Softwarica
College of IT & E-commerce

2024

Submitted to:
Shrawan Thakur

Programming

About

This project is completed as part of Task 3 for Computer Programming, covering the completion of various C programming tasks.

~ Abhishek Adhikari

Project Documentation: C Programming Exercises

1. Introduction

This document provides a comprehensive overview of various C programming exercises. Each section includes the purpose of the program, the code, and instructions for compiling and running the program.

2. Programs

2.1 Check if a Number is Even or Odd

Explanation:

- The program prompts the user to enter an integer.
- It checks if the number is even by using the modulo operator %.
- If the number is divisible by 2, it is even; otherwise, it is odd.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Even or Odd Number
    Purpose: Check whether a given number is even or odd.
    Hint: number%2 - even else odd

    Additional: The program will keep asking for input until a valid integer is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main(){
    int number, validInput;

    do
    {
        printf("Enter an integer: ");
        validInput = scanf("%d", &number);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter an integer.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if (number % 2 == 0)
        printf("\033[1;33m The number %d is Even.\033[0m\n\n", number);
    else
        printf("\033[1;33m The number %d is Odd.\033[0m\n\n", number);
    system("pause");
    return 0;
}
```

Output:

2.2 Check if a Number is Positive or Negative

Explanation:

- The program checks if the number entered is positive, negative, or zero.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Positive or Negative Number
    Purpose: Check whether a given number is positive or negative.
    Hint: number > 0; number < 0

    Additional: The program will keep asking for input until a valid integer is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main(){
    int number, validInput;

    do
    {
        printf("Enter an integer: ");
        validInput = scanf("%d", &number);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter an integer.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if (number > 0)
        printf("The number is Positive.\n");
    else if (number < 0)
        printf("The number is Negative.\n");
    else
        printf("The number is Zero.\n");
    system("pause");
    return 0;
}
```

Output:

2.3 Check if a Year is a Leap Year

Explanation:

- The program determines if a year is a leap year using conditional checks.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Leap Year Check
    Purpose: Find whether a given year is a leap year or not.

    Additional: The program will keep asking for input until a valid integer is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int year, validInput;

    do
    {
        printf("Enter a year: ");
        validInput = scanf("%d", &year);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter an integer.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
        printf("The year is a Leap Year.\n\n");
    else
        printf("The year is not a Leap Year.\n\n");
    system("pause");
    return 0;
}
```

Output:

2.4 Check Voting Eligibility Based on Age

Explanation:

- The program checks if the user's age is 18 or above to determine voting eligibility.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Voting Eligibility
    Purpose: Determine if a candidate is eligible for voting.
    Hint: age >= 18

    Additional: The program will keep asking for input until a valid integer is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int age, validInput;

    do
    {
        printf("Enter your age: ");
        validInput = scanf("%d", &age);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter an integer.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if (age >= 18)
        printf("You are eligible to vote.\n\n");
    else
        printf("You are not eligible to vote.\n\n");
    system("pause");
    return 0;
}
```

Output:

2.5 Determine Value of `n` Based on `m`

Explanation:

- The program evaluates the value of `m` and sets `n` based on the given conditions.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Height Categorization
    Purpose: Categorize a person according to their height.
    Hint: height < 150 is short, 150 <= height < 165 is average, height >= 165 is tall

    Additional: The program will keep asking for input until a valid integer is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int height, validInput;

    do
    {
        printf("Enter your height in cm: ");
        validInput = scanf("%d", &height);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter an integer.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if (height < 150)
        printf("You are categorized as Short.\n");
    else if (height >= 150 && height < 165)
        printf("You are categorized as Average.\n");
    else
        printf("You are categorized as Tall.\n");
    system("pause");
    return 0;
}
```

Output:

2.6 Categorize a Person's Height

Explanation:

- The program categorizes the height of a person into "Short", "Average", or "Tall".

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Determine Value of n
    Purpose: Display the value of n based on the value of m.
    Hint: n = 1 when m > 0, n = 0 when m = 0, n = -1 when m < 0

    Additional: The program will keep asking for input until a valid integer is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int m, n, validInput;

    do
    {
        printf("Enter an integer value for m: ");
        validInput = scanf("%d", &m);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter an integer.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if (m > 0)
        n = 1;
    else if (m == 0)
        n = 0;
    else
        n = -1;

    printf("The value of n is: %d\n", n);
    system("pause");
    return 0;
}
```

Output:

2.7 Find the Largest of Three Numbers

Explanation:

- The program determines the largest of the three input numbers.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Largest of Three Numbers
    Purpose: Find the largest of three numbers.
    Hint: Use conditional statements to compare the numbers.

    Additional: The program will keep asking for input until three valid integers are entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a, b, c, validInput;

    do
    {
        printf("Enter three integers: ");
        validInput = scanf("%d %d %d", &a, &b, &c);

        if (validInput != 3)
        {
            printf("\n\033[1;31mInvalid input. Please enter three integers.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 3);

    if (a > b && a > c)
        printf("The largest number is: %d\n\n", a);
    else if (b > a && b > c)
        printf("The largest number is: %d\n\n", b);
    else
        printf("The largest number is: %d\n\n", c);
    system("pause");
    return 0;
}
```

Output:

2.8 Calculate the Root of a Quadratic Equation

Explanation:

- The program computes the roots of a quadratic equation based on the discriminant value.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Roots of a Quadratic Equation
    Purpose: Calculate the roots of a quadratic equation.
    Formula: ax^2 + bx + c = 0

    Additional: The program will keep asking for input until valid integers are entered.
*/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
int main()
{
    int a, b, c, validInput;
    float discriminant, root1, root2;

    do
    {
        printf("Enter coefficients a, b, and c: ");
        validInput = scanf("%d %d %d", &a, &b, &c);

        if (validInput != 3)
        {
            printf("\n\033[1;31mInvalid input. Please enter three integers.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 3);

    discriminant = b * b - 4 * a * c;

    if (discriminant > 0)
    {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);
        printf("Roots are real and different: %.2f and %.2f\n\n", root1, root2);
    }
    else if (discriminant == 0)
    {
        root1 = root2 = -b / (2 * a);
        printf("Roots are real and the same: %.2f and %.2f\n\n", root1, root2);
    }
    else
    {
        float realPart = -b / (2 * a);
        float imaginaryPart = sqrt(-discriminant) / (2 * a);
        printf("Roots are complex and different: %.2f + %.2fi and %.2f - %.2fi\n\n", realPart, imaginaryPart, realPart, imaginaryPart);
    }
    system("pause");
    return 0;
}
```

Output:

2.9 Calculate Total, Percentage, and Division

Explanation:

- The program calculates total marks, percentage, and division based on user input.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Student Marks and Division
    Purpose: Read roll no, name, and marks of three subjects and calculate the total, percentage, and division.
    Hint: total = subject1 + subject2 + subject3; percentage = (total/300)*100;

    Additional: The program will keep asking for valid input until proper values are entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int rollNo, subject1, subject2, subject3, total, validInput;
    char name[50];
    float percentage;

    printf("Enter Roll No: ");
    scanf("%d", &rollNo);

    printf("Enter Name: ");
    scanf("%s", name);

    do
    {
        printf("Enter marks of three subjects: ");
        validInput = scanf("%d %d %d", &subject1, &subject2, &subject3);

        if (validInput != 3)
        {
            printf("\n\033[1;31mInvalid input. Please enter three integers.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 3);

    total = subject1 + subject2 + subject3;
    percentage = (total / 300.0) * 100;

    printf("Total: %d\n\n", total);
    printf("Percentage: %.2f%%\n\n", percentage);

    if (percentage >= 60)
        printf("Division: First\n\n");
    else if (percentage >= 50)
        printf("Division: Second\n\n");
    else if (percentage >= 40)
        printf("Division: Third\n\n");
    else
        printf("Division: Fail\n\n");
    system("pause");
    return 0;
}
```

Output:

2.10 Temperature Categorization

Explanation:

- The program evaluates the temperature and prints a suitable weather message.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Temperature Message
    Purpose: Display a suitable message according to the temperature in centigrade.
    Hint: Temp < 0 then Freezing weather; Temp 0-10 then Very Cold weather; etc.

    Additional: The program will keep asking for input until a valid integer is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int temperature, validInput;

    do
    {
        printf("Enter temperature in centigrade: ");
        validInput = scanf("%d", &temperature);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter an integer.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if (temperature < 0)
        printf("Freezing weather.\n\n");
    else if (temperature >= 0 && temperature < 10)
        printf("Very Cold weather.\n\n");
    else if (temperature >= 10 && temperature < 20)
        printf("Cold weather.\n\n");
    else if (temperature >= 20 && temperature < 30)
        printf("Normal in Temperature.\n\n");
    else if (temperature >= 30 && temperature < 40)
        printf("It's Hot.\n\n");
    else
        printf("It's Very Hot.\n\n");
    system("pause");
    return 0;
}
```

Output:

2.11 Check Triangle Type

Explanation:

- The program categorizes the triangle based on its side lengths.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Triangle Type Check
    Purpose: Determine if a triangle is equilateral, isosceles, or scalene.
    Hint: Compare the three sides of the triangle.

    Additional: The program will keep asking for input until three valid integers are entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int side1, side2, side3, validInput;

    do
    {
        printf("Enter the three sides of the triangle: ");
        validInput = scanf("%d %d %d", &side1, &side2, &side3);

        if (validInput != 3)
        {
            printf("\n\033[1;31mInvalid input. Please enter three integers.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 3);

    if (side1 == side2 && side2 == side3)
        printf("The triangle is Equilateral.\n");
    else if (side1 == side2 || side2 == side3 || side1 == side3)
        printf("The triangle is Isosceles.\n");
    else
        printf("The triangle is Scalene.\n");
    system("pause");
    return 0;
}
```

Output:

2.12 Check if a Triangle Can be Formed

Explanation:

- The program checks if the sum of the three angles equals 180 degrees to validate the triangle.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Triangle Formation Check
    Purpose: Check whether a triangle can be formed by the given angles.
    Hint: Sum of angles = 180 degrees.

    Additional: The program will keep asking for input until three valid integers are entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int angle1, angle2, angle3, validInput;

    do
    {
        printf("Enter the three angles of the triangle: ");
        validInput = scanf("%d %d %d", &angle1, &angle2, &angle3);

        if (validInput != 3)
        {
            printf("\n\033[1;31mInvalid input. Please enter three integers.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 3);

    if (angle1 + angle2 + angle3 == 180)
        printf("The angles can form a triangle.\n\n");
    else
        printf("The angles cannot form a triangle.\n\n");
    system("pause");
    return 0;
}
```

Output:

2.13 Check if a Character is an Alphabet, Digit, or Special Character

Explanation:

- The program uses standard library functions to classify the character.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Character Type Check
    Purpose: Check whether a character is an alphabet, digit, or special character.
    Hint: Use ASCII values to determine character type.

    Additional: The program will keep asking for input until a valid character is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    char ch;
    int validInput;

    do
    {
        printf("Enter a character: ");
        validInput = scanf(" %c", &ch);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter a single character.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))
        printf("The character is an Alphabet.\n\n");
    else if (ch >= '0' && ch <= '9')
        printf("The character is a Digit.\n\n");
    else
        printf("The character is a Special Character.\n\n");
    system("pause");
    return 0;
}
```

Output:

2.14 Check if an Alphabet is a Vowel or Consonant

Explanation:

- The program determines if the alphabet is a vowel or consonant.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Vowel or Consonant
    Purpose: Check whether an alphabet is a vowel or consonant.
    Hint: Vowels are 'a', 'e', 'i', 'o', 'u'.

    Additional: The program will keep asking for input until a valid alphabet is entered.
*/

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
int main()
{
    char ch;
    int validInput;

    do
    {
        printf("Enter an alphabet: ");
        validInput = scanf(" %c", &ch);

        if (validInput != 1 || !((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z')))
        {
            printf("\n\033[1;31mInvalid input. Please enter a single alphabet.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1 || !((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z')));

    ch = tolower(ch);
    if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
        printf("The alphabet is a Vowel.\n");
    else
        printf("The alphabet is a Consonant.\n");
    system("pause");
    return 0;
}
```

Output:

2.15 Check Character Type (Uppercase, Lowercase, Digit, Special Symbol)

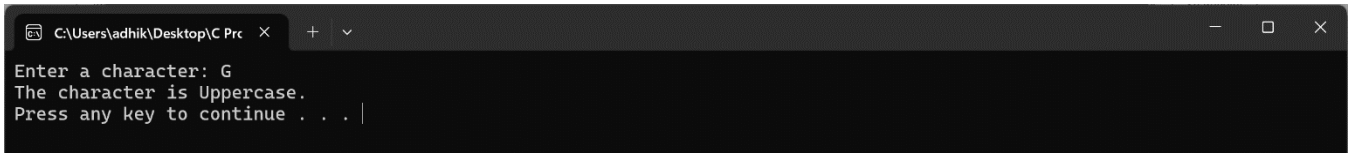
Explanation:

- The program classifies the character based on its type.

Code:

```c
/*
    Author: Abhishek Adhikari
    Date: 10th August 2024
    Topic: Character Type Check
    Purpose: Check if a character is uppercase, lowercase, digit, or special symbol.
    Hint: Use ASCII values to determine the type.

    Additional: The program will keep asking for input until a valid character is entered.
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    char ch;
    int validInput;

    do
    {
        printf("Enter a character: ");
        validInput = scanf(" %c", &ch);

        if (validInput != 1)
        {
            printf("\n\033[1;31mInvalid input. Please enter a single character.\033[0m\n\n");
            while (getchar() != '\n')
                ;
        }

    } while (validInput != 1);

    if (ch >= 'A' && ch <= 'Z')
        printf("The character is Uppercase.\n");
    else if (ch >= 'a' && ch <= 'z')
        printf("The character is Lowercase.\n");
    else if (ch >= '0' && ch <= '9')
        printf("The character is a Digit.\n");
    else
        printf("The character is a Special Symbol.\n");
    system("pause");
    return 0;
}
```

Output:



3. Conclusion

This document provides a detailed overview of the various C programming exercises, including explanations, code, and instructions for each program. Please refer to the screenshots provided to see the programs in action.

4. References

C Programming Language - K&R

- *Authors:* Brian W. Kernighan and Dennis M. Ritchie
- *Description:* A seminal book on C programming, providing foundational knowledge and in-depth explanations of C syntax and features.
- *Link:* The C Programming Language

GNU C Library Documentation

- *Description:* Official documentation for the GNU C Library, providing detailed information on C standard library functions and their usage.
- *Link:* GNU C Library Documentation

GeeksforGeeks: C Programming Language

- *Description:* A comprehensive collection of articles, tutorials, and coding examples for learning C programming concepts and solving problems.
- *Link:* GeeksforGeeks C Programming