

# Assignment Lab 4 - 6

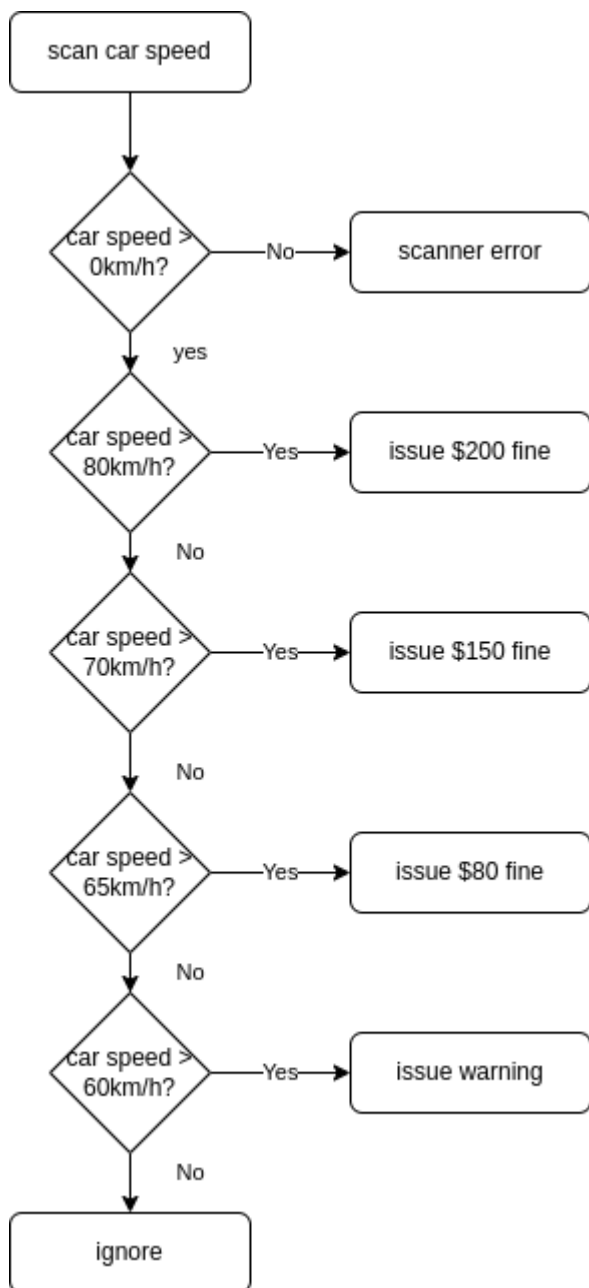
## Lab 4

### Algorithm and Flowchart - Exercise 5

- Algorithm

1. scan / input car speed.
2. if car not moving more than 0km/h, return that scanner is experiencing error.
3. else if car is moving more than 80km/h issue \$200 fine.
4. else if car is moving more than 70km/h issue \$150 fine.
5. else if car is moving more than 65km/h issue \$80 fine.
6. else if car is moving more than 60km/h issue warning.
7. else ignore.

- Flowchart



#### Testing - Exercise 6

Test #	Test Description	Inputs	Expected Outputs	Algorithm Output	Program success / failure
1	Not speeding but very close	59	ignored	ignored	success
2	Speeding but only just (warning)	61	warning	warning	success

3	Still speeding, but only just (warning)	64	warning	warning	success
4	\$80 fine, but only just	66	\$80 fine	\$80 fine	success
5	\$80 fine, but only just	67	\$80 fine	\$80 fine	success
6	\$150 fine but only just	71	\$150 fine	\$150 fine	success
7	Still \$150 fine but only just	72	\$150 fine	\$150 fine	success
8	\$500 fine but only just	81	\$200 fine	\$200 fine	success
9	null input		scanner error	scanner error	success
10	zero input	0	scanner error	scanner error	success
11	negative speed	-1	scanner error	scanner error	success
12	invalid input	asd	scanner error	scanner error	success

## Programming - Exercise 7

CODE LINK: <https://onlinegdb.com/B7PAW2CGy>

```
#include <stdio.h>

int main() {

    int speed = 0; // declare variable for speed

    printf("\n\n Input car speed: \t"); // tell user to input car speed

    scanf("%d", &speed); // store car speed input.

    // issue fines depending on speed
    if (speed <= 0) {
        printf("\n\t Scanner error.");
    } else if (speed > 80) {
        printf("\n\t Issue $200 fine.");
    } else if (speed > 70) {
        printf("\n\t Issue $150 fine.");
    } else if (speed > 65) {
        printf("\n\t Issue $80 fine.");
    } else if (speed > 60) {
        printf("\n\t Issue warning.");
    } else {
        printf("\n\t Ignore");
    }

    return 0;
}
```

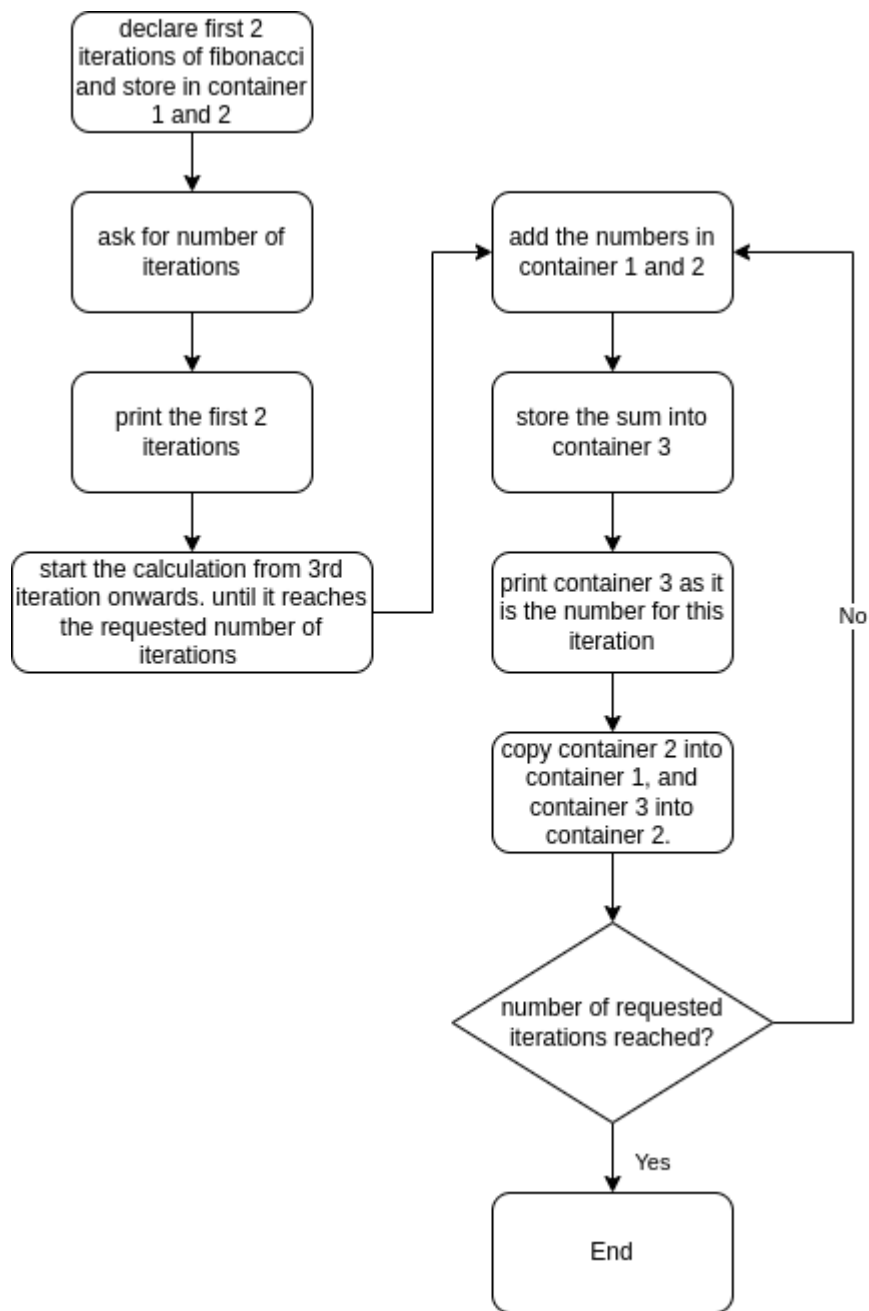
## Lab 5

### Algorithm and Flowchart

- Algorithm

1. Declare that 0 and 1 as the first 2 iterations' set numbers. Name them container 1 and container 2.
2. Declare a variable that holds the addition of each iteration. Name this container 3.
3. Request user to input number of iterations. Must be positive integer.
4. Start a loop that adds the two declared iterations and store into container 3. Counter must start at 3 as the first 2 iterations are already established.
5. Print container 3's value for the iteration.
6. Copy container 2 into container 1 & copy container 3 into container 2.
7. Continue loop until counter reaches iteration number.

- Flowchart



#### Test Cases

Test no.	Input	Expected Output	Program Output	Result
1	10	10th iteration = 34	34	success
2	5	5th iteration = 3	3	success
3	0	Makes user try again	Try again	success
4	-10	Makes user try again	Try again	success

5	90	90th iteration = 1779979416004714189 (sourced online)	1779979416004714189	success
---	----	---	---------------------	---------

## Programming

CODE LINK: <https://onlinegdb.com/dQGbCSIde>

```
#include <stdio.h>

int main() {

    // declare variables
    long long container1 = 0; // set number of the first iteration
    long long container2 = 1; // set number of the second iteration
    long long container3 = 0; // stores the additions
    int iterations = 0; // stores the requested iterations
    int counter = 0; // counter

    // prompt user to input number of iterations make sure only to accept a number
    bigger than 0
    while (iterations <= 0) {
        printf("\n-----FIBONACCI SEQUENCE-----\n\n");
        printf("\n How many iterations would you like?\n MUST BE HIGHER THAN 0: \t");
        scanf("%d", &iterations); //take input and store
    }

    printf("\n\n Iteration #1 : %lld", container1);
    if(iterations>1){
        printf("\n Iteration #2 : %lld", container2);
    }

    // counter set as 3, as the first 2 Iterations are already set and printed.
    for (counter = 3; counter <= iterations; counter++) {
        container3 = container1 + container2;
        printf("\n Iteration #%d : %lld", counter, container3);
        container1 = container2; // copy container 2's value to container1
        container2 = container3; // copy container3's value to container2
    }

    return 0;
}
```

## Lab 6

### Algorithm

- Algorithm
1. Request user to input 3 values and store them.
  2. Execute function "max" that uses if and else statements to find the largest value and prints it. Change the value within the pointer for "max" as it needs to be used later.

3. Execute function "min" that uses if and else statements to find the smallest value and prints it. Change the value within the pointer for "min" as it needs to be used later.
4. Execute function for average that takes 3 arguments which are the inputs, and averaged by 3. Not programming dynamically so hardcode the number 3. Store the result in the "average" pointer and print it.
5. Execute function for "differences". Take arguments for max and min input values, and the average value. Print the differences between max and min, and between max and average.
6. Execute function for "absolute". Function only takes the min input value. If input value is 0 or higher just print that number. If not multiply by -1 and print that number as absolute value.

## Programming

CODE LINK: <https://onlinegdb.com/-Vzc7cplJ>

```
#include <stdio.h>

// function prototypes
void max(int input1, int input2, int input3, int *max);      // find max
void min(int input1, int input2, int input3, int *min);      // find min
void avg(int input1, int input2, int input3, float *average); // find avg
void diff(int maxNum, float average,
           int minNum);   // find diff between max and min, and max and avg
void absolute(int minNum); // find absolute value

int main() {

    // declare variables
    int input1 = 0;    // store first input
    int input2 = 0;    // store second input
    int input3 = 0;    // store third input
    int maxNum = 0;    // store max num
    int minNum = 0;    // store min num
    float average = 0; // store average value

    // prompt user to input numbers
    printf("\n Please input the first number: ");
    scanf("%d", &input1);
    printf("\n\n Please input the second number: ");
    scanf("%d", &input2);
    printf("\n\n Please input the third number: ");
    scanf("%d", &input3);

    max(input1, input2, input3, &maxNum);
    min(input1, input2, input3, &minNum);
    avg(input1, input2, input3, &average);
    diff(maxNum, average, minNum);
    absolute(minNum);

    return 0;
}

// function declaration to find max val
void max(int input1, int input2, int input3, int *maxNum) {
```

```

    if (input1 >= input2) {
        *maxNum = input1;
    } else {
        *maxNum = input2;
    }
    if (input3 > input2) {
        *maxNum = input3;
    }
    printf("\n The maximum input value is: %4d", *maxNum);
}

// function definition to find min val
void min(int input1, int input2, int input3, int *minNum) {
    if (input1 <= input2) {
        *minNum = input1;
    } else {
        *minNum = input2;
    }
    if (input3 < input2) {
        *minNum = input3;
    }
    printf("\n The minimum input value is: %4d", *minNum);
}

// function definition to find avg val
void avg(int input1, int input2, int input3, float *average) {
    *average = (input1 + input2 + input3) /
        3.0; // the number 3 is a set number as the
        // question only requested 3 NOT dynamic.
    printf("\n The average is: %4.2f", *average);
}

// function definition to find differences
void diff(int maxNum, float average, int minNum) {
    printf("\n The difference between max and min: %4d", maxNum - minNum);
    printf("\n The difference between max and average: %4.2f", maxNum - average);
}

// function definition to find absolute value
void absolute(int minNum) {
    int abVal = 0; // store absolute value. doesnt need to be in main program, not
        // needed in any other functions.
    if (minNum >= 0) {
        abVal = minNum;
    } else {
        abVal = minNum * -1;
    }
    printf("\n The absolute value for the minimum input: %4d", abVal);
}

```