

UE20CS152 Week 3

Anirudh Rowjee PES2UG20CS050

Task 1

Write a function to reverse a given number and check whether the number is a palindrome or not.

Code

```
#include <stdio.h>
#include <stdbool.h>

/*
 * Program to use Functions to
 * 1. reverse a number
 * 2. check if a given number is a palindrome or not
 */

int reverse_number(int input)
{
    // function to reverse a number

    int sum = 0;
    int copy = input;

    while (input != 0)
    {
        int rem = input % 10;
        sum = sum * 10 + rem;
        input -= rem;
        input /= 10;
    }
    // printf("The number is %d\n", copy);
    // printf("The Reversed version of the number is %d\n", sum);
    return sum;
}

bool is_paldinrome_number(int input)
{
    // function to check if the number is the reverse
    // of itself i.e. is a palindrome
    return (reverse_number(input) == input);
}

int main()
{
```

```

int num = 0;
printf("Enter the number to check> ");
scanf("%d", &num);
printf(
    "The number %d %s a palindrome number.\n",
    num,
    is_paldinrome_number(num) ? "is" : "is not"
);
return 0;
}

```

Console

```

nidavellir :: sem2/UE20CS152/week2 » ./1
Enter the number to check> 121
The number 121 is a palindrome number.
nidavellir :: sem2/UE20CS152/week2 » ./1
Enter the number to check> 123
The number 123 is not a palindrome number.
nidavellir :: sem2/UE20CS152/week2 » ./1
Enter the number to check> 1
The number 1 is a palindrome number.

```

Task 2

Write a Program to compute the GCD Of three numbers using functions

Code

```

#include <stdio.h>

int euclidean_gcd(int a, int b)
{
    // compute the GCD of two numbers using
    // Euclid's Algorithm
    if (a == b)
    {
        return a;
    }
    else if (a > b)
    {
        return euclidean_gcd(a-b, b);
    }
    else
    {
        return euclidean_gcd(b-a, a);
    }
}

```

```

int three_number_gcd(int a, int b, int c)
{
    return euclidean_gcd(a, euclidean_gcd(b, c));
}

int main()
{
    int first, second, third;
    printf("Enter the three numbers separated by a comma >");
    scanf("%d, %d, %d", &first, &second, &third);
    printf(
        "the GCD of %d and %d and %d is %d\n",
        first, second, third,
        three_number_gcd(first, second, third)
    );
    return 0;
}

```

Console

```

nidavellir :: sem2/UE20CS152/week2 » ./2
Enter the three numbers separated by a comma >10, 20, 30
the GCD of 10 and 20 and 30 is 10
nidavellir :: sem2/UE20CS152/week2 » ./2
Enter the three numbers separated by a comma >48, 102, 66
the GCD of 48 and 102 and 66 is 6

```

Task 3

Write a program in C to check whether a given number is a perfect number and an armstrong number.

Code

```

#include <stdio.h>
#include <stdbool.h>
#include <math.h>

bool is_perfect(int number)
{
    // function to check if a number is perfect
    int div_sum = 0;
    for (int i = 1; i <= number / 2; i++)
    {
        if (number % i == 0)
        {
            div_sum += i;
        }
    }
}

```

```

    }
}
return (div_sum == number);
}

bool is_armstrong(int number)
{
    // function to check if a number is an armstrong number

    int digit_count = 0;
    int copy = number;
    int power_sum = 0;

    // find the number of digits the number has
    while (copy != 0)
    {
        int remainder = copy % 10;
        digit_count++;
        copy -= remainder;
        copy /= 10;
    }
    copy = number;
    // sum each digit raised to the
    // power of the number of digits
    while (copy != 0)
    {
        int remainder = copy % 10;
        power_sum += pow(remainder, digit_count);
        copy -= remainder;
        copy /= 10;
    }

    return (power_sum == number);
}

char* format_boolean_output(bool output)
{
    return output ? "is" : "is not";
}

int main()
{
    int test;
    printf("Enter the number you want to test > ");
    scanf("%d", &test);
    printf(
        "%d %s an Armstrong number.\n",
        test,
        format_boolean_output(is_armstrong(test))
    );
    printf(
        "%d %s a Perfect number.\n",
        test,
        format_boolean_output(is_perfect(test))
    );
};

```

```
    return 0;
}
```

Console

```
nidavellir :: sem2/UE20CS152/week2 » ./3
Enter the number you want to test > 135
135 is not an Armstrong number.
135 is not a Perfect number.
nidavellir :: sem2/UE20CS152/week2 » ./3
Enter the number you want to test > 6
6 is an Armstrong number.
6 is a Perfect number.
```

Task 4

Write a program to check whether a number is a prime number or not using functions.

Code

```
#include <stdio.h>
#include <stdbool.h>

bool is_prime(int number)
{
    // function to check if a number is prime using exhaustive enumeration
    bool flag = true;
    bool is_num_two = (number == 2);
    if (is_num_two)
    {
        return true;
    }
    for (int i = 2; i < number; i++)
    {
        if (number % i == 0)
            return false;
    }
    return true;
}

int main()
{
    int test;
    printf("Enter the number to test > ");
    scanf("%d", &test);
    printf("%d %s a prime number\n", test, is_prime(test) ? "is" : "is not");
    return 0;
}
```

```
}
```

Console

```
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 1
1 is a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 2
2 is a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 3
3 is a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 4
4 is not a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 5
5 is a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 6
6 is not a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 7
7 is a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 8
8 is not a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 9
9 is not a prime number
nidavellir :: sem2/UE20CS152/week2 » ./4
Enter the number to test > 10
10 is not a prime number
```

Task 5

Write a program to convert a decimal number into octal using functions

Code

```
#include <stdio.h>
#include <math.h>

int decimal_to_octal(int input)
{
    // function to convert an integer into
```

```

// its octal representation
int answer = 0;
int counter = 0;

while (input != 0)
{
    int remainder = input % 8;
    answer += pow(10, counter) * remainder;
    counter++;
    input -= remainder;
    input /= 8;
}

return answer;
}

int main()
{
    int input;
    printf("Enter the decimal number you wish to convert to octal > ");
    scanf("%d", &input);
    printf(
        "The number %d in decimal is %d in octal.\n",
        input,
        decimal_to_octal(input)
    );
    return 0;
}

```

Console

```

nidavellir :: sem2/UE20CS152/week2 » ./5
Enter the decimal number you wish to convert to octal > 7
The number 7 in decimal is 7 in octal.
nidavellir :: sem2/UE20CS152/week2 » ./5
Enter the decimal number you wish to convert to octal > 10
The number 10 in decimal is 12 in octal.
nidavellir :: sem2/UE20CS152/week2 » ./5
Enter the decimal number you wish to convert to octal > 100
The number 100 in decimal is 144 in octal.

```

Task 6

Write a program to find the sum of the series $n!/n$ using a function.

Code

```

#include <stdio.h>

int factorial(int n)
{
    if (n <= 1)
    {
        return 1;
    }
    else
    {
        return n * factorial(n-1);
    }
}

int main()
{
    // program to find the sum of the (n-1)! from one to n
    int sum = 0;
    int limit;

    printf("Enter the limit of the sum > ");
    scanf("%d", &limit);

    for (int i = 1; i <= limit; i++)
    {
        sum += factorial(i - 1);
    }

    printf("The answer is %d\n", sum);

    return 0;
}

```

Console

```

nidavellir :: sem2/UE20CS152/week2 » ./6
Enter the limit of the sum > 1
The answer is 1
nidavellir :: sem2/UE20CS152/week2 » ./6
Enter the limit of the sum > 2
The answer is 2
nidavellir :: sem2/UE20CS152/week2 » ./6
Enter the limit of the sum > 3
The answer is 4
nidavellir :: sem2/UE20CS152/week2 » ./6
Enter the limit of the sum > 4
The answer is 10
nidavellir :: sem2/UE20CS152/week2 » ./6
Enter the limit of the sum > 5
The answer is 34

```

Practice Problem 1

Write a program to display the fibonacci series within a given range.

Code

```
#include <stdio.h>

int nth_fib(int count)
{
    if (count <= 2)
    {
        return 1;
    }
    else
    {
        return nth_fib(count-1) + nth_fib(count-2);
    }
}

int main()
{
    int target;

    printf("Enter the number limit > ");
    scanf("%d", &target);

    for (int i = 1; i <= target; i++)
    {
        printf("%d ", nth_fib(i));
    }
    printf("\n");

    return 0;
}
```

Output

```
nidavellir :: sem2/UE20CS152/week2 » ./pp1
Enter the number limit > 5
1 1 2 3 5
nidavellir :: sem2/UE20CS152/week2 » ./pp1
Enter the number limit > 10
1 1 2 3 5 8 13 21 34 55
```

Practice Problem 2

Write a program to check if the angles of a triangle are valid or not.

Code

```
#include <stdio.h>
#include <stdbool.h>

bool is_valid_triangle(float a, float b, float c)
{
    return (a + b + c == 180.0);
}

int main()
{
    float a, b, c;
    printf("Enter the three angles separated by commas and spaces > ");
    scanf("%f, %f, %f", &a, &b, &c);
    printf("%s\n", is_valid_triangle(a, b, c) ? "VALID" : "INVALID");
    return 0;
}
```

Output

```
nidavellir :: sem2/UE20CS152/week2 » ./pp2
Enter the three angles separated by commas and spaces > 60, 60, 60
VALID
nidavellir :: sem2/UE20CS152/week2 » ./pp2
Enter the three angles separated by commas and spaces > 60, 80, 10
INVALID
```