A Micro Project Report

on

Problem Solving using C Language

Submitted by

Unnava Naga Harshitha (23471A05D7)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier-1

NIRF rank in the band of 201-300 and is an ISO 9001:2015 certified Approved by AICTE, New Delhi, Permanently affiliated to JNTU Kakinada, Approved by AICTE, Accredited by NBA and accredited 'A+' grade by NAAC Narasaraopet-522601, Palnadu(Dt.), Andhra Pradesh, India

2024-2025

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Unnava Naga Harshitha, Roll No: 23471A05D7, a Second Year Student of the Department of Computer Science and Engineering, has completed the Micro Project Satisfactorily in "Problem Solving using C Language" for the Academic Year 2024-2025..

Project Co-Ordinator
Mr. Shaik Rafi, M.Tech., (Ph.D).
Asst. Professor

HEAD OF THE DEPARTMENT
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.
Professor

INDEX

Triangles Sorted By Their Areas

AIM:

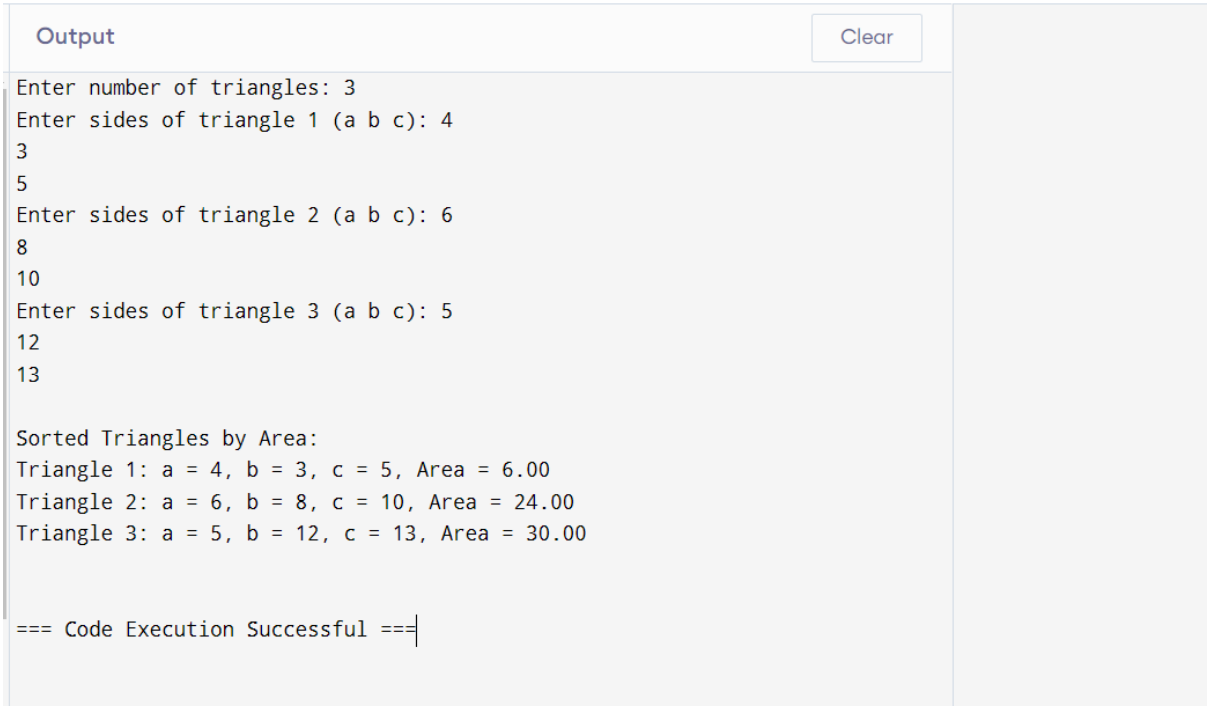
you are given triangles specifically their sides and print them in the same style but sorted by their areas from the smallest one to the largest one it is guaranteed that all the areas are different

```
#include <stdio.h>

#include <math.h>


int main()

{

   int n;

   printf("Enter number of triangles: ");

   scanf("%d", &n);


   int a[n], b[n], c[n];

   double area[n];


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

 {
```

```
        printf("Enter sides of triangle %d (a b c): ", i + 1);
        scanf("%d %d %d", &a[i], &b[i], &c[i]);

        double s = (a[i] + b[i] + c[i]) / 2.0;
        area[i] = sqrt(s * (s - a[i]) * (s - b[i]) * (s - c[i]));
    }

    for (int i = 0; i < n - 1; i++)
{
            for (int j = i + 1; j < n; j++)
        {
                  if (area[i] > area[j])
                {
                        int tempA = a[i], tempB = b[i], tempC = c[i];
                        double tempArea = area[i];
                        a[i] = a[j]; b[i] = b[j]; c[i] = c[j]; area[i] = area[j];
                        a[j] = tempA; b[j] = tempB; c[j] = tempC; area[j] = tempArea;
                 }
        }
 }
```

```
    printf("\nSorted Triangles by Area:\n");
    for (int i = 0; i < n; i++)
{
        printf("Triangle %d: a = %d, b = %d, c = %d, Area = %.2f\n", i + 1, a[i], b[i],
c[i], area[i]);
 }


    return 0;
}
```

Output:



Displaying numbers in words

Aim:

C program to read a number and display its digit in words

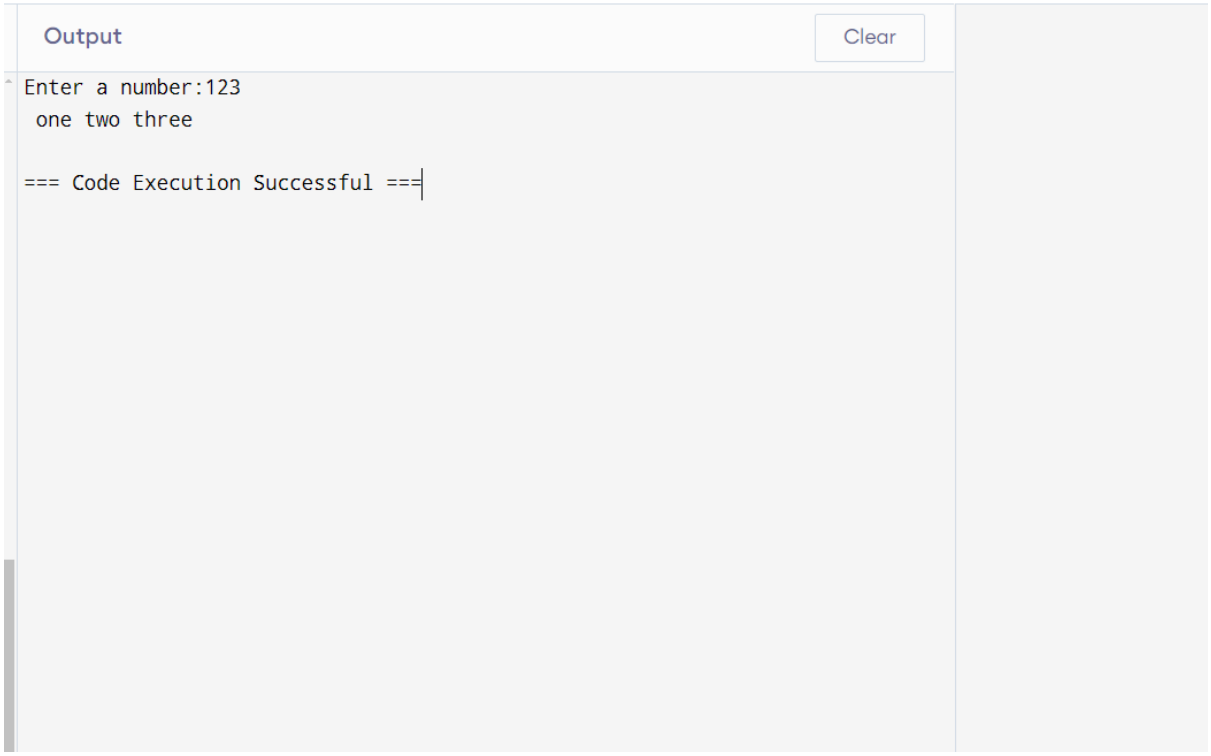
```
#include<stdio.h>

int main()

{

   int n,no,r;

   printf("Enter a number:");

   scanf("%d",&n);

   while(n>0)

   {

      no=(no*10)+(n%10);

      n=n/10;



   }

   while(no>0)

   {

      r=no%10;

      if(r==1)

      {
```

```
    printf(" one");


}
if(r==2)
{
    printf(" two");


}
if(r==3)
{
    printf(" three");


}
if(r==4)
{
    printf(" four");


}
if(r==5)
{
    printf(" five");
```

```
}
if(r==6)
{
   printf(" six");

}
if(r==7)
{
   printf(" seven");

}
if(r==8)
{
   printf(" eight");

}
if(r==9)
{
   printf(" nine");
}
no=no/10;
```

```
 }

}
```

Output:


Output
Clear
Enter a number:123
one two three
=== Code Execution Successful ===

Prime numbers

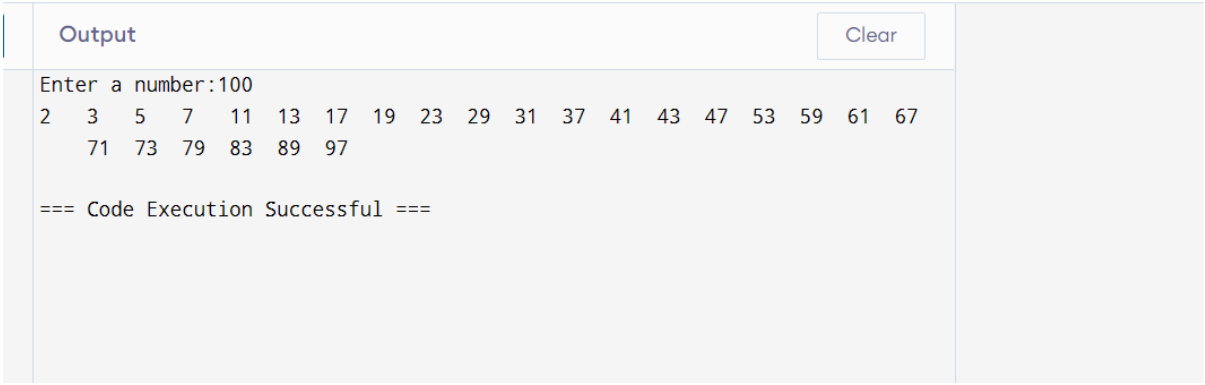
Aim:

C program to generate first N prime numbers where N is given by user

```
#include<stdio.h>
int main()
{
   int i,j,num,flage=0;
   printf("Enter a number:");
   scanf("%d",&num);
   for(i=1;i<=num;i++)
   {
      flage=0;
      for(j=1;j<=i;j++)
      {
         if(i%j==0)
         {
            flage+=1;
         }
      }
      if(flage==2)
      printf("%d\t",i);
   }
```

}

Output:


Output
Clear
Enter a number:100
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67
71 73 79 83 89 97
=== Code Execution Successful ===

Matchstick Game

Aim:

Write a program for a matchstick game being played between the computer and a user your program should ensure that the computer always wins. Rules for the game are as follows:

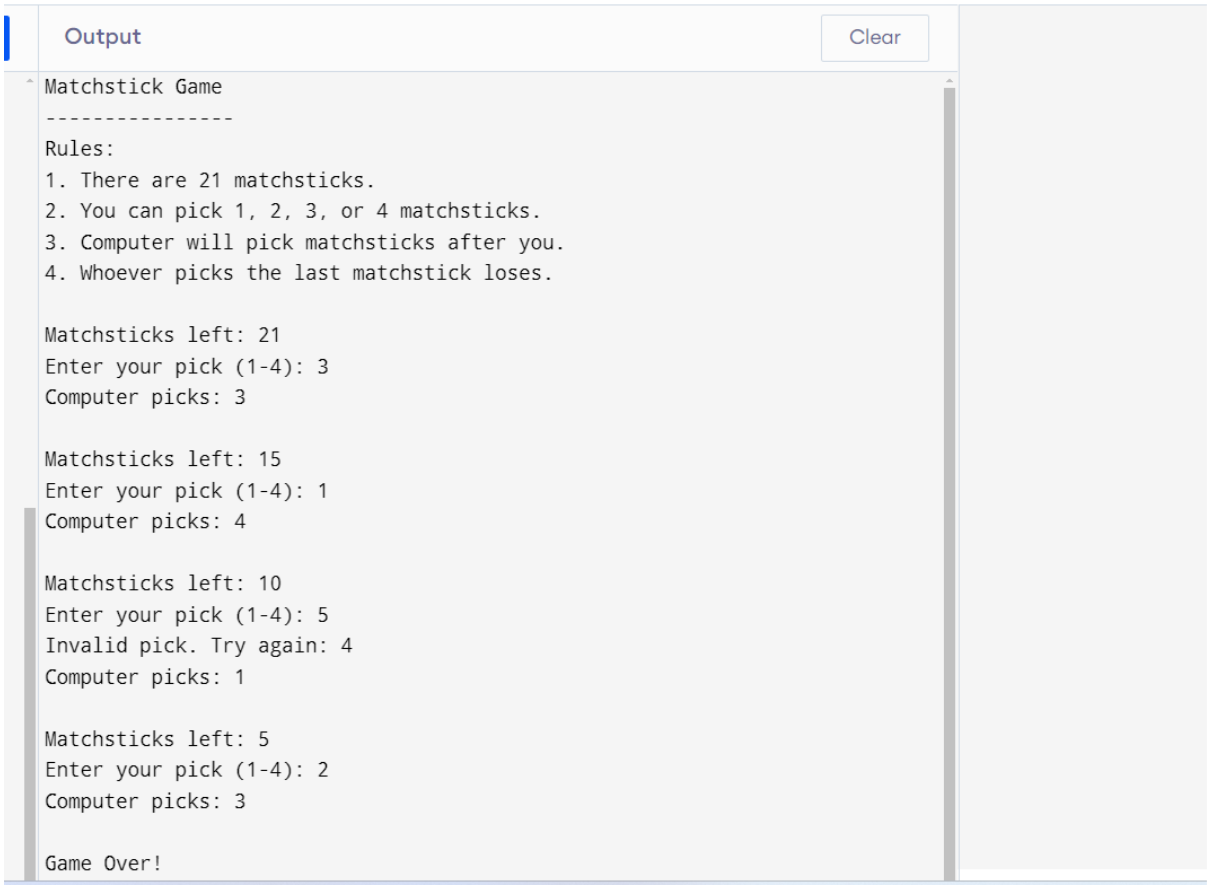
- **There are 21 matchsticks.**
- **The computer asks the user to pick 1,2,3 or 4 matchsticks**
- **After the user picks, the computer does it picking.**
- **Whoever is forced to pick up the last matchstick loses the game**

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

int main()
{
   int matchsticks = 21;
   int userPick, computerPick;
```

```
printf("Matchstick Game\n");
printf("----------------\n");
printf("Rules:\n");
printf("1. There are 21 matchsticks.\n");
printf("2. You can pick 1, 2, 3, or 4 matchsticks.\n");
printf("3. Computer will pick matchsticks after you.\n");
printf("4. Whoever picks the last matchstick loses.\n");

while (matchsticks > 0)
{
    printf("\nMatchsticks left: %d\n", matchsticks);
    printf("Enter your pick (1-4): ");
    scanf("%d", &userPick);
    while (userPick < 1 || userPick > 4 || userPick > matchsticks)
    {
        printf("Invalid pick. Try again: ");
        scanf("%d", &userPick);
    }

    matchsticks -= userPick;
    if (matchsticks > 0)
    {
        computerPick = matchsticks % 5;
        if (computerPick == 0)
```

```
            {
                computerPick = 4;
            }
            printf("Computer picks: %d\n", computerPick);
            matchsticks -= computerPick;
        }
        if (matchsticks == 0)
        {
            printf("\nGame Over!\n");
            if (userPick > 0)
            {
                printf("Computer wins!\n");
            }
             else
            {
                printf("You win!\n");
            }
        }
    }

    return 0;
}
```

Output:


Output
Clear
Matchstick Game

Rules:
1. There are 21 matchsticks.
2. You can pick 1, 2, 3, or 4 matchsticks.
3. Computer will pick matchsticks after you.
4. Whoever picks the last matchstick loses.
Matchsticks left: 21
Enter your pick (1-4): 3
Computer picks: 3
Matchsticks left: 15
Enter your pick (1-4): 1
Computer picks: 4
Matchsticks left: 10
Enter your pick (1-4): 5
Invalid pick. Try again: 4
Computer picks: 1
Matchsticks left: 5
Enter your pick (1-4): 2
Computer picks: 3
Game Over!