**A Micro Project Report**

**on**

**Problem Solving using C Language**

Submitted by

**Janigarla shebha(23471A05FV)**

****

**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**

**2024-20****25**

**NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

**This is to certify that Janigarla shebha, Roll No: 23471A05FVa 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 HEAD OF THE DEPARTMENT

**Mr. M. Venkata Rao, M.Tech.**  **Dr. S. N. Tirumala Rao,** **M.Tech., Ph.D.**

**Asst. Professor Professor**

**INDEX**

|  |  |
| --- | --- |
| **S.No** | **Description** |
|  | Read Records of n Different Students in Structure & Sort on the Basis of Marks in Ascending Order |
|  | Employee Record in Descending Order by Age in Structure |
|  | C Program to Convert Roman Number to Decimal Number |
|  | Write a program for a matchstick game played between the computer and a user. |

**AIM**:

**Read Records of n Different Students in Structure & Sort on the Basis of Marks in Ascending Order**

#include <stdio.h>

#include <stdlib.h>

struct student {

char name[50];

int rollNo;

int marks;

};

void swap(struct student \*x, struct student \*y) {

struct student temp = \*x;

\*x = \*y;

\*y = temp;

}

void sortStudentsByMarks(struct student \*students, int n) {

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

for (int j = 0; j < n - i - 1; j++) {

if (students[j].marks > students[j + 1].marks) {

swap(&students[j], &students[j + 1]);

}

}

}

}

int main() {

int n;

printf("Enter the number of students: ");

scanf("%d", &n);

struct student \*students = (struct student \*)malloc(n \* sizeof(struct student));

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

printf("Enter details for student %d:\n", i + 1);

printf("Name: ");

scanf("%s", students[i].name);

printf("Roll No: ");

scanf("%d", &students[i].rollNo);

printf("Marks: ");

scanf("%d", &students[i].marks);

}

sortStudentsByMarks(students, n);

printf("\nSorted student records:\n");

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

printf("Name: %s, Roll No: %d, Marks: %d\n", students[i].name, students[i].rollNo, students[i].marks);

}

free(students);

return 0;

}

**INPUT &OUTPUT:**

Enter the number of students: 2

Enter details for student 1:

Name: lalitha

Roll No: 69

Marks: -22

Enter details for student 2:

Name: sheba

Roll No: 2

Marks: 99

Sorted student records:

Name: lalitha, Roll No: 69, Marks: -22

Name: sheba, Roll No: 2, Marks: 99

**AIM**:

**Employee Record in Descending Order by Age in Structure**

#include <stdio.h>

#include <stdlib.h>

struct Employee {

int id;

char name[50];

int age;

};

int compare(const void \*a, const void \*b) {

struct Employee \*empA = (struct Employee \*)a;

struct Employee \*empB = (struct Employee \*)b;

return empB->age - empA->age;

}

int main() {

struct Employee employees[] = {

{1, "Alice", 28},

{2, "Bob", 35},

{3, "Charlie", 25},

{4, "David", 40},

{5, "Eve", 30}

};

int n = sizeof(employees) / sizeof(employees[0]);

qsort(employees, n, sizeof(struct Employee), compare);

printf("Employee records sorted by age in descending order:\n");

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

printf("ID: %d, Name: %s, Age: %d\n", employees[i].id, employees[i].name, employees[i].age);

}

return 0;

}

**INPUT & OUTPUT:**

Employee records sorted by age in descending order:

ID: 4, Name: David, Age: 40

ID: 2, Name: Bob, Age: 35

ID: 5, Name: Eve, Age: 30

ID: 1, Name: Alice, Age: 28

ID: 3, Name: Charlie, Age: 25

**AIM**:

**C Program to Convert Roman Number to Decimal Number**#include <stdio.h>

#include <string.h>

int romanToDecimal(char \*roman) {

int decimal = 0;

int i, len = strlen(roman);

for (i = 0; i < len; i++) {

if (roman[i] == 'I') {

if (i + 1 < len && (roman[i + 1] == 'V' || roman[i + 1] == 'X')) {

decimal -= 1;

} else {

decimal += 1;

}

} else if (roman[i] == 'V') {

decimal += 5;

} else if (roman[i] == 'X') {

if (i + 1 < len && (roman[i + 1] == 'L' || roman[i + 1] == 'C')) {

decimal -= 10;

} else {

decimal += 10;

}

} else if (roman[i] == 'L') {

decimal += 50;

} else if (roman[i] == 'C') {

if (i + 1 < len && (roman[i + 1] == 'D' || roman[i + 1] == 'M')) {

decimal -= 100;

} else {

decimal += 100;

}

} else if (roman[i] == 'D') {

decimal += 500;

} else if (roman[i] == 'M') {

decimal += 1000;

}

}

return decimal;

}

int main() {

char roman[100];

printf("Enter a Roman numeral: ");

scanf("%s", roman);

int decimal = romanToDecimal(roman);

printf("The decimal value is: %d\n", decimal);

return 0;

}

**INPUT & OUTPUT:**

Enter a Roman numeral: XLII

The decimal value is: 42

**AIM**:

**Write a program for a matchstick game played between the computer and a user.**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void printMatchsticks(int matchsticks) {

printf("Matchsticks remaining: %d\n", matchsticks);

}

int userTurn(int matchsticks) {

int take;

printf("Your turn. How many matchsticks will you take (1, 2, or 3)? ");

scanf("%d", &take);

while (take < 1 || take > 3 || take > matchsticks) {

printf("Invalid input. You can take 1, 2, or 3 matchsticks, but not more than the remaining matchsticks.\n");

printf("How many matchsticks will you take (1, 2, or 3)? ");

scanf("%d", &take);

}

matchsticks -= take;

return matchsticks;

}

int computerTurn(int matchsticks) {

int take;

if (matchsticks % 4 == 0) {

take = rand() % 3 + 1;

} else {

take = matchsticks % 4;

if (take == 0) take = 3;

}

printf("Computer's turn. It takes %d matchsticks.\n", take);

matchsticks -= take;

return matchsticks;

}

int main() {

int matchsticks = 21;

srand(time(0));

printf("Welcome to the Matchstick Game!\n");

printf("There are 21 matchsticks initially.\n");

printMatchsticks(matchsticks);

while (matchsticks > 0) {

matchsticks = userTurn(matchsticks);

if (matchsticks == 0) {

printf("You took the last matchstick. You lose!\n");

break;

}

printMatchsticks(matchsticks);

matchsticks = computerTurn(matchsticks);

if (matchsticks == 0) {

printf("The computer took the last matchstick. You win!\n");

break;

}

printMatchsticks(matchsticks);

}

return 0;

}

**INPUT & OUTPUT:**

Welcome to the Matchstick Game!

There are 21 matchsticks initially.

Matchsticks remaining: 21

Your turn. How many matchsticks will you take (1, 2, or 3)? 3

Matchsticks remaining: 18

Computer's turn. It takes 2 matchsticks.

Matchsticks remaining: 16

Your turn. How many matchsticks will you take (1, 2, or 3)? 2

Matchsticks remaining: 14

Computer's turn. It takes 3 matchsticks.

Matchsticks remaining: 11

Your turn. How many matchsticks will you take (1, 2, or 3)? 3

Matchsticks remaining: 8

Computer's turn. It takes 3 matchsticks.

Matchsticks remaining: 5

Your turn. How many matchsticks will you take (1, 2, or 3)? 2

Matchsticks remaining: 3

Computer's turn. It takes 3 matchsticks.

Matchsticks remaining: 0

The computer took the last matchstick. You win!