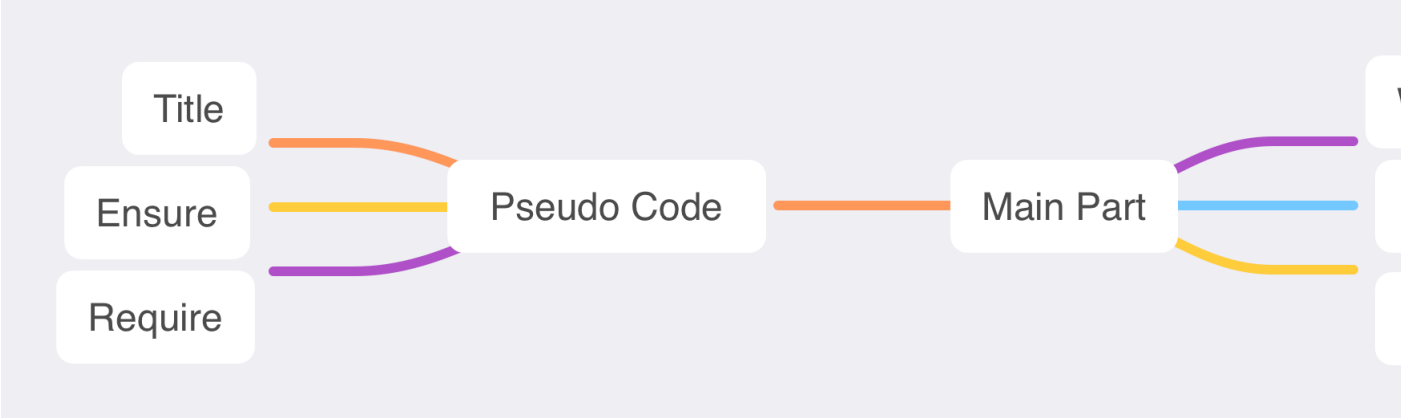
Problem 3

Name: Cheng Chen
ID:40222770
function: tan(x)
Concordia University
SOEN 6011: Software Engineering Processes 17 July 2022

1 Mindmap



2 Algorithm 1

2.1 Description

Algorithm 1 is based on the taylor series of $tan(x)$. ($tan(x) = x + 1/3 \times x^3 + 2/15 \times x^5 + ...$). This formula is used to calculate the approximation of $tan(x)$.

2.2 Technical reasons

- advantages:Since the taylor series is a hard coded formula, it is easy to implement.
- disadvantages:Since it's impossible to include all items in the full taylor serie of $tan(x)$, the result is an approximation, not the real result. What can be done is to include as more items as possible to increase the accuracy.

2.3 Pseudo code

Algorithm 1 Implementation

Require: $x \in R, x \neq \pi/2 + k \times \pi$
Ensure: x is real number, if not, ask the user to input again
Start
make x between 1.57 and -1.57
while True do
$x \leftarrow input$
$y \leftarrow x + x^3/(1 \times 3) + (2 \times x^5)/(1 \times 3 \times 5)$
output y
end while if user exits the program
end while

3 Algorithm 2

3.1 Description

Algorithm 2 uses the value of $tan(0.01)$ and $tan(a+b) = (tan(a) + tan(b))/(1 - tan(a) * tan(b))$. The input x will be separated by 0.01 for the calculation. For example, $tan(0.01 + 0.01) = (tan(0.01) + tan(0.01))/(1 - tan(0.01) * tan(0.01))$, then $tan(0.02)$ can be calculated. Therefore it can be easily conducted that $tan(0.03), tan(0.04) and ...$ can be calculated.

3.2 Technical reasons

- advantages:Since the $tan(x)$ in algorithm 2 is calculated according to $tan(a+b) = (tan(a) + tan(b))/(1 - tan(a) * tan(b))$, the output is closer to the real $tan(x)$ than algorithm 1.
- disadvantages:First, every input x will be separated by 0.01 one by one, it takes more time than algorithm 1 to calculate $tan(x)$. Second, the problem of accuracy also exists. If the user input a x like 0.0003 or 12.384727. Algorithm 2 can just calculate them as $tan(0.00)$ or $tan(12.39)$. The more accuracy it is (like being separated by 0.0001 or a smaller number), the more time it takes. Therefore, the number to be used for separated here is 0.01.

3.3 Pseudo code

Algorithm 2 Implementation

Require: $x \in R$
Ensure: x is real number, if not, ask the user to input again

```
Start
make x between 1.57 and -1.57
while True do
    x ← input
    out ← 0;
    while x > 0 do
        out= (out+0.010000333346667207)/(1-out*0.010000333346667207)
        x-=0.01
    end while
    y ← out
    output y
    end while if user exits the program
end while
```