A Micro Project Report

on

Problem Solving using C Language

Submitted by

Kichamsetty Akhila (23471A05DX)

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 **kichamsetty Akhila** , **Roll No: 23471A05DX**, 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

Dr. Rama Krishna. Eluri, M.Tech., Ph.D.

Asst. Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.

Professor

INDEX

prime numbers count from minimum to maximum

AIM:

Write a C program to count numbers of prime numbers in given minimum to maximum Ranges

Source code :

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

int main()
{
    int minimum, maximum, flag, count=0, i, j;
    clrscr();
    printf("Enter minimum number: ");
    scanf("%d", &minimum);
    printf("Enter maximum number: ");
    scanf("%d", &maximum);
    for(i=minimum; i<=maximum; i++)
    {
        flag = 0;
        for(j=2; j <= i/2; j++)
        {
            if(i%j==0)
            {
                flag=1;
                break;
            }
        }
        if(flag==0 && i>=2)
```

```
        {
            count++;

        }
    }
    printf("\n Prime Count = %d", count);
    getch();
}
```

Input:

enter minimum value:1
 enter maximum value:20

Output:

 prime count = 8

```
enter minimum value :1
enter maximum value 20

 prime count =8
```

Armstrong Numbers from minimum to maximum

AIM:

Write a C program to generate armstrong numbers in given minimum to maximum to ranges

Source code :

```
#include<stdio.h>

#include<math.h>


int main()

{

        int minimum, maximum, Count, number, sum, rem, i;

        printf("Enter minimum number: ");

        scanf("%d", &minimum);

        printf("Enter maximum number: ");

        scanf("%d", &maximum);


    for(i=minimum; i<=maximum; i++)

    {

            number = i;

            Count = 0;
```

```
while(number != 0)
{
        Count++;
        number = number/10;
}
number = i;
sum = 0;
while(number != 0)
{
        rem = number%10;
        sum = sum + pow(rem, Count);
            number = number/10;
        }
        if(sum == i)
        {
            printf("%d\t", i);
        }
}
        return(0);
}
```

Input:

Enter a minimum value:10

Enter a maximum value:500

Output:

153 370 371 407

```
Enter minimum number: 10
Enter maximum number: 500
153     370     371     407
```

First N prime numbers

AIM:

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

Source code :

```
#include<stdio.h>

int main()

{

        int n, count=1, flag, i=2, j;

        printf("Enter a number:");

        scanf("%d", &n);

while(count <= n)

{

        flag = 0;

        for(j=2; j <= i/2; j++)

        {

                if(i%j==0)

{

 flag=1;

 break;

}
```

```
        }
        if(flag==0)
        {
                printf("%d\t",i);
                count++;
                }
                i++;
        }
        return(0);
}
```

Input:

Enter a value:5

Output:

2 3 5 7 11

```
Enter a number:5
2       3       5       7       11
```

Perfect numbers from minimum to maximum

AIM:

Write a C program to generate perfect numbers in given minimum to maximum Ranges

Source code :

```
#include<stdio.h>

int main()

{

        int n, count=1, flag, i=2, j;

        printf("Enter a number:");

        scanf("%d", &n);

while(count <= n)

{

        flag = 0;

        for(j=2; j <= i/2; j++)

        {

                if(i%j==0)

{

 flag=1;

 break;

}
```

```
        }
        if(flag==0)
        {
                printf("%d\t",i);
                count++;
        }
                }
                i++;
        }
        return(0);
}
```

Input:

Enter a minimum value:1

Enter a maximiimum value:500

Output:

6 28 496

```
Enter the minimum number: 1
Enter the maximum number: 500
6       28      496
```