**A Micro Project Report**

**on**

**Problem Solving using C Language**

Submitted by

#### Shaik Sameer (23471A0562)



**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 Shaik Sameer Roll No: 23471A0562, 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 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** |
|  | **C Program to Read a Number and Displaying Its Digit in Words** |
|  | **2.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** |
|  | **4.Write a program to produce the following output:**  **ABCDEFGFEDCBA**  **ABCDEF FEDCВА**  **ABCDE EDCBA**  **ABCD DCBA**  **ABC CBA**  **AB BA**  **A A** |

**Read a Number and Displaying its Digits in Words**

**AIM**:

**C Program to Read a Number and Displaying Its Digit in Word**

#include <stdio.h>

void printDigitInWords(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;

}

}

int main() {

int number, digit, temp;

printf("Enter a number: ");

scanf("%d", &number);

if (number == 0) {

printf("Zero\n");

return 0;

}

if (number < 0) {

printf("Negative ");

number = -number;

}

temp = number;

int reversed = 0;

while (temp > 0) {

reversed = (reversed \* 10) + (temp % 10);

temp /= 10;

}

while (reversed > 0) {

digit = reversed % 10;

printDigitInWords(digit);

reversed /= 10;

}

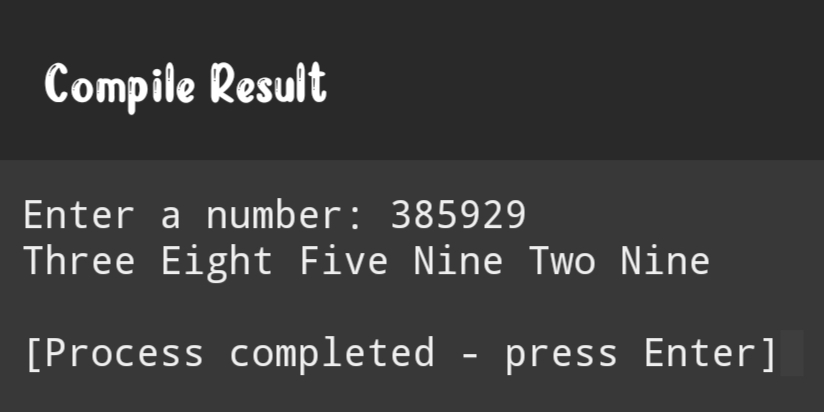
printf("\n");

return 0;

}

**Output:**

**Enter a number: 385929**

**Three Eight Five Nine Two Nine **

Area of the Triangle

**2.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**

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

typedef struct {

double a, b, c;

} Triangle;

double calculate\_area(Triangle t) {

double s = (t.a + t.b + t.c) / 2.0;

return sqrt(s \* (s - t.a) \* (s - t.b) \* (s - t.c));

}

int compare\_triangles(const void \*t1, const void \*t2) {

double area1 = calculate\_area(\*(Triangle \*)t1);

double area2 = calculate\_area(\*(Triangle \*)t2);

if (area1 < area2) return -1;

if (area1 > area2) return 1;

return 0;

}

void print\_triangle(Triangle t) {

printf("Triangle with sides: %.2f, %.2f, %.2f\n", t.a, t.b, t.c);

}

int main() {

Triangle triangles[] = {

{3, 4, 5},

{6, 8, 10},

{5, 12, 13}

};

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

qsort(triangles, n, sizeof(Triangle), compare\_triangles);

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

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

print\_triangle(triangles[i]);

}

return 0;

}

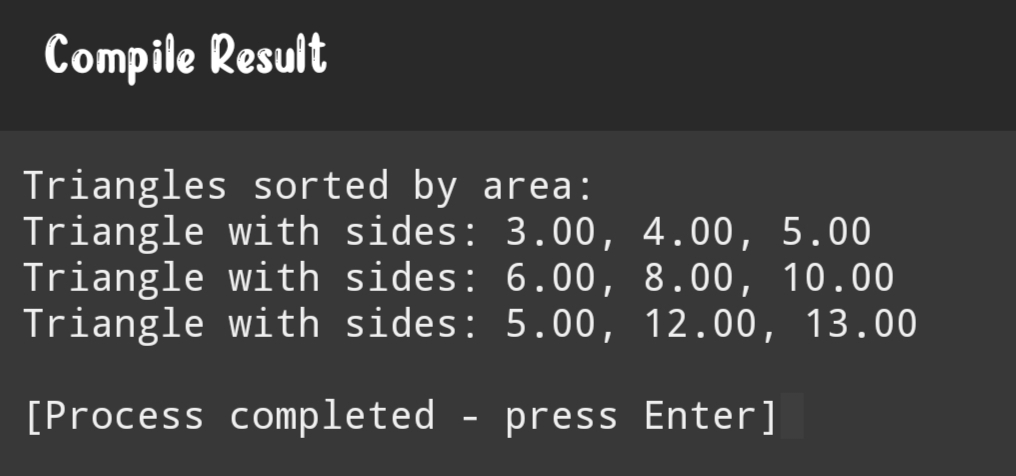
**Output:**

**Triangles sorted by area:**

**Triangle with sides: 3.00, 4.00, 5.00**

**Triangle with sides: 6.00, 8.00, 10.00**

**Triangle with sides: 5.00, 12.00, 13.00**

****

**v**

Print the pattern

**AIAA**

AIM:

**4.Write a program to produce the following output:**

**ABCDEFGFEDCBA**

**ABCDEF FEDCВА**

**ABCDE EDCBA**

**ABCD DCBA**

**ABC CBA**

**AB BA**

**A A**

#include <stdio.h>

int main() {

int i, j, spaces;

int n = 7;

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

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

printf("%c", 'A' + j);

}

spaces = 2 \* i;

for (j = 0; j < spaces; j++) {

printf(" ");

}

for (j = n - i - 1; j >= 0; j--) {

printf("%c", 'A' + j);

}

printf("\n");

}

return 0;

}

**Output:**

**ABCDEFGFEDCBA**

**ABCDEF FEDCВА**

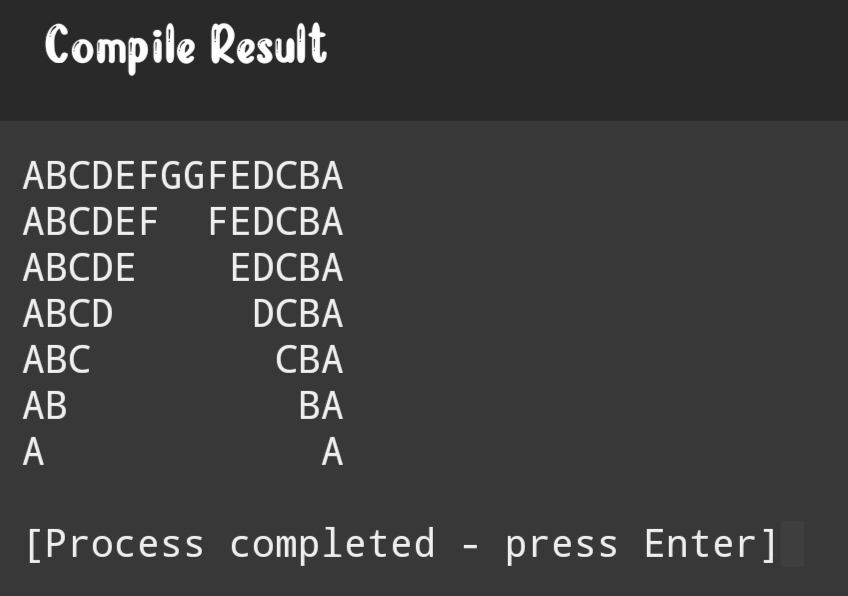
**ABCDE EDCBA**

**ABCD DCBA**

**ABC CBA**

**AB BA**

**A A**

****