

A Micro Project Report

on

Problem Solving using C Language

Submitted by

MUDAVATH MUTYALU NAIK(23471A05HX)



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 **Mudavath Mutyalu Naik**, Roll No: **23471A05HX**, 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

S.No	Description
1	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.
2	C Program to Read a Number and Displaying Its Digit in Words
3	C Program to Generate First N prime Numbers Where N is Given by User
4	<p>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.</p> <p>- The computer asks the player to pick 1, 2, 3, or 4 matchsticks. - After the person picks, the computer does its picking.</p> <p>Whoever is forced to pick up the last matchstick loses the game.</p>

TRIANGLES AREAS FROM THE SMALLEST ONE TO THE LARGEST ONE.

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.

Source code:

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>


// Structure to store triangle information
struct Triangle {
    int a, b, c; // sides
    double area; // to store area
};


// Function to calculate area using Heron's formula
double calculateArea(int a, int b, int c) {
    double s = (a + b + c) / 2.0;
    return sqrt(s * (s - a) * (s - b) * (s - c));
}
```

```
// Comparison function for qsort

int compare(const void* v1, const void* v2) {
    struct Triangle *t1 = (struct Triangle *)v1;
    struct Triangle *t2 = (struct Triangle *)v2;

    if (t1->area < t2->area) return -1;
    if (t1->area > t2->area) return 1;
    return 0;
}

int main() {
    int n;

    printf("Enter number of triangles: ");
    scanf("%d", &n);

    // Array of triangles
    struct Triangle tr[n];

    // Input triangles
    for(int i = 0; i < n; i++) {
        printf("Enter sides of triangle %d: ", i+1);
        scanf("%d %d %d", &tr[i].a, &tr[i].b, &tr[i].c);
    }
}
```

```
// Calculate area

tr[i].area = calculateArea(tr[i].a, tr[i].b, tr[i].c);
}


// Sort triangles based on area

qsort(tr, n, sizeof(struct Triangle), compare);


// Print sorted triangles

printf("\nTriangles sorted by area:\n");

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

    printf("Triangle sides: %d %d %d (Area: %.2f)\n",

        tr[i].a, tr[i].b, tr[i].c, tr[i].area);

}


return 0;

}
```

Input:

Enter number of triangles: 2

Enter sides of triangle 1: 2

3

4

Enter sides of triangle 2: 1

1

1

Output:

Triangles sorted by area:

Triangle sides: 1 1 1 (Area: 0.43)

Triangle sides: 2 3 4 (Area: 2.90)

READ A NUMBER AND DISPLAYING ITS DIGIT IN WORDS

AIM: C Program to Read a Number and Displaying Its Digit in Words

Source code:

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

```
void displayDigitInWords(int digit) {
```

```
    switch (digit) {
```

```
        case 0: printf("Zero "); break;
```

```
        case 1: printf("One "); break;
```

```
        case 2: printf("Two "); break;
```

```
        case 3: printf("Three "); break;
```

```
        case 4: printf("Four "); break;
```

```
        case 5: printf("Five "); break;
```

```
        case 6: printf("Six "); break;
```

```
        case 7: printf("Seven "); break;
```

```
        case 8: printf("Eight "); break;
```

```
        case 9: printf("Nine "); break;
```

```
    }
```

```
}
```

```
void displayNumberInWords(int number) {
```

```
    int digits[10], i = 0;
```

```
    // Handle zero case
```



```
if (number == 0) {
    displayDigitInWords(0);
    return;
}
// Extract digits in reverse order
while (number > 0) {
    digits[i++] = number % 10;
    number /= 10;
}
// Print digits in correct order
for (int j = i - 1; j >= 0; j--) {
    displayDigitInWords(digits[j]);
}
}

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);
    printf("Number in words: ");
    displayNumberInWords(number);
    printf("\n");
    return 0;
}
```

INPUT:1

Enter a number: 2004

Output:1

Number in words: Two Zero Zero Four.

Input:2

Enter a number:93818

Output:2

Number in words : Nine Three Eight One Eight

Input:3

Enter a number : 97022

Output:3

Number in Words : Nine Seven Zero Two Two

TO GENERATE FIRST N PRIME NUMBERS

AIM: C Program to generate first N prime numbers where N is given by user.

Source code:

```
#include <stdio.h>

#include <stdbool.h>

bool isPrime(int num) {
    if (num <= 1) return false;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) return false;
    }
    return true;
}

int main() {
    int N;
    printf("Enter the value of N: ");
    scanf("%d", &N);

    if (N <= 0) {
        printf("Please enter a positive number!\n");
        return 1;
    }
```

```
// Array to store prime numbers

int primes[N];

int count = 0, num = 2;


// Generate and store prime numbers
while (count < N) {
    if (isPrime(num)) {
        primes[count] = num;
        count++;
    }
    num++;
}


// Print prime numbers
printf("First %d prime numbers are:\n", N);
for (int i = 0; i < N; i++) {
    printf("%d ", primes[i]);
}

printf("\n");


return 0;
}
```

Input:1

Enter the value of N: 5

Output:1

First 5 prime numbers are:

2 3 5 7 11

Input:2

Enter the value of N: 2

Output:2

First 2 prime numbers are:

2 3

Input:3

Enter the value of N:7

Output:3

First 7 prime numbers:

2 3 5 7 11 13 17

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 player to pick 1, 2, 3, or 4 matchsticks.
- After the person picks, the computer does its picking.
- Whoever is forced to pick up the last matchstick loses the game.

Source code:

```
#include <stdio.h>

int main() {
    int matchsticks = 21, user_pick, computer_pick;
    printf("\n---MATCHSTICK GAME---\n");
    printf("Total Matchsticks: 21\n");
    printf("You can pick 1, 2, 3 or 4 matchsticks\n");
    printf("The player who picks the last matchstick loses\n\n");
    while (matchsticks > 1) {
        // User's turn
        do {
            printf("\nMatchsticks remaining: %d\n", matchsticks);
            printf("Your turn - Pick matchsticks (1-4): ");
            scanf("%d", &user_pick);
```

```
    if (user_pick < 1 || user_pick > 4 || user_pick > matchsticks) {  
        printf("Invalid pick! Choose between 1 to 4 matchsticks\n");  
    }  
} while (user_pick < 1 || user_pick > 4 || user_pick > matchsticks);  
matchsticks -= user_pick;  
// Check if user picked the last matchstick  
if (matchsticks == 1) {  
    printf("\nOnly 1 matchstick remaining...\n");  
    printf("Computer wins! You have to pick the last matchstick!\n");  
    break;  
}  
// Computer's turn  
// Winning strategy: Make sure to leave (5n + 1) matchsticks  
computer_pick = 5 - user_pick;  
printf("Computer picks: %d matchstick(s)\n", computer_pick);  
matchsticks -= computer_pick;  
if (matchsticks == 1) {  
    printf("\nOnly 1 matchstick remaining...\n");  
    printf("You must pick the last matchstick. Computer wins!\n");  
    break;  
}  
}  
return 0;  
}
```

Output:

---MATCHSTICK GAME---

Total Matchsticks: 21

You can pick 1, 2, 3 or 4 matchsticks

The player who picks the last matchstick loses

Matchsticks remaining: 21

Your turn - Pick matchsticks (1-4): 4

Computer picks: 1 matchstick(s)

Matchsticks remaining: 16

Your turn - Pick matchsticks (1-4): 3

Computer picks: 2 matchstick(s)

Matchsticks remaining: 11

Your turn - Pick matchsticks (1-4): 4

Computer picks: 1 matchstick(s)

Matchsticks remaining: 6

Your turn - Pick matchsticks (1-4): 4

Computer picks: 1 matchstick(s)

Only 1 matchstick remaining...

You must pick the last matchstick. Computer wins!

