# CSCI-12042-Structured programming II

# Course Work

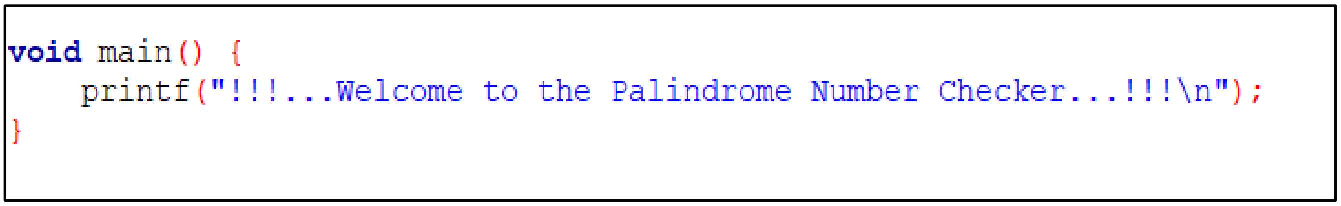
### Utility application title: - Palindrome Number Checker

## Introduction

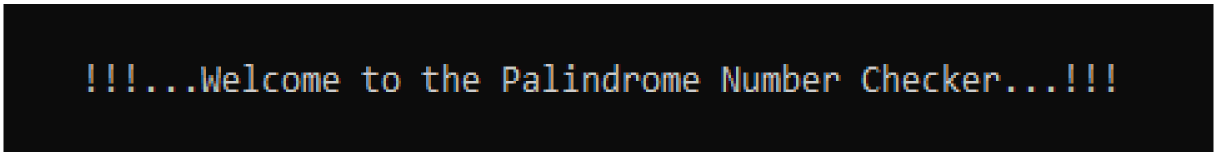
The Palindrome Number Checker in C is a program that verifies if a given integer reads the same backward as forward (palindrome). It greets the user, prompts for a number, and checks if it is a valid integer. The core function is Palindrome reverses the digits and compares them with the original number. The program then informs the user if the number is a palindrome or not.

## Implementation of the program

1) Display an introduction message about the program.



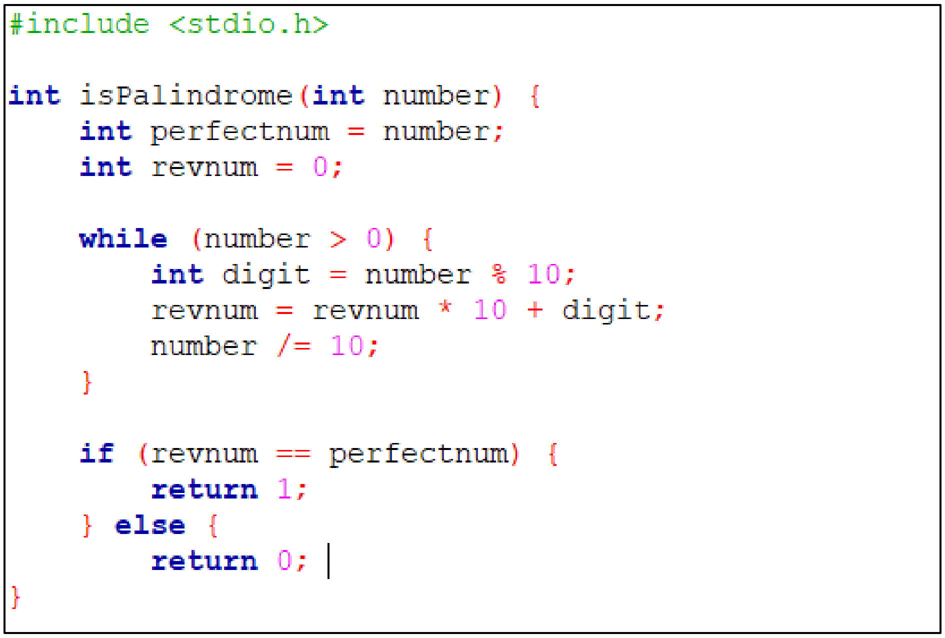
```c
void main() {
    printf("!!!...Welcome to the Palindrome Number Checker...!!!\n");
}
```



```
!!!...Welcome to the Palindrome Number Checker...!!!
```

2) Function to check if a number is a palindrome

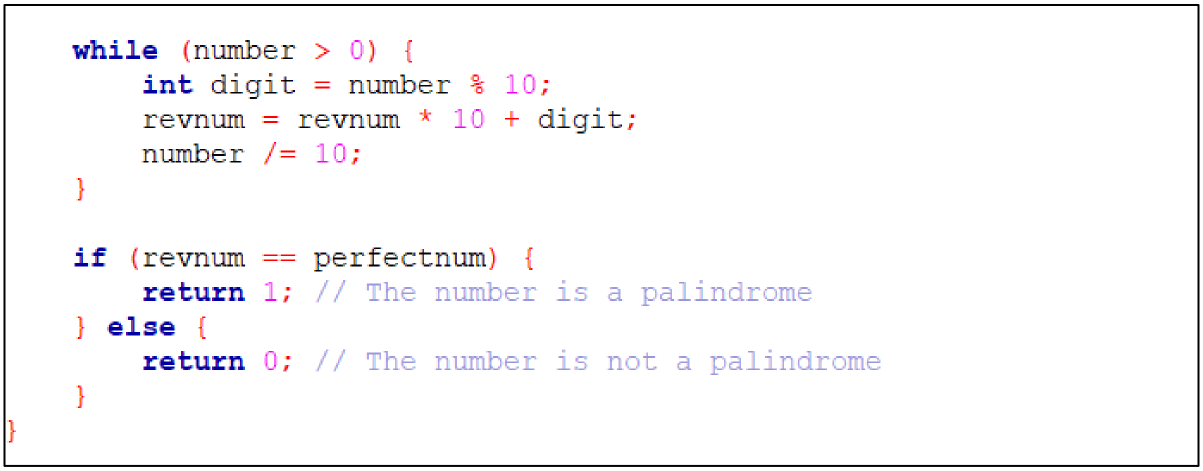
This function takes an integer ‘**number**’ as input, and within the function, it reverses the digits of the ‘**number**’ and stores the ‘**revnum**’ in the variable ‘**revnum**’. After reversing the digits, it compares ‘**revnum**’ with the original ‘**number**’. If both are equal, it means the number is a palindrome, and the function returns 1. Otherwise, it returns 0, indicating that the number is not a palindrome.



```c
#include <stdio.h>

int isPalindrome(int number) {
    int perfectnum = number;
    int revnum = 0;

    while (number > 0) {
        int digit = number % 10;
        revnum = revnum * 10 + digit;
        number /= 10;
    }

    if (revnum == perfectnum) {
        return 1;
    } else {
        return 0; |
}
```

3) Reverse number process and compare it with perfect number

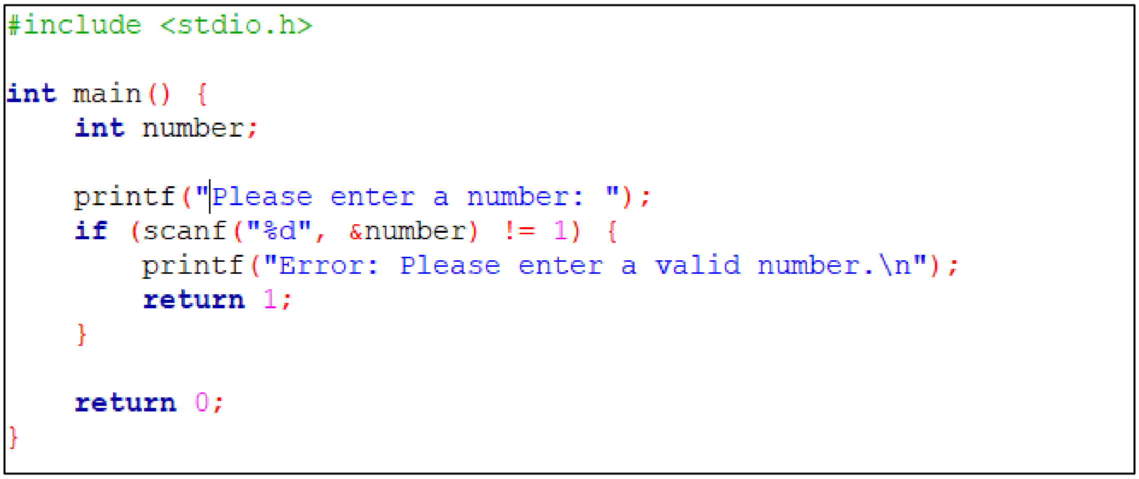
- Initialize ‘**perfectnum** ‘to the input number and ‘**revnum**’  to 0.
- Using a loop, extract the last digit of the ‘**number**’, add it to the rightmost position of **‘revnum’**, and remove that last digit from ‘**number**’.
- Repeat step 2 until all digits are processed, resulting in ‘**revnum**’ having the digits of the input number in reverse order.
- Compare ‘**revnum**’ with the ‘**perfectnum**’ . If they are equal, the input number is a palindrome. Otherwise, it is not.



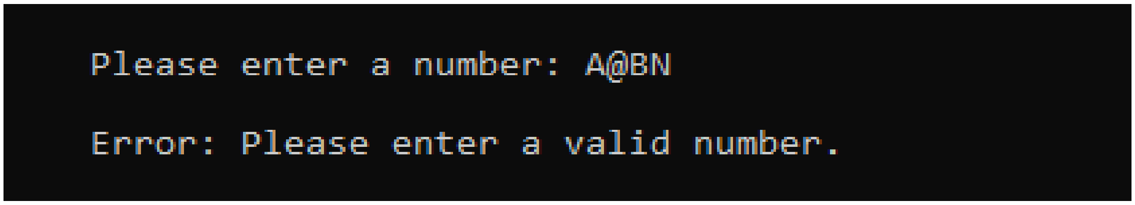
```c
    while (number > 0) {
        int digit = number % 10;
        revnum = revnum * 10 + digit;
        number /= 10;
    }

    if (revnum == perfectnum) {
        return 1; // The number is a palindrome
    } else {
        return 0; // The number is not a palindrome
    }
}
```

4) Prompt for input and check for errors

In the provided code, the error option checks if the user entered a valid integer when prompted for input. If the input is not a valid integer or the user cancels the input, an error message is displayed, and the program exits with a non-zero status to indicate that an error occurred. This ensures that the program handles invalid input gracefully and prevents unexpected behavior.

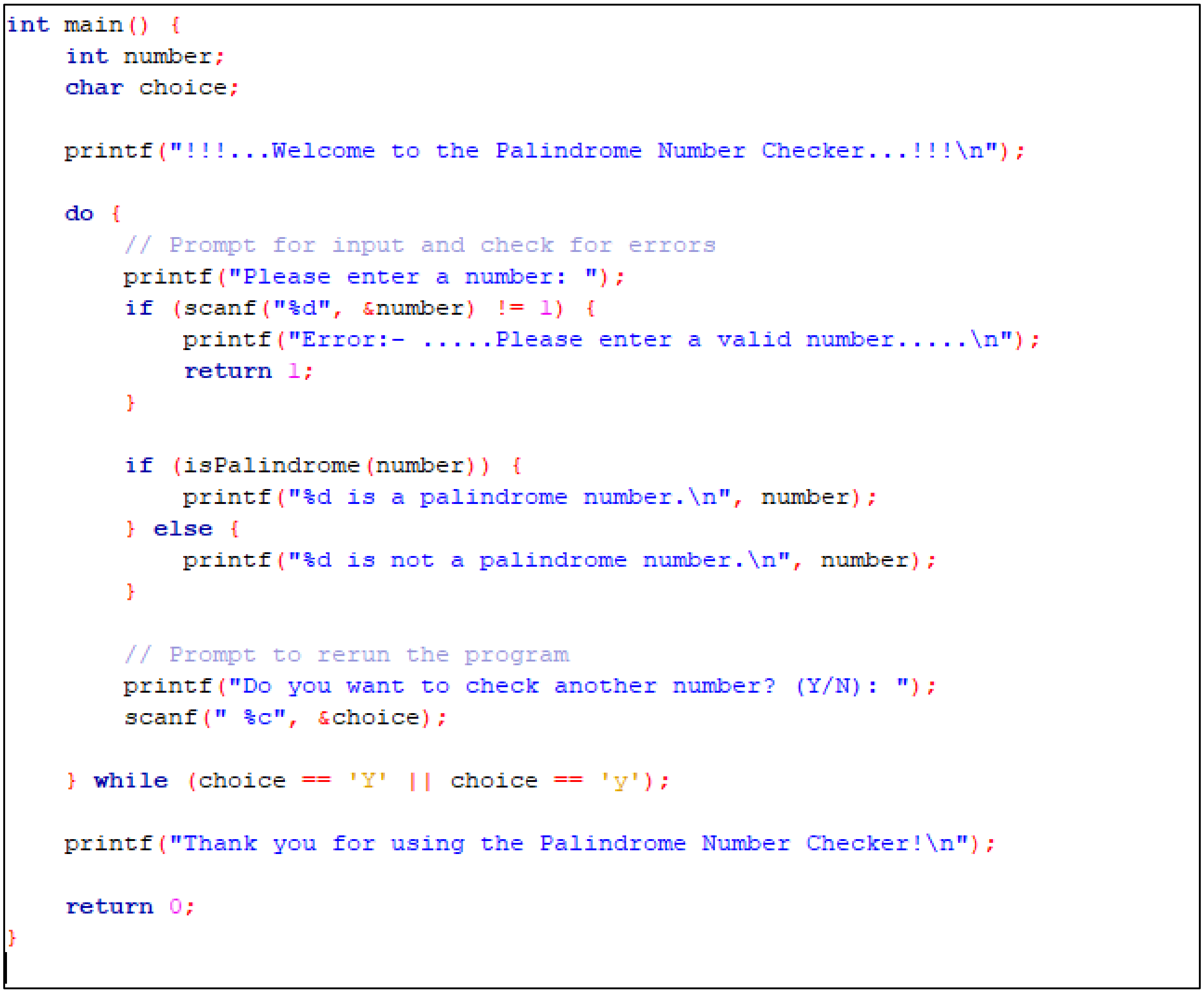


```c
#include <stdio.h>

int main() {
    int number;

    printf("Please enter a number: ");
    if (scanf("%d", &number) != 1) {
        printf("Error: Please enter a valid number.\n");
        return 1;
    }

    return 0;
}
```

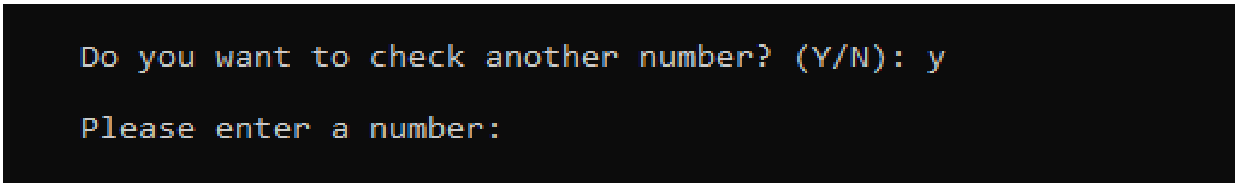


```
Please enter a number: A@BN

Error: Please enter a valid number.
```

5) If you wish to rerun the program.

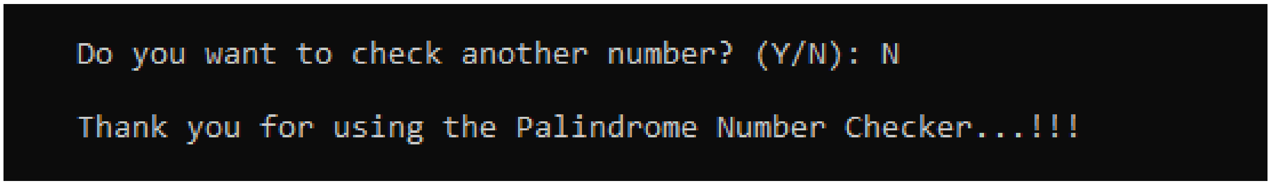


```
int main() {
    int number;
    char choice;

    printf("!!!...Welcome to the Palindrome Number Checker...!!!\n");

    do {
        // Prompt for input and check for errors
        printf("Please enter a number: ");
        if (scanf("%d", &number) != 1) {
            printf("Error:- .....Please enter a valid number.....\n");
            return 1;
        }

        if (isPalindrome(number)) {
            printf("%d is a palindrome number.\n", number);
        } else {
            printf("%d is not a palindrome number.\n", number);
        }

        // Prompt to rerun the program
        printf("Do you want to check another number? (Y/N): ");
        scanf(" %c", &choice);

    } while (choice == 'Y' || choice == 'y');

    printf("Thank you for using the Palindrome Number Checker!\n");

    return 0;
}
```

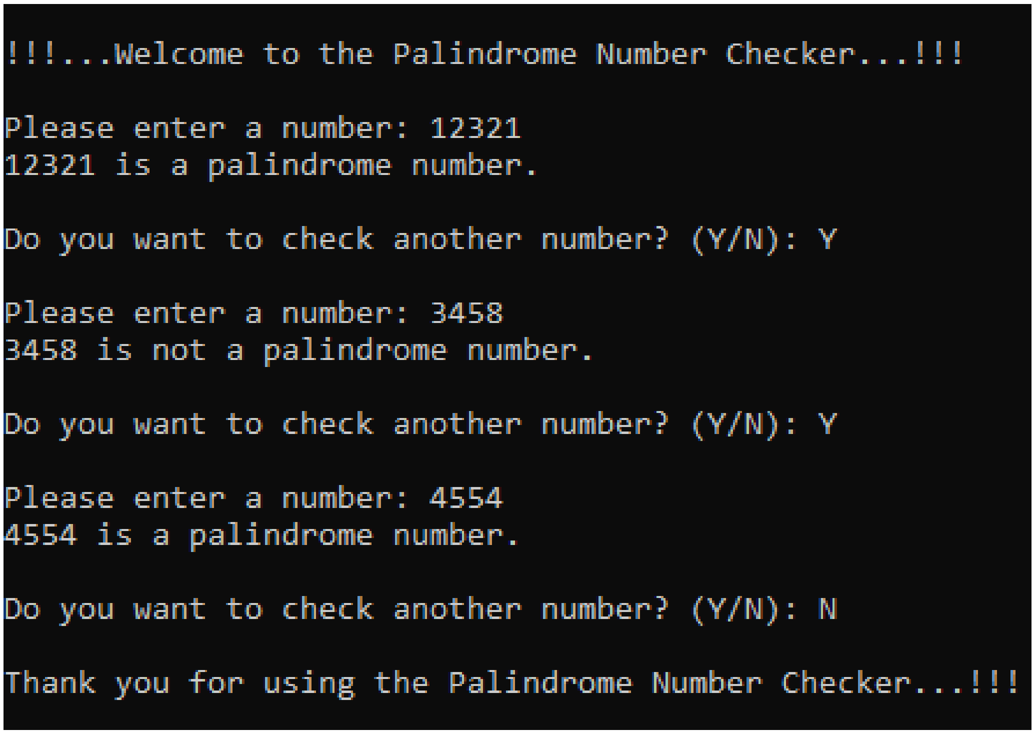


```
Do you want to check another number? (Y/N): y

Please enter a number:
```

6) When you enter “N” exit the programme.



```
Do you want to check another number? (Y/N): N

Thank you for using the Palindrome Number Checker...!!!
```



```
!!!...Welcome to the Palindrome Number Checker...!!!

Please enter a number: 12321
12321 is a palindrome number.

Do you want to check another number? (Y/N): Y

Please enter a number: 3458
3458 is not a palindrome number.

Do you want to check another number? (Y/N): Y

Please enter a number: 4554
4554 is a palindrome number.

Do you want to check another number? (Y/N): N

Thank you for using the Palindrome Number Checker...!!!
```

## Benefits

The above C code checks if a given number is a palindrome or not. It can be used to verify if a number remains the same when its digits are reversed. The code is helpful for,

- educational purposes
- algorithm development
- data validation
- code testing
- interactive user experiences

It serves as a practical example to understand functions, loops, conditionals, and user input handling in C programming.