

# **A Micro Project Report**

**on**

## **Problem Solving using C Language**

Submitted by  
**Arigila Sowmya (23471A05D9)**



**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 **Arigila Sowmya**, **Roll No: 23471A05D9**, 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.  
Ph.D. Asst. Professor

**HEAD OF THE DEPARTMENT**

**Dr. S. N. Tirumala Rao**, M.Tech.,  
Professor

# INDEX

S.No	Description
1.	Read Records of n different students in structure and sort on the basis of marks in Ascending order
2.	Employee Record in descending order by age in structure
3.	C program to convert Roman number to decimal number
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:</p> <ul style="list-style-type: none"><li>• There are 21 matchsticks.</li><li>• The computer asks the player to pick 1,2,3 or 4 matchsticks</li><li>• After the person picks, the computer does its picking</li><li>• Whoever is forced to pick up the last matchstick loses the game</li></ul>

## Students records in Ascending order

### AIM:

**Read Records of n different students in structure and sort on the basis of marks in Ascending order**

```
#include<stdio.h>
struct student
{
char name[30];
float marks;
};
int main()
{
struct student s[20], temp;
int i,j,n;
printf ("Enter n:\n");
scanf ("%d",&n);
printf ("Enter name and marks of student:\n");
for(i=0;i< n;i++)
{
scanf("%s%f",s[i].name, &s[i].marks);
}
for(i=0;i< n-1;i++)
{
for(j=i+1;j< n;j++)
{
if(s[i].marks>s[j].marks)
{
temp = s[i];
s[i] = s[j];
s[j] = temp;
}
}
}
printf("Sorted records are:\n");
for(i=0;i< n;i++)
```

```
{  
printf("Name: %s\n", s[i].name);  
printf("Marks: %0.2f\n\n", s[i].marks);  
}  
return 0;  
}
```

### **Inout:**

Enter n:

5

Enter name and marks of student:

sowmya 40

munni 35

sreya 50

jhansi 38

jwala 42

### **Output:**

Sorted records are:

Name: munni

Marks: 35.00

Name: jhansi

Marks: 38.00

Name: sowmya

Marks: 40.00

Name: jwala

Marks: 42.00

Name: sreya

Marks: 50.00

## Employee records in Descending order

### AIM:

#### Employee Record in descending order by age in structure

```
#include<stdio.h>
struct student
{
char name[30];
int id;
int age;
};
int main()
{
struct student s[20], temp;
int i,j,n;
printf("Enter n:\n");
scanf("%d",&n);
printf("Enter employee name ,id and age:\n");
for(i=0;i< n;i++)
{
scanf("%s%d%d",s[i].name,&s[i].id, &s[i].age);
}
for(i=0;i< n-1;i++)
{
for(j=i+1;j< n;j++)
{
if(s[i].age<s[j].age)
{
temp = s[i];
s[i] = s[j];
s[j] = temp;
}
}
}
printf("Sorted records are:\n");
```

```
for(i=0;i< n;i++)  
{  
printf("Name: %s\n", s[i].name);  
printf("id: %d\n",s[i].id);  
printf("Marks: %d\n\n", s[i].age);  
}  
return 0;  
}
```

### **Inout:**

Enter n:

3

Enter employee name ,id and age:

Sana 4567 40

Jani 4568 50

Hari 4569 51

### **Output:**

Sorted records are:

Name: hari

id: 4569

Marks: 51

Name: jani

id: 4568

Marks: 50

Name: sana

id: 4567

Marks: 40

## Convert Roman number to decimal number

### AIM:

#### C program to convert Roman number to decimal number

```
#include <stdio.h>
#include<string.h>
int digit(char);
int main() {
    char romannumber[1000];
    int i=0;
    long int number=0;
    printf("enter any roman number(valid digits are I,V,X,L,C,D,M):\n");
    scanf("%s",romannumber);
    while(romannumber[i]!='\0')
    {
        if(digit(romannumber[i])>=digit(romannumber[i+1])){
            number=number+digit(romannumber[i]);}
        else{
            number=number+(digit(romannumber[i+1])-digit(romannumber[i]));
            i++;
        }
        i++;
    }
    printf("its decimal value is:%ld",number);
    return 0;
}
int digit(char c)
{
    int value=0;
    switch(c)
    {
        case 'I':value=1;
        break;
        case 'V':value=5;
        break;
```



```
case 'X':value=10;
break;
case 'L':value=50;
break;
case 'C':value=100;
break;
case 'D':value=500;
break;
case 'M':value=1000;
break;
case '\0':value=0;
break;
default: value=-1;
}
return value;
}
```

**Inout:**

enter any roman number(valid digits are I,V,X,L,C,D,M):  
XIII

**Output:**

its decimal value is:13

**Inout:**

enter any roman number(valid digits are I,V,X,L,C,D,M):  
VL

**Output:**

its decimal value is:45

# Matchstick game between the Computer and User

## 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

```
#include<stdio.h>
int main()
{
int m=21,p,c;
while(m>1)
{
printf("no of match sticks left=%d\n",m);
printf("pick 1 or 2 or 3 or 4 matches\n");
scanf("%d",&p);
if(p>=1 || p<=4)
{
m=m-p;
printf("no of match sticks left after person picked=%d\n",m);
if(m==1)
{
printf("person lost game");
break;
}
}
c=5-p;
printf("out of computer picked %d\n",c);
m=m-c;
printf("no of match sticks left after computer picked=%d\n",m);
if(m==1)
```

```
{  
printf("computer wins game");  
break;  
}  
}  
return 0;  
}
```

**Input:**

no of match sticks left=21  
pick 1 or 2 or 3 or 4 matches  
3

**Output:**

no of match sticks left after person picked=18  
out of computer picked 2  
no of match sticks left after computer picked=16

**Input:**

no of match sticks left=16  
pick 1 or 2 or 3 or 4 matches  
4

**Output:**

no of match sticks left after person picked=12  
out of computer picked 1  
no of match sticks left after computer picked=11

**Input:**

no of match sticks left=11  
pick 1 or 2 or 3 or 4 matches  
2

**Output:**

no of match sticks left after person picked=9  
out of computer picked 3  
no of match sticks left after computer picked=6

**Input:**

no of match sticks left=6

pick 1 or 2 or 3 or 4 matches

3

**Output:**

no of match sticks left after person picked=3

out of computer picked 2

no of match sticks left after computer picked=1

computer wins game