Computational Thinking Part 2

CLARUSWAY©
WAY TO REINVENT YOURSELF

Remember the path

- Abstraction
- Decomposition
- Pattern Recognition
- Algorithms

What is Computational Thinking?

Computational thinking is:

- Organising data/information logically (Abstraction)
- Breaking problems into parts (Decomposition)
- Interpreting patterns and models (Pattern Recognition)
- Designing and implementing algorithms (Algorithms)

Were you able to finish pre-class work for Computational Thinking?

Students choose an option

REINVENT YOURSELF

Pear Deck Interactive Slide

Do not remove this bar

Computational Thinking in Practice

How would you analyse and solve an issue such as below :

An application needs a piece of code that takes a number as input from the user and displays an output if it's in between 10 … 13

Abstraction, Decomposition, Pattern Recognition, Implementation of an algorithm possible?

Students, write your response!

Case Study : 1

Perform the actions below for the request below :
Request : *An application needs a piece of code that takes a number as input from the user and displays an output if it's in between 10 ... 13*

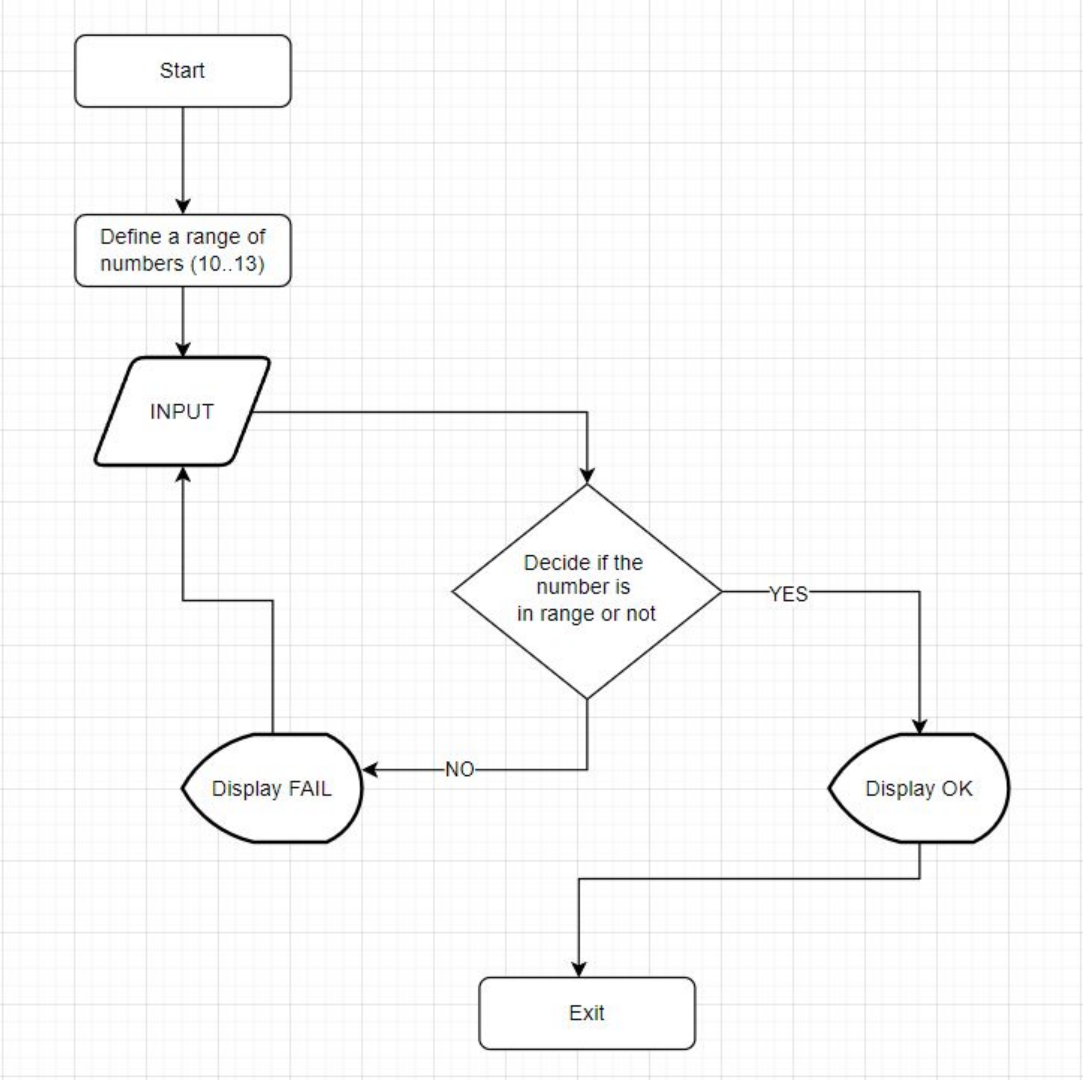
- Write the pseudo code.
- Sketch of the algorithm.
- Write the actual code that runs considering the requirements

Case Study 1 : Pseudocode

- Define the accepted range in start and end variables.
- Take the input and store it in a variable.
- Compare the input variable with the requested range.
- Display the necessary information to the user whether the input value is in range or not

Case Study 1

Algorithm


Start
Define a range of numbers (10..13)
INPUT
Decide if the number is in range or not
YES
NO
Display FAIL
Display OK
Exit

Case Study 1 - The code in c#

```
int start = 10;
int end = 13;
Console.WriteLine("Please input a number that's in range 10..13 : ");
Int32 InputValue = Convert.ToInt32(Console.ReadLine());
if (InputValue >= start &&
        InputValue <= end)
{
        Console.WriteLine("OK");
}
else
{
        Console.WriteLine("FAIL");
}
```

Case Study 1 - The code in Python

```
start = 10
end = 13

try:
   user_input = int(input('Please enter a number between {} and {} : '.format(start, end)))
except ValueError:
    print("Invalid entry detected ...")
    exit()

if user_input in range(start, end):
   print("Yes the number entered: {}".format(user_input))
else:
    print("No the number entered is out of range: {}".format(user_input))
```

Case Study 1 : Summary

What we learned from Case Study 1 :

- Not every request is applicable to Abstraction, Decomposition, Pattern Recognition
- If statement is a general technical keyword used in Pseudocode to actual code.
- How to get user inputs in a console application.
- How to do branching after decisions are made.

Case Study : 2

Perform the actions below for the request below :

Request : *Extend the capabilities of the application completed in Case Study 1 and display a warning if the input value is a negative number while keeping the code that checks the range of 10 ... 13*

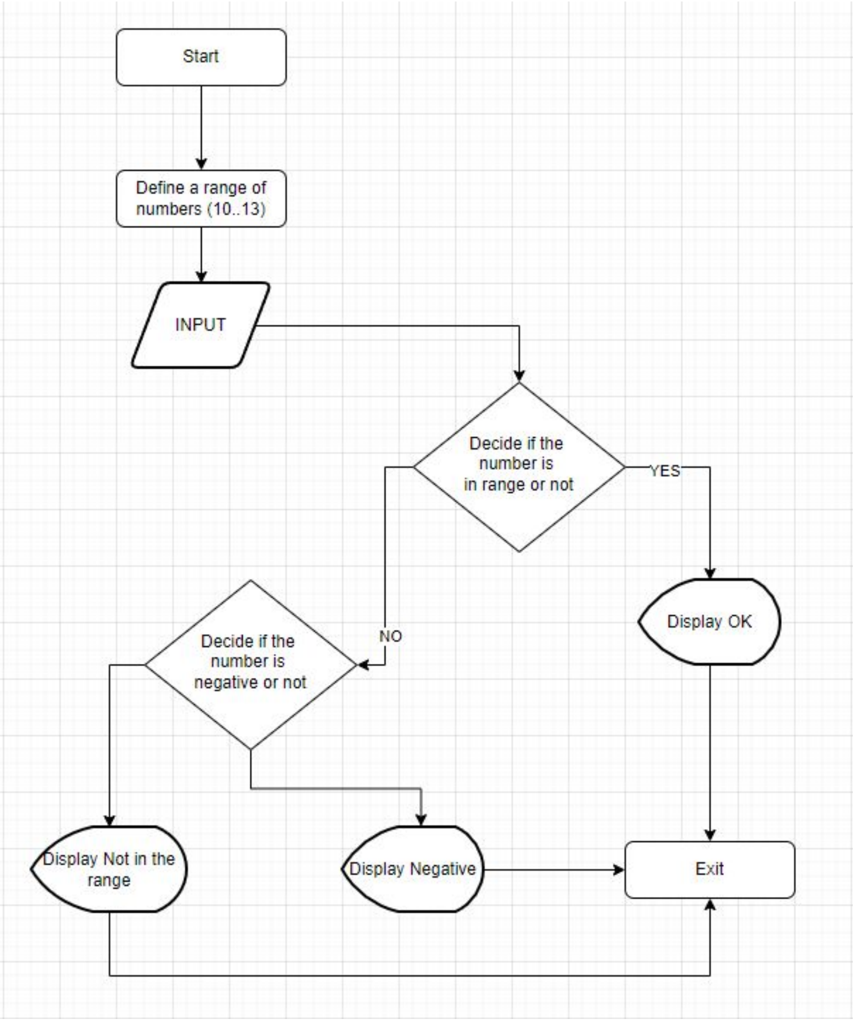
- Write the pseudo code.
- Sketch of the algorithm.
- Write the actual code that runs considering the requirements
- **What types of control logic could be added?**

Case Study 2 : Pseudocode

- Define the range of numbers.
- Take the input and store it in a variable.
- Compare the input variable with the predefined range.
- Display the necessary information to the user whether the input value is in range or not
- Display a warning message to the user if the input value is a negative number.
- Exit the application.

Case Study 2

Algorithm


Start
Define a range of numbers (10..13)
INPUT
Decide if the number is in range or not
YES
NO
Display OK
Decide if the number is negative or not
Display Not in the range
Display Negative
Exit

Case Study 2 - The code in c#

```
List<int> Range = new List<int> { 10, 11, 12, 13 };

Console.WriteLine("Please input a number that's in range 10..13 : ");
var Input = Console.ReadLine();

if (Convert.ToInt32(Input) < 0)
{
        Console.WriteLine("The input value is in NEGATIVE RANGE!!!");
        if (Convert.ToInt32(Input) < -10)
        {
                Console.WriteLine("The number is negative, and even less than -10!!");
        }
}
else
{
        Console.WriteLine("NO the input is NOT negative!!!");
}

if (Range.Contains(Convert.ToInt32(Input)))
{
        Console.WriteLine("YES the input is in defined range!!!");
}
else
{
        Console.WriteLine("NO the input is not in the defined range!!!");
}
Console.ReadKey();
```

Case Study 2 - The python code

```
start = 10
end = 13

try:
   user_input = int(input('Please enter a number between {} and {} : '.format(start, end)))
except ValueError:
    print("Invalid entry detected ...")
    exit()

if user_input < 0:
   if user_input < -10:
       print("The entered value is even less than -10")
       exit()
   print ("The value entered is less than 0...")
   exit()

if user_input in range(start, end):
   print("Yes the number entered is in range: {}".format(user_input))
else:
    print("No the number entered is out of range: {}".format(user_input))
```

Case Study 2 : Summary

What we learned from Case Study 2 :

- IF statements may contain inner if statements, where they can include inner if statements as well, there is no limit, but leads to spaghetti code, we must be aware of this fact.

Case Study : 3

Perform the actions below for the request below :
Request : *Extend the capabilities of the application completed in Case Study 1 and check the input value if it fits in ranges [10..13] [18..22] [25..44]*

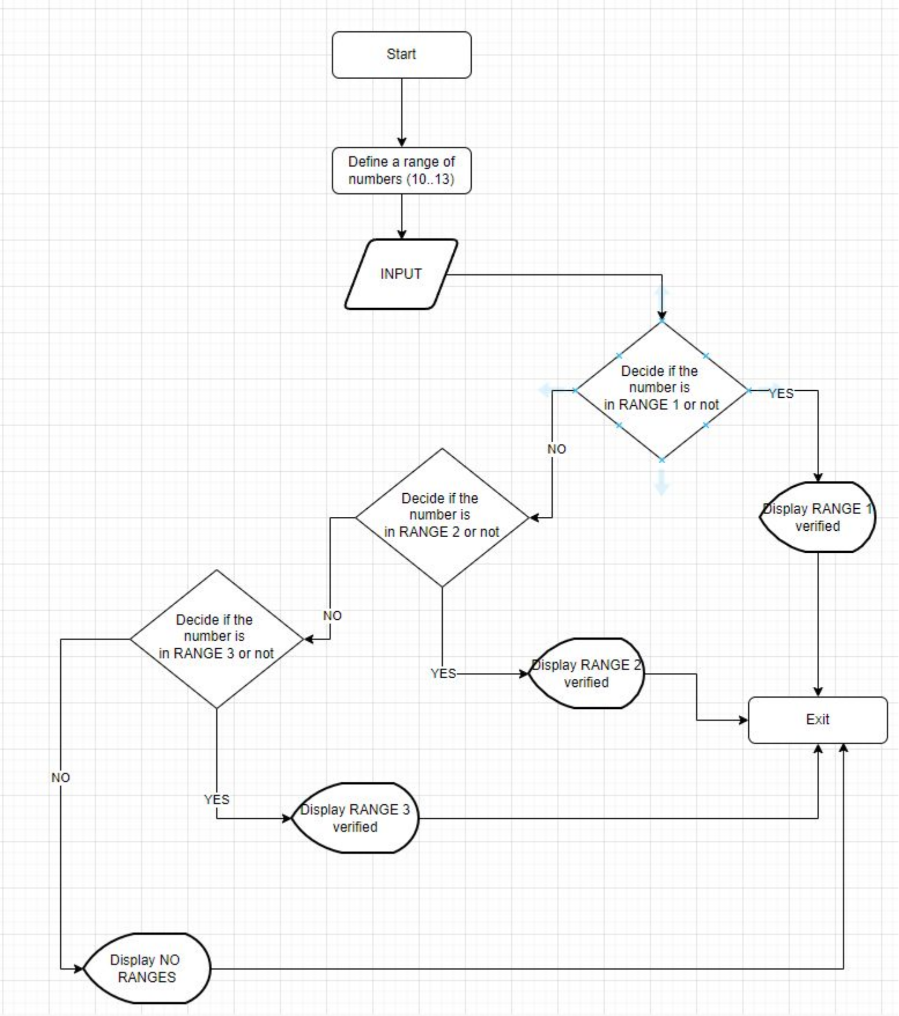
- Write the pseudo code.
- Sketch of the algorithm.
- Write the actual code that runs considering the requirements

Case Study 3 : Pseudocode

- Define the range of numbers.
- Take the input and store it in a variable.
- Compare the input value with the predefined range1.
- Compare the input value with the predefined range2.
- Compare the input value with the predefined range3.
- Display the corresponding messages for each ranges.
- Exit the application.

Case Study 3

Algorithm



Start
Define a range of numbers (10..13)
INPUT
Decide if the number is in RANGE 1 or not
YES
NO
Display RANGE 1 verified
Decide if the number is in RANGE 2 or not
NO
YES
Display RANGE 2 verified
Decide if the number is in RANGE 3 or not
Exit
NO
YES
Display RANGE 3 verified
Display NO RANGES

CLARUSWAY©
WAY TO REINVENT YOURSELF

Case Study 3 - The code in c#

```
List<int> Range1 = new List<int> { 10, 11, 12, 13, 50 };
List<int> Range2 = new List<int> { 18, 19, 20, 21, 22, 50 };
List<int> Range3 = new List<int> { 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37 ,38 ,39, 40, 41, 42, 43, 44, 45, 50};


Console.WriteLine("Please input a number that's in ranges [10..13], [18..22], [25..45]");
var Input = Console.ReadLine();
if (Range1.Contains(Convert.ToInt32(Input)))
{
    Console.WriteLine("YES the input is in RANGE 1!!!");
}
else if (Range2.Contains(Convert.ToInt32(Input)))
{
    Console.WriteLine("YES the input is in RANGE 2!!!");
}
else if (Range3.Contains(Convert.ToInt32(Input)))
{
    Console.WriteLine("YES the input is in RANGE 3!!!");
}
else
{
        Console.WriteLine("NO the input is NOT in defined in any of ranges!!!");
}
Console.ReadKey();
```

Case Study 3 - The python code

```
user_input = int(input('Please enter a number in one of ranges:10..13 - 18..22 - 25..44 : '))
result = False
if user_input in range(10, 13):
   result = True
if(user_input in range(18,22)):
   result = True
if (user_input in range(25,44)):
   result = True

if result:
   print("The entered value is in range")
else:
   print("No the entered value is not in any of ranges : {}".format(user_input))
```

Case Study 3 : Summary

What we learned from Case Study 3 :

- IF statements can break down to branches by using IF ELSE statements however this would not entirely save the application from developing in direction of spaghetti code. By means of readability of the code there are better alternatives.

Case Study : 4

Perform the actions below for the request below :
Request : *Extend the capabilities of the application completed in Case Study 3 and check the input value if it fits in ranges [10..13] [18..22] [25..44] until user presses the key X when asked.*

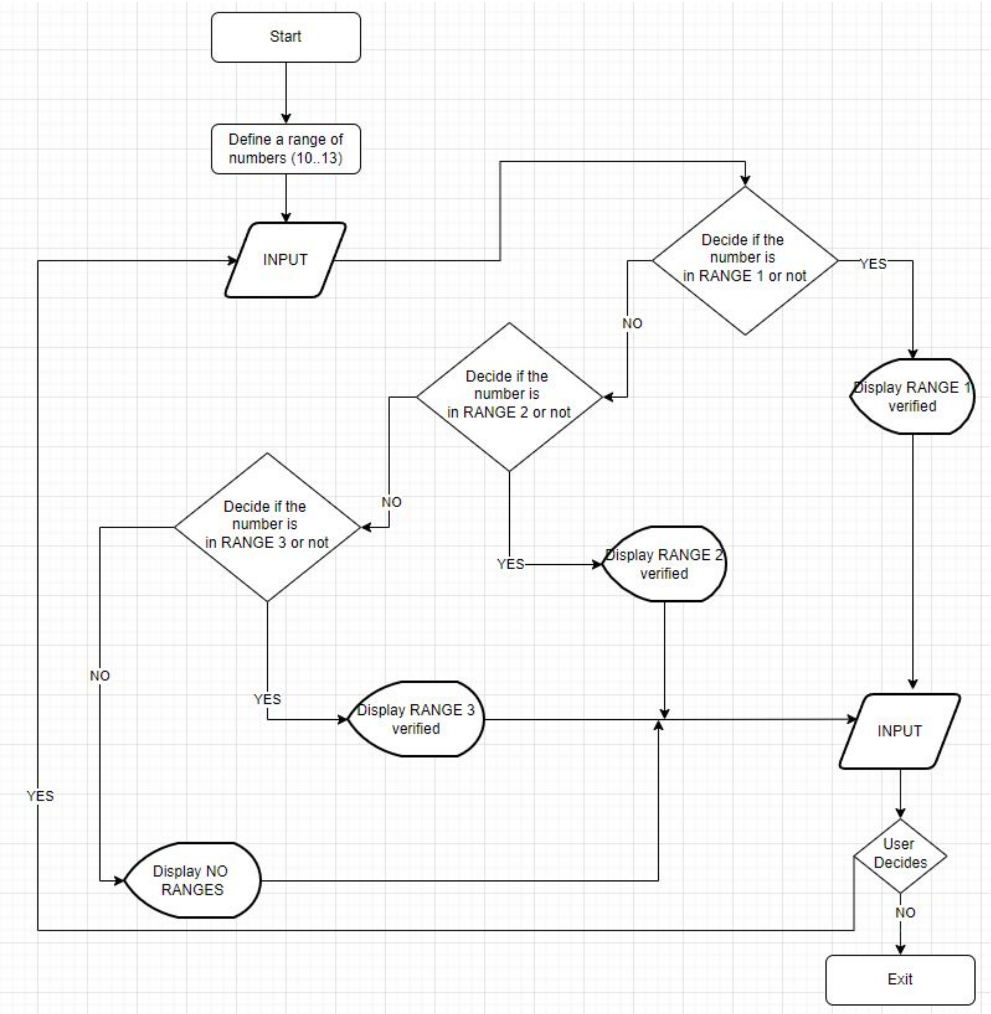
- Write the pseudo code.
- Sketch of the algorithm.
- Write the actual code that runs considering the requirements

Case Study 4 : Pseudocode

- Define the range of numbers.
- Take the input and store it in a variable.
- Compare the input value with the predefined range1.
- Compare the input value with the predefined range2.
- Compare the input value with the predefined range3.
- Display the corresponding messages for each ranges.
- Exit the application if user presses the X key, else repeat the action from the first step.
- If user presses the X key exit the application.

Case Study 4

Algorithm


Start
Define a range of numbers (10..13)
INPUT
Decide if the number is in RANGE 1 or not
YES
NO
Display RANGE 1 verified
Decide if the number is in RANGE 2 or not
NO
YES
Display RANGE 2 verified
Decide if the number is in RANGE 3 or not
NO
YES
Display RANGE 3 verified
Display NO RANGES
INPUT
User Decides
YES
NO
Exit

CLARUSWAY©
WAY TO REINVENT YOURSELF

Case Study 4 - The code in c#

```
internal class Program
    {
        static bool Check(int TheInput)
        {
            switch (TheInput)
            {
                case 0:
                    Console.WriteLine("The input is 0"); break;
                case 1:
                    Console.WriteLine("The input is 1"); break;
                case 2:
                    Console.WriteLine("The input is 2");
                    Console.WriteLine("What to do with number 2");
                    break;
                default:
                    Console.WriteLine("The input is none of above");
                    break;
            }
            return true;
        }
        static void Main(string[] args)
        {
            for (int i = 0; i < 10; i++)
            {
                Console.WriteLine("The number is : " + i.ToString());
            }
            Console.ReadKey();
        }
    }
```

Case Study 4 - The python code

```
user_input = int(input('Please enter a number in one of ranges:10..13 - 18..22 - 25..44 : '))

while(user_input != 0):
    result = False
    if user_input in range(10, 13):
        result = True
    if(user_input in range(18,22)):
        result = True
    if (user_input in range(25,44)):
        result = True

    if result:
        print("The entered value is in range")
    else:
        print("No the entered value is not in any of ranges : {}".format(user_input))
    user_input = int(input('Please enter a number in one of ranges:10..13 - 18..22 - 25..44 : '))

print("End of the application")
```

Case Study 4 : Summary

What we learned from Case Study 4 :

- We may wrap up our algorithms within an infinite loops, we may exit the loop when a specific condition is met.
- In the next session on Friday we will investigate and actively use the switch/case statement as well as for loops and several common structural decision making statements such as break/continue and etc.

Case Study 5

Write a code snippet that takes input from the user until it's a valid e-mail address.

Case Study 5: Pseudocode

- Take the input from the user.
- Create a function that verifies the user input
- Display the result

Case Study 5 - The python code

```
import re

regex = '^[a-z0-9]+[\._]?[a-z0-9]+[@]\w+[.]\w{2,3}$'

def check(email):
    if(re.search(regex,email)):
        return True
    else:
        return False

result = False

while(result != True):
   email = input('Please enter a valid email : ')
   result = check(email)
   if result:
      print('Thank you, mail is :{}'.format(email))
      exit()
   else:
      print('Not a valid email, please try again ...')

print("End of the application")
```

Case Study 6

Car Rental Application

Display a menu
1] Enter customer details : Customer Age, Name, Phone, Address, verified e-mail address
2] Enter car details to be rented.
3] Enter days/hours and km (calculation)
4] Print the billing info.
5] Quit the application

* If 0..5km IndexForKm = 0.05
* If 5..10km IndexForKm = 0.1
* If 10..15km IndexForKm = 0.15
* If 15..20km IndexForKm = 0.2

How can we decompose the application?
Is there a chance to apply pattern recognition?
What can we do to satisfy Abstraction?
How will the algorithm be developed?

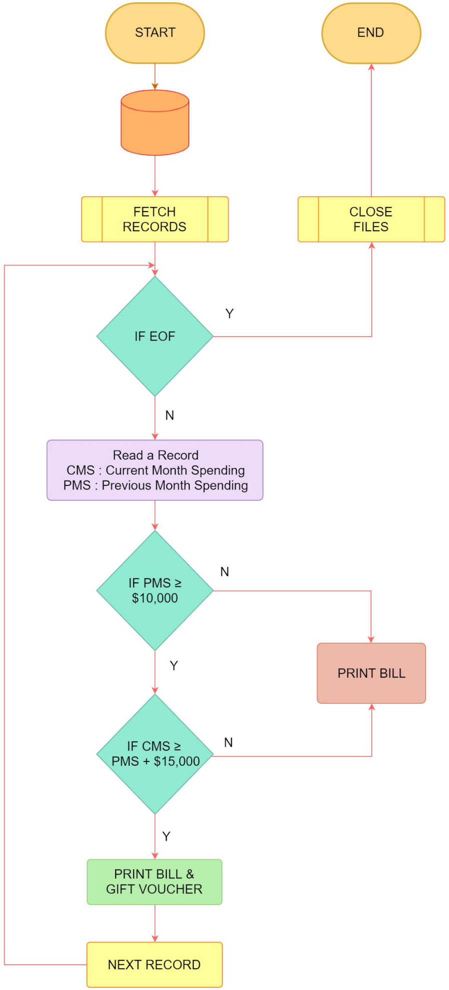
Case Study 7

Bank of America has launched a promotion for its credit card customers. According to the promotion, the customers will receive a gift voucher worth $500 with their monthly bill if they spend $15,000 more than their last month spending and their last month bill is not less than $10,000.

Case Study 7

Flowchart

Draw the flowchart of the promotion



CLARUSWAY©
WAY TO REINVENT YOURSELF

THANKS!

Any questions?

You can find me at:

- @volkan
- volkan@clarusway.com