INDEX

Different sorting methods for problem sloving

1.Read Records of n Different Students in Structure & Sort on the Basis of Marks in Ascending Order. #include <stdio.h> struct Student

```
{
   char name[50];
int marks;
};
int main()
{    int
n;
   printf("Enter number of students: ");
scanf("%d", &n);    struct Student
students[n];
   for (int i = 0; i < n; i++)
  {
      printf("\nEnter name of student %d: ", i + 1);
scanf(" %[^\n]%*c", students[i].name);
printf("Enter marks of student %d: ", i + 1);
scanf("%d", &students[i].marks);

   }
   for (int i = 0; i < n-1; i++)
```

```
    {
        for (int j = i+1; j < n; j++)
     {
           if (students[i].marks > students[j].marks)
      {
              struct Student temp = students[i];
students[i] = students[j];              students[j]
= temp;
       }
       }
       }
    printf("\nStudents sorted by marks (ascending order):\n");
    for (int i = 0; i < n; i++) {        printf("Name: %s, Marks: %d\n",
students[i].name, students[i].marks);
    }
 return 0;
}
```

Output

```
Enter number of students: 2

Enter name of student 1: uday
Enter marks of student 1: 90

Enter name of student 2: kiran
Enter marks of student 2: 94

Students sorted by marks (ascending order):
Name: uday, Marks: 90
Name: kiran, Marks: 94
```

2. Employee Record in Descending Order by Age in Structure.

```
#include <stdio.h> struct Employee
{
   char name[50];
   int age;
};
int main()
{
 int n;     printf("Enter number of
employees: ");     scanf("%d", &n);

   struct Employee employees[n];
```

```
    for (int i = 0; i < n; i++)
    {
        printf("\nEnter name of employee %d: ", i + 1);
scanf(" %[^\n]%*c", employees[i].name);
printf("Enter age of employee %d: ", i + 1);
scanf("%d", &employees[i].age);
     }
    for (int i = 0; i < n-1; i++)
      {
       for (int j = i+1; j < n; j++)
        {
          if (employees[i].age < employees[j].age)
            {
              struct Employee temp = employees[i];
employees[i] = employees[j];              employees[j]
= temp;
            }
         }
    }
    printf("\nEmployees sorted by age (descending order):\n");
    for (int i = 0; i < n; i++)
    {
        printf("Name: %s, Age: %d\n", employees[i].name, employees[i].age);
```

```
    }
 return 0;
}
```

Output

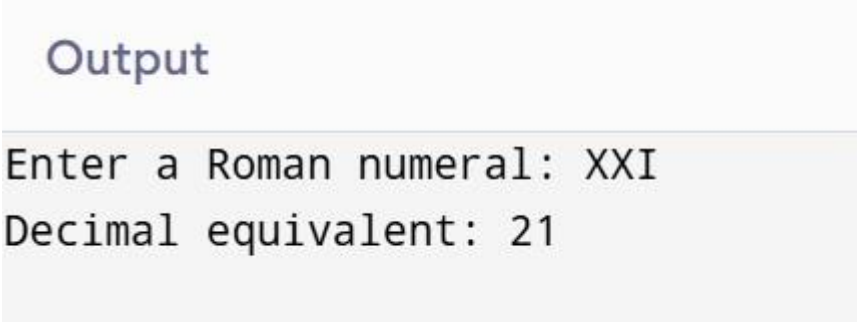
```
Enter number of employees: 2

Enter name of employee 1: uday
Enter age of employee 1: 18

Enter name of employee 2: kiran
Enter age of employee 2: 21

Employees sorted by age (descending order):
Name: kiran, Age: 21
Name: uday, Age: 18
```

3.C Program to Convert Roman Number to Decimal Number

```
#include <stdio.h> #include
<string.h> int
romanToDecimal(char roman[])
{
   int decimalValue = 0;     int
romanValues[256] = {0};
romanValues['I'] = 1;
romanValues['V'] = 5;
```

```
romanValues['X'] = 10;
romanValues['L'] = 50;
romanValues['C'] = 100;
romanValues['D'] = 500;
romanValues['M'] = 1000;
int length = strlen(roman);
for (int i = 0; i < length; i++)
 {
      int currentValue = romanValues[roman[i]];
      if (i + 1 < length && romanValues[roman[i]] < romanValues[roman[i +
1]])
        {
          decimalValue -= currentValue;
        }
else
 {
          decimalValue += currentValue;
       }
   }
   return decimalValue;
}
int main()
{
```

```
   char roman[100];       printf("Enter a
Roman  numeral:  ");          scanf("%s",
roman);     if (strlen(roman) == 0)
{
      printf("Invalid input: Empty Roman numeral!\n");
      return 1;
   }
   int decimal = romanToDecimal(roman);
printf("Decimal equivalent: %d\n", decimal);     return
0;
}
```



4. 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 matchsticks = 21;    int playerPick,
computerPick;    printf("Welcome to the
Matchstick Game!\n");
    printf("There are 21 matchsticks. You can pick 1 to 4 matchsticks each
  turn.\n");   while (matchsticks > 1)
      {
       printf("\nThere are %d matchsticks left.\n", matchsticks);
  printf("How many matchsticks will you pick (1-4)? ");
  scanf("%d", &playerPick);    if (playerPick < 1 || playerPick
  > 4)
       {
         printf("Invalid choice! You can only pick between 1 and 4
  matchsticks.\n");
         continue;
       }
      matchsticks -= playerPick;
  if (matchsticks == 1)
       {
         printf("You are forced to pick the last matchstick. You lose!\n");
         break;
       }
      if (matchsticks % 5 == 0)
       {
```

```
            computerPick = 1;
        }
 Else
      {
            computerPick = matchsticks % 5;
       }
  printf("Computer picks %d matchstick(s).\n", computerPick);
matchsticks -= computerPick;        if (matchsticks == 1)
        {
        printf("The computer is forced to pick the last matchstick. You
win!\n");
            break;
        }
    }
 return 0;
}
```

Output Clear

```
Welcome to the Matchstick Game!
There are 21 matchsticks. You can pick 1 to 4 matchsticks each turn.
The player who is forced to pick the last matchstick loses.

There are 21 matchsticks left.
How many matchsticks will you pick (1-4)? 3
Computer picks 3 matchstick(s).

There are 15 matchsticks left.
How many matchsticks will you pick (1-4)? 2
Computer picks 3 matchstick(s).

There are 10 matchsticks left.
How many matchsticks will you pick (1-4)? 1
Computer picks 4 matchstick(s).

There are 5 matchsticks left.
How many matchsticks will you pick (1-4)? 4
You are forced to pick the last matchstick. You lose!
```