1. Write a C code to check if a 3 x 3 matrix is invertible. A matrix is not invertible if its determinant is 0.

Ans:

#include<stdio.h>

int main()

{

int a[3][3], i, j;

long determinant;

// 9 elements of matrix is taken as input from test data

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

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

scanf("%d", &a[i][j]);

/\*Use the printf statements as:

printf("The given matrix is not invertible");

printf("The given matrix is invertible");

\*/

determinant = a[0][0] \* ((a[1][1]\*a[2][2]) - (a[2][1]\*a[1][2])) -a[0][1]\*((a[1][0]\* a[2][2]) - a[2][0] \* a[1][2]) + a[0][2] \* ((a[1][0] \* a[2][1] - a[2][0]\*a[1][1]));

if ( determinant == 0)

printf("The given matrix is not invertible");

else

printf("The given matrix is invertible");

return 0;

}

1. Write a C program to sort a 1D array using pointer by applying Bubble sort technique.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | 7  70  40  80  10  200  30  60 | 10\n  30\n  40\n  60\n  70\n  80\n  200 | 10\n  30\n  40\n  60\n  70\n  80\n  200\n | Passed |

#include<stdio.h>

void sort(int \*a, int n);

int main()

{

int a[20];

int n,i;

scanf("%d",&n); // Enter number of elements to sort is taken from test case data

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

{

scanf("%d",&a[i]); // The elements of the array is taken from the test data

}

sort(a, n); // Calling the sorting function

//Printing the sorted array

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

{

printf("%d\n",a[i]);

}

return 0;

}

void sort(int \*a, int n)

{

int i,temp,j;

for(i=1;i<n;i++)

{

for(j=0;j<n-i;j++)

{

if(\*(a+j)>=\*(a+j+1))

{

temp = \*(a+j);

\*(a+j)= \*(a+j+1);

\*(a+j+1)= temp;

}

}

}

}

1. Write a C program to sort a given 1D array using pointer in ascending order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | 8  90  70  30  -10  -40  20  100  50 | -40\n  -10\n  20\n  30\n  50\n  70\n  90\n  100 | -40\n  -10\n  20\n  30\n  50\n  70\n  90\n  100\n | Passed |

#include <stdio.h>

int main()

{

int a[100],i, n;

scanf("%d",&n); //Number of elements of the array is taken from the test case data

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

{

scanf("%d",a+i); // Input the array elements

}

int j,t;

for (i=0; i<(n-1); i++)

{

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

{

if (\*(a+i)>\*(a+j))

{

t=\*(a+i);

\*(a+i)=\*(a+j);

\*(a+j)=t;

}

}

}

// Printing sorted array in ascending order

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

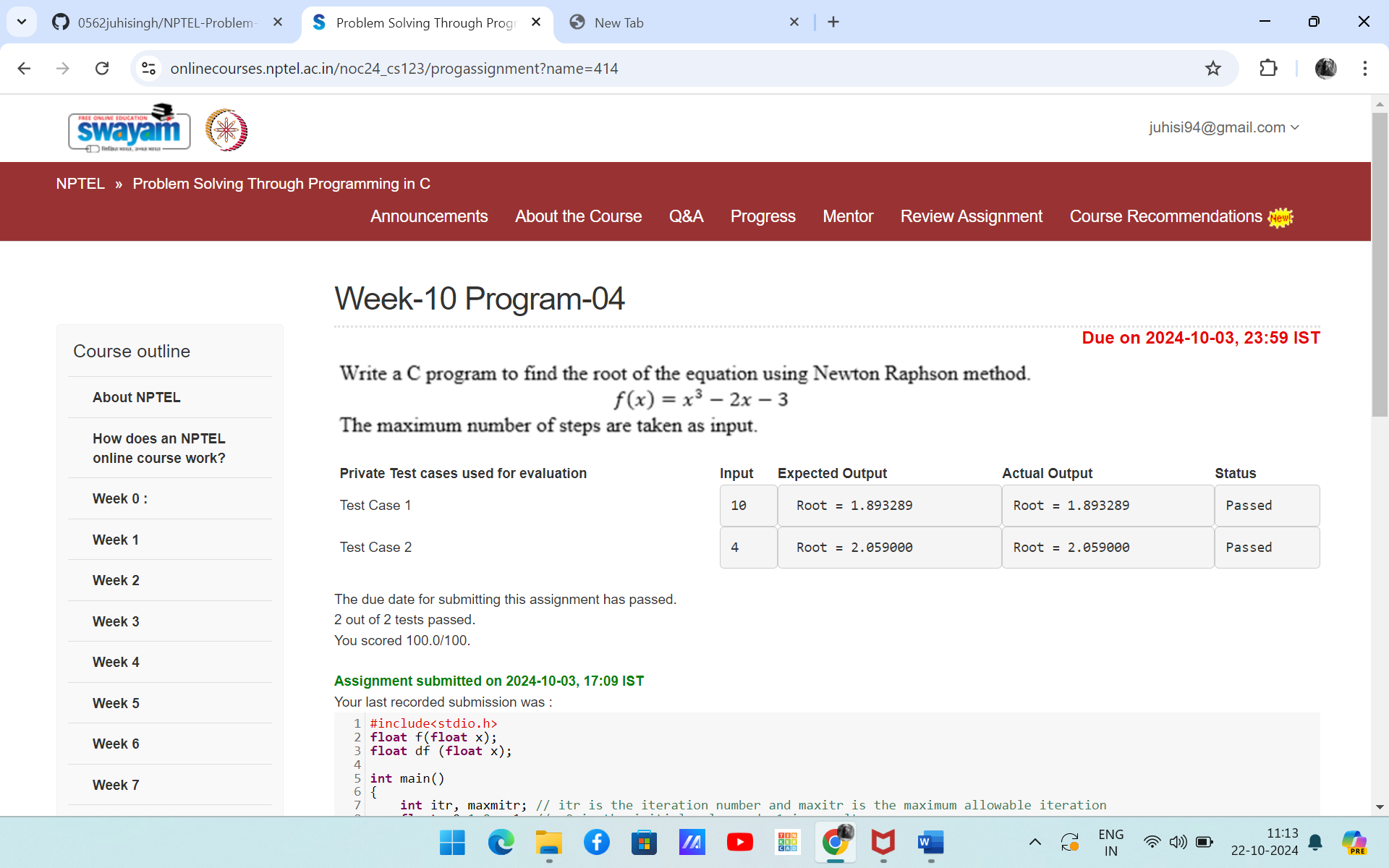
{

printf("%d\n",\*(a+i));

}

return 0;

}



#include<stdio.h>

float f(float x);

float df (float x);

int main()

{

int itr, maxmitr; // itr is the iteration number and maxitr is the maximum allowable iteration

float x0=1.0, x1; // x0 is the initial value and x1 is result

scanf("%d", &maxmitr); // Taken from the test cases

// use the printf statement as printf("Root = %8.6f\n", x1);

float h;

for (itr=1; itr<=maxmitr; itr++)

{

h=f(x0)/df(x0);

x1=x0-h;

x0=x1;

}

printf("Root = %8.6f", x1);

return 0;

}

float f(float x)

{

return x\*x\*x - 2\*x - 3;

}

float df (float x)

{

return 3\*x\*x-2;

}