\ms-vscode.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-i2nkxhoi.5ij' '--stdout=Microsoft-MIEngine-Out-e33qjwhh.wvd' '--stderr=Microsoft-
MIEngine-Error-bxpu5zjb.wxq' '--pid=Microsoft-MIEngine-Pid-ilhhec3n.k5y' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter ten numbers: 1 2 3 2 1 6 3 4 5 2
The number of distinct numbers is: 6
The distinct numbers are: 1 2 3 6 4 5
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>
```

Example 2:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions
\ms-vscode.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-q5jdly4o.nef' '--stdout=Microsoft-MIEngine-Out-c1dswq0s.ocp' '--stderr=Microsoft-
MIEngine-Error-0f0tr0zv.are' '--pid=Microsoft-MIEngine-Pid-yz3edvwq.jlm' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter ten numbers: 4 1 4 9 1 4 2 9 4 1
The number of distinct numbers is: 4
The distinct numbers are: 4 1 9 2
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>
```

4. (ii) **(Find the smallest element)**

Write a program that finds the smallest element in an array of double values. Test program that prompts the user to enter ten numbers, finds the minimum value, and displays the minimum value. Here is a sample run of the program:

```
Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2

The minimum number is: 1.5
```

Answer:

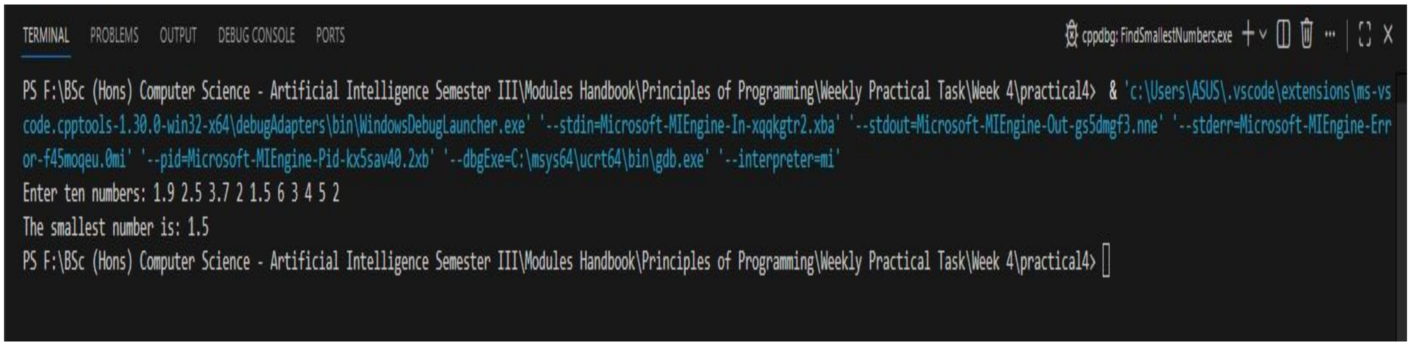
Following code for input:

```
/* read values from input file

Practical 4 - More Programming exercises, Part 4 - 4.2 (Find the smallest number)

@Nirvik K.C. */

#include <stdio.h>

int main()

{

    double numbers[10]; // declaring an array to store 10 double values

    double min;
```

```
int i;

 printf("Enter ten numbers: ");

 // Read 10 double numbers from the user

 for (int i = 0; i < 10; i++)

 {

     scanf("%lf", &numbers[i]);

 }

 // Find the smallest number

 min = numbers[0];

 for (int i = 0; i < 10; i++)

 {

     if (numbers[i] < min)

     {

         min = numbers[i];

     }
 }

   // Display the smallest number

   printf("The smallest number is: %.1f\n", min);

   return 0;

}


/*Output:

Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2

The smallest number is: 1.5



Enter ten numbers: 4.8 9.1 2.3 0.0 -5.6 7.2 1.1 3.4 6.9 8.0

The smallest number is: -5.6

*/
```

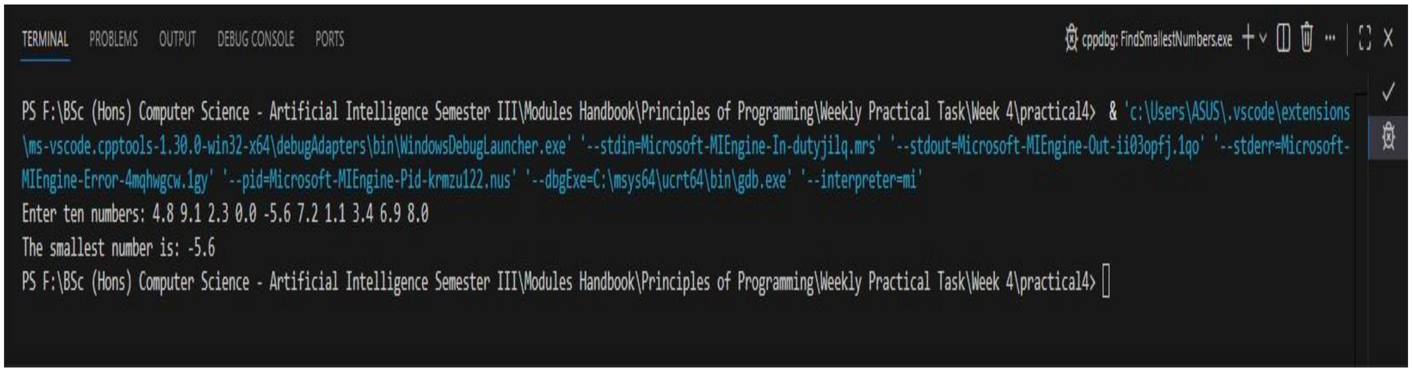
Output obtained in execution:

Example 1:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions\ms-vs
code.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-xqqkgtr2.xba' '--stdout=Microsoft-MIEngine-Out-gs5dmgf3.nne' '--stderr=Microsoft-MIEngine-Err
or-f45moqeu.0mi' '--pid=Microsoft-MIEngine-Pid-kx5sav40.2xb' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2
The smallest number is: 1.5
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4> 
```

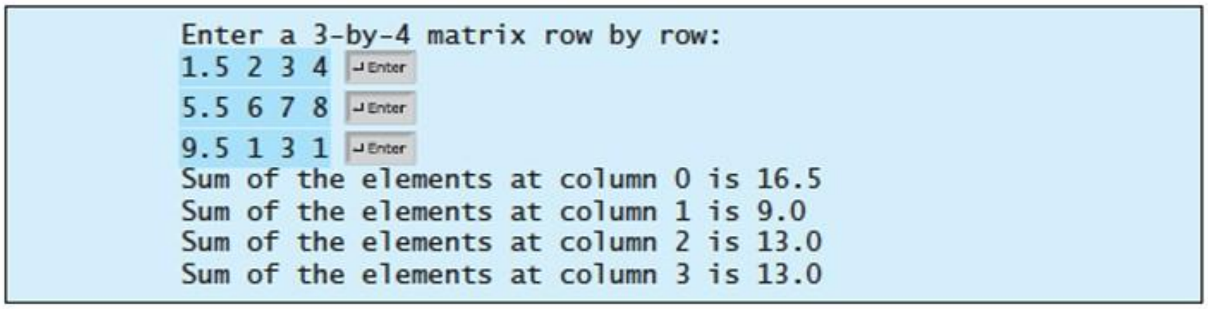
Example 2:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions
\ms-vscode.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-dutyjilq.mrs' '--stdout=Microsoft-MIEngine-Out-ii03opfj.1qo' '--stderr=Microsoft-
MIEngine-Error-4mqhwgcw.1gy' '--pid=Microsoft-MIEngine-Pid-krmzu122.nus' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter ten numbers: 4.8 9.1 2.3 0.0 -5.6 7.2 1.1 3.4 6.9 8.0
The smallest number is: -5.6
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4> 
```

4. (iii) **(Sum elements column by column)**

Write a program that returns the sum of all the elements in a specified column in a matrix.



```
Enter a 3-by-4 matrix row by row:
1.5 2 3 4 ↵Enter
5.5 6 7 8 ↵Enter
9.5 1 3 1 ↵Enter
Sum of the elements at column 0 is 16.5
Sum of the elements at column 1 is 9.0
Sum of the elements at column 2 is 13.0
Sum of the elements at column 3 is 13.0
```

Answer:

Following code for input:

```
/* read values from input file

Practical 4 - More Programming exercises, Part 4 - 4.3 (Sum elements column by column)

@Nirvik K.C. */

#include <stdio.h>

int main()

{

    double matrix[3][4]; // 3 rows and 4 columns
```

```
 int row, col;

 double sum; // sum of elements in each column

 printf("Enter a 3-by4 matrix row by row:\n");

 // Read the matrix elements from user input

 for (row = 0; row < 3; row++)

 {

     for (col = 0; col < 4; col++)

     {

         scanf("%lf", &matrix[row][col]);

     }

 }

// Calculate and display the sum of each column

for (col = 0; col < 4; col++)

 {

     sum = 0.0;

     // Sum of all elements in the current column
```

```
        for (row = 0; row < 3; row++)

        {

            sum += matrix[row][col];

        }

        printf("Sum of the elements at column %d is %.1f\n", col, sum);

    }

    return 0;

}

/*Output:

Enter a 3-by-4 matrix row by row:

1.5 2 3 4

5.5 6 7 8

9.5 1 3 1

Sum of the elements at column 0 is 16.5

Sum of the elements at column 1 is 9.0

Sum of the elements at column 2 is 13.0

Sum of the elements at column 3 is 13.0
```

Enter a 3-by4 matrix row by row:

1.2 0.5 4.0 2.1

3.0 1.5 2.0 0.0

5.5 2.5 1.0 3.6

Sum of the elements at column 0 is 9.7

Sum of the elements at column 1 is 4.5

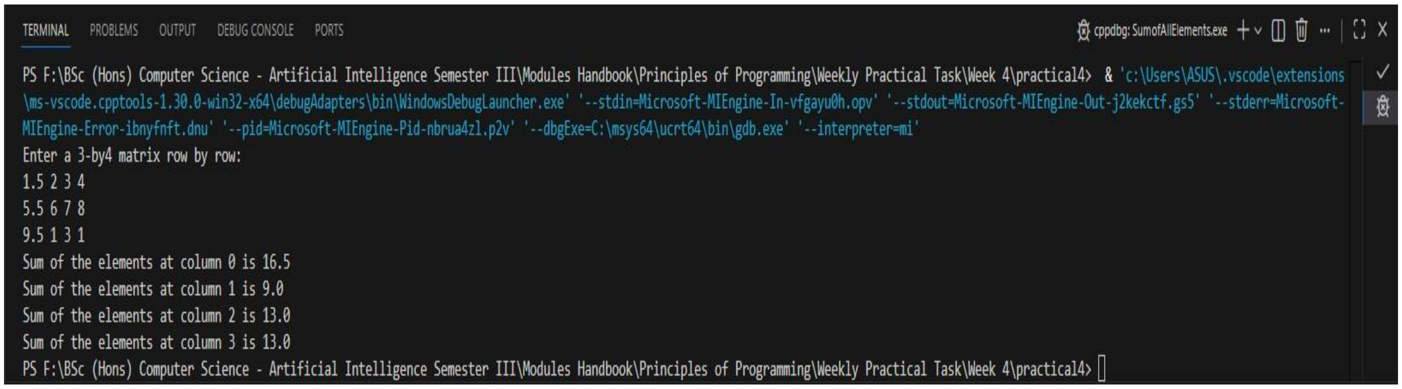
Sum of the elements at column 2 is 7.0

Sum of the elements at column 3 is 5.7

*/

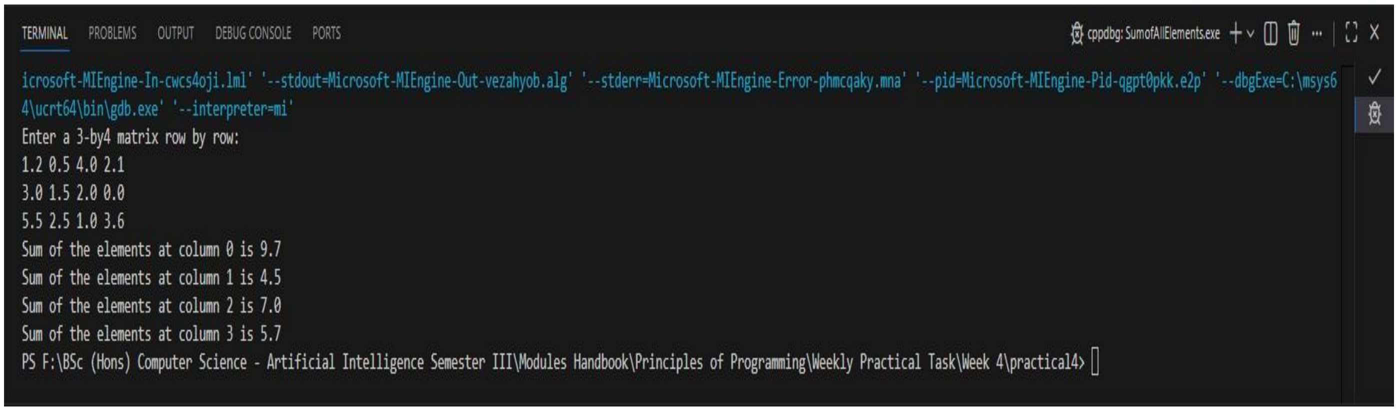
Output obtained in execution:

Example 1:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-vfgayu0h.opv' '--stdout=Microsoft-MIEngine-Out-j2kekctf.gs5' '--stderr=Microsoft-MIEngine-Error-ibnyfnft.dnu' '--pid=Microsoft-MIEngine-Pid-nbrua4zl.p2v' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter a 3-by4 matrix row by row:
1.5 2 3 4
5.5 6 7 8
9.5 1 3 1
Sum of the elements at column 0 is 16.5
Sum of the elements at column 1 is 9.0
Sum of the elements at column 2 is 13.0
Sum of the elements at column 3 is 13.0
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>
```

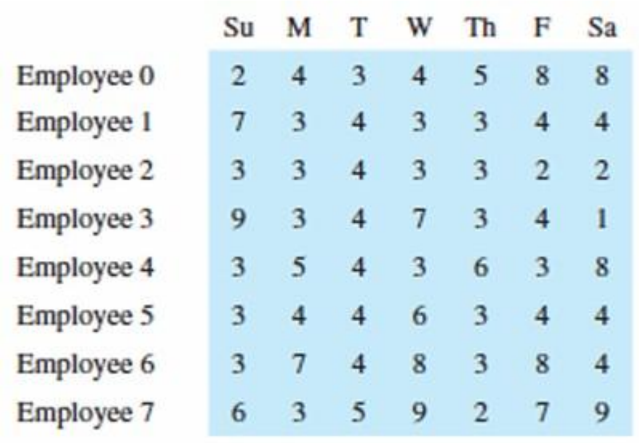
Example 2:



```
icrosoft-MIEngine-In-cwcs4oji.1ml' '--stdout=Microsoft-MIEngine-Out-vezahyob.alg' '--stderr=Microsoft-MIEngine-Error-phmcqaky.mna' '--pid=Microsoft-MIEngine-Pid-qgpt0pkk.e2p' '--dbgExe=C:\msys6
4\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter a 3-by4 matrix row by row:
1.2 0.5 4.0 2.1
3.0 1.5 2.0 0.0
5.5 2.5 1.0 3.6
Sum of the elements at column 0 is 9.7
Sum of the elements at column 1 is 4.5
Sum of the elements at column 2 is 7.0
Sum of the elements at column 3 is 5.7
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>
```

4. (iv) (Compute the weekly hours for each employee)

Suppose the weekly hours for all employees are stored in a two-dimensional array. Each row records an employee's seven day work hours with seven columns. For example, the following array stores the work hours for eight employees. Write a program that displays employees and their total hours in decreasing order of the total hours.



	Su	M	T	W	Th	F	Sa
Employee 0	2	4	3	4	5	8	8
Employee 1	7	3	4	3	3	4	4
Employee 2	3	3	4	3	3	2	2
Employee 3	9	3	4	7	3	4	1
Employee 4	3	5	4	3	6	3	8
Employee 5	3	4	4	6	3	4	4
Employee 6	3	7	4	8	3	8	4
Employee 7	6	3	5	9	2	7	9

Answer:

Following code for input:

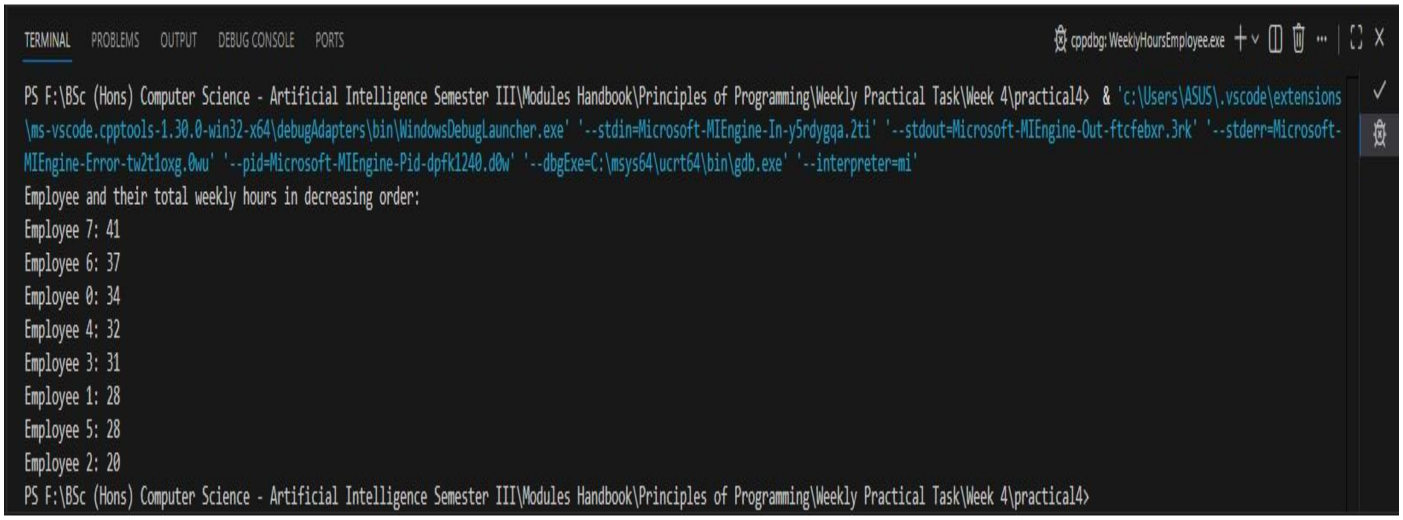
```
/* Practical 4 - More Programming exercises, Part 4 - 4.4 (Compute the weekly
hours for each employee)
@Nirvik K.C. */
#include <stdio.h>
int main()
{
   // 8 employees, 7 days
   int hours[8][7] = {
      {2, 4, 3, 4, 5, 8, 8},
      {7, 3, 4, 3, 3, 4, 4},
      {3, 3, 4, 3, 3, 2, 2},
      {9, 3, 4, 7, 3, 4, 1},
      {3, 5, 4, 3, 6, 3, 8},
      {3, 4, 4, 6, 3, 4, 4},
      {3, 7, 4, 8, 3, 8, 4},
      {6, 3, 5, 9, 2, 7, 9}};
```

```
int total[8];

int employeeOrder[8];

int i, j, temp;

int currentEmployee;

// Calculate total hours for each employee

for (i = 0; i < 8; i++)

{

   total[i] = 0;

   for (j = 0; j < 7; j++)

   {

      total[i] += hours[i][j];

   }

   employeeOrder[i] = i;

}

// Sort employees based on total hours

for (i = 0; i < 7; i++)

{

   for (j = 0; j < 7 - i; j++) {
```

```
            // Compare the totals using the current order
            if (total[employeeOrder[j]] < total[employeeOrder[j + 1]])
            {
                // Swap total hours
                temp = employeeOrder[j];
                employeeOrder[j] = employeeOrder[j + 1];
                employeeOrder[j + 1] = temp;
            }
        }
    }
    // Display result
    printf("Employee and their total weekly hours in decreasing order:\n");
   for (int k = 0; k < 8; k++) {
       int currentEmployee = employeeOrder[k];
       printf("Employee %d: %d\n", currentEmployee, total[currentEmployee]);
    }
    return 0;
}
```

```
/*Output:

Employee and their total hours in decreasing order:

Employee 7: 41

Employee 6: 37

Employee 0: 34

Employee 4: 32

Employee 3: 31

Employee 1: 28

Employee 5: 28

Employee 2: 20

*/
```

Output obtained in execution:



```
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>  & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.30.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-y5rdygqa.2ti' '--stdout=Microsoft-MIEngine-Out-ftcfebxr.3rk' '--stderr=Microsoft-MIEngine-Error-tw2t1oxg.0wu' '--pid=Microsoft-MIEngine-Pid-dpfk1240.d0w' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Employee and their total weekly hours in decreasing order:
Employee 7: 41
Employee 6: 37
Employee 0: 34
Employee 4: 32
Employee 3: 31
Employee 1: 28
Employee 5: 28
Employee 2: 20
PS F:\BSc (Hons) Computer Science - Artificial Intelligence Semester III\Modules Handbook\Principles of Programming\Weekly Practical Task\Week 4\practical4>
```