1. Write a C program to reverse a word using Recursion. Input to the program is a string that is to be taken from the user and output is reverse of the input word. Note that you have to use recursion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | NPTELONLINE | The reversed string is : ENILNOLETPN | The reversed string is : ENILNOLETPN\n | Passed |
| Test Case 2 | MADRAS | The reversed string is : SARDAM | The reversed string is : SARDAM\n | Passed |

#include <stdio.h>

#define MAX 100

char \*reverse(char[]);

int main()

{

char str[MAX], \*rev;

//printf("Enter a String: ");

scanf("%s", str);

rev = reverse(str); //You have to write this function

printf("The reversed string is : %s\n", rev);

return 0;

}

char\* reverse(char str[])

{

static int i= 0;

static char rev[MAX];

if (\*str)

{

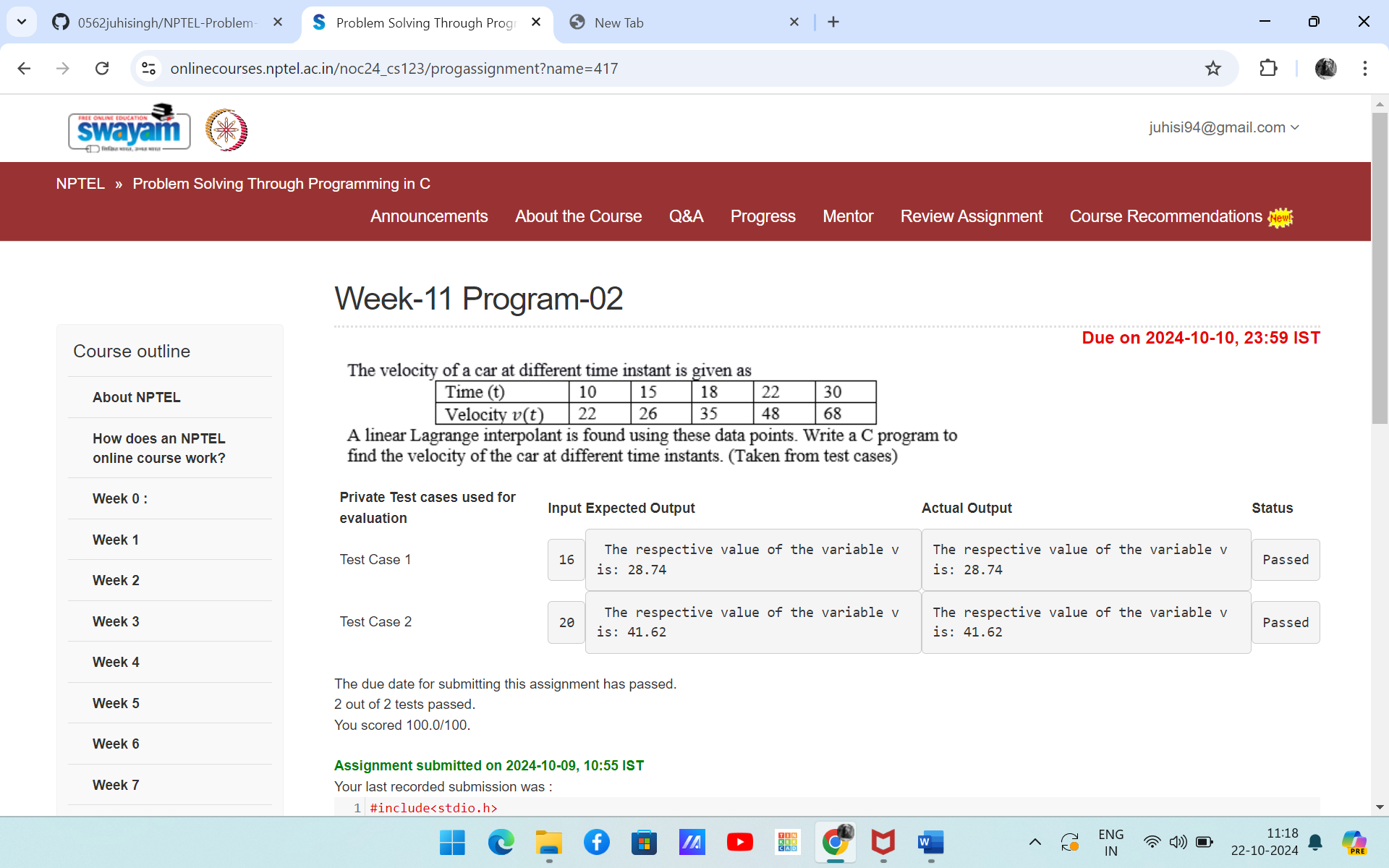
reverse(str+1);

rev[i++]= \*str;

}

return rev;

}



#include<stdio.h>

int main()

{

float t[100]={10,15,18,22,30}, v[100]={22,26,35,48,68};

float a; //Value of the t to find the respective value of v(t)

scanf("%f", &a); // This will be taken from test cases

int i,j;

float b, c, k =0;

for(i=0; i<5; i++)

{

b=1;

c=1;

for(j=0; j<5; j++)

{

if(j!=i)

{

b=b\*(a-t[j]);

c=c\*(t[i]-t[j]);

}

}

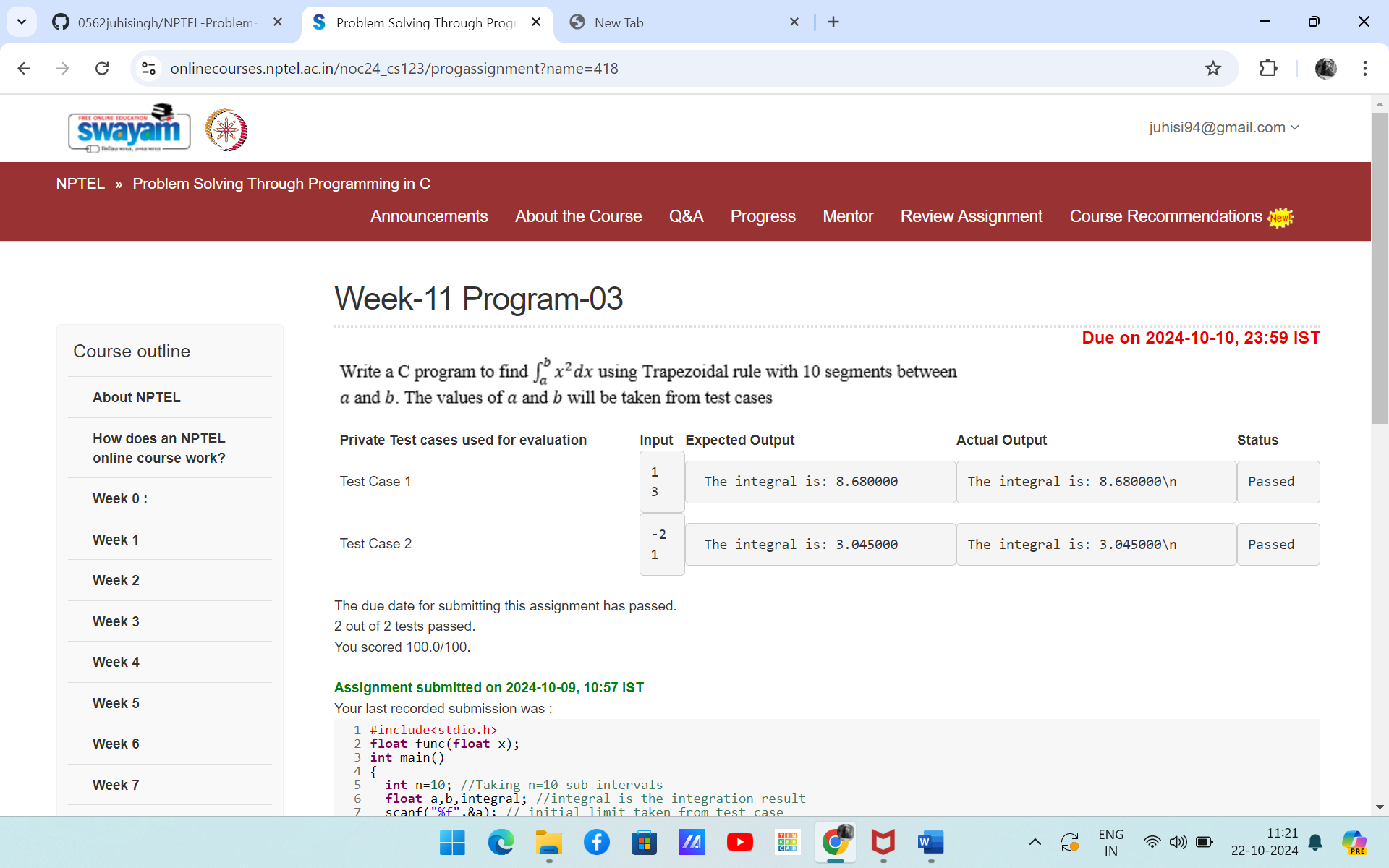
k=k+((b/c)\*v[i]);

}

printf("The respective value of the variable v is: %.2f", k);

return 0;

}



#include<stdio.h>

float func(float x);

int main()

{

int n=10; //Taking n=10 sub intervals

float a,b,integral; //integral is the integration result

scanf("%f",&a); // initial limit taken from test case

scanf("%f",&b); // Final limit taken from test case

//Use the printf statement as printf("The integral is: %0.6f\n",integral);

int i;

float h,x, sum=0;

if(b>a)

h=(b-a)/n;

else

h=-(b-a)/n;

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

{

x=a+i\*h;

sum=sum+func(x);

}

integral=(h/2)\*(func(a)+func(b)+2\*sum);

printf("The integral is: %0.6f\n",integral);

return 0;

}

float func(float x)

{

return x\*x;

}

1. Write a C program to check whether the given input number is Prime number or not using recursion. So, the input is an integer and output should print whether the integer is prime or not. Note that you have to use recursion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | 51 | 51 is not a prime number | 51 is not a prime number\n | Passed |
| Test Case 2 | 29 | 29 is a prime number | 29 is a prime number\n | Passed |

#include <stdio.h>

int checkPrime(int, int); //Function to check prime or not

int main()

{

int num, check;

scanf("%d", &num); //The number is taken from test case data

check = checkPrime(num, num/2);

if (check == 1)

{

printf("%d is a prime number\n", num);

}

else

{

printf("%d is not a prime number\n", num);

}

return 0;

}

int checkPrime(int num, int i)

{

if (i == 1)

return 1;

else

{

if (num % i == 0)

return 0;

else

return checkPrime(num, i - 1);

}

}