

# Programming Lab Report

Vivid Padungkiatsakul 6601023620026<sup>1</sup>

<sup>1</sup>Robotic Engineering and Automation System, Faculty of Engineering, King  
Mongkut's University of Technology North Bangkok

[010243107] C Programming 2023

## Lab Week 4

# Loop operation and Condition Statement

---

### Topic

- while
  - do..while
  - For
  - Nested loop
  - Break/Continue
-

# Experiment I

---

## 1 Problem description

- By using **while loop**, write a program that gets 5 value from user and shows the sum value.
- Also, verify your program.

## 2 Program design

- So we set 3 variables. count[For counting loop], user[for user input value] and sum[for sum number]
- For program, we will get user input
- For verify and testing. we will use x[5] set. In x will have {1,2,3,4,5}.
  - If testing return 1, the program is correct.
  - If testing return 0, the program is incorrect.

## 3 Program text

```
#include <stdio.h>

int count=1,user,sum=0;
int main(){
    while(count<=5){
        printf("Input number: ");
        scanf("%d",&user);
        sum = sum+user;
        count++;
    }

    printf("Sum of numbers: %d\n", sum);
}
```

Main Program

```
#include <stdio.h>

int count=1,user,sum=0,i;
int x[5] = {1,2,3,4,5};

int main(){
    for(i=1;i<=5;i++){
        i = count;
        while(count<=5){
            user = x[count-1];
            sum = sum+user;
            count++;
        }
    }
    printf("%d",sum == 15);
}
```

Test Program

Part 4: Terminal output is on the next page.

#### 4 Terminal output

```
Input number: 1  
Input number: 2  
Input number: 3  
Input number: 4  
Input number: 5  
Sum of numbers: 15
```

```
1
```

Termianl: Test program

Termianl: Main program

---

# Experiment II

---

## 1 Problem description

- By using **do-while loop**, write a program that gets number of people that enter or exit a supermarket and shows current number of people in the supermarket. Stop the program when '0' is obtained.
- Also, verify your program.

## 2 Program design

- The do-while loop work as while loop but write in different prefix.
- So we set 2 variables. current[For sum total people], input[For get user input]
- The loop will break when user input 0. That mean, the condition in while is input != 0.
- For program, we will get user input.
- For verify and testing. we will use x[10] set. In x will have 1,2,3,4,5,-2,-3,6,-4,0
  - If testing return 1, the program is correct.
  - If testing return 0, the program is incorrect.

## 3 Program text

```
#include <stdio.h>

int current=0, input;

int main(){
    do
    {
        printf("Insert No. people:");
        scanf("%d",&input);
        current += input;
        printf("Sum of numbers: %d\n",
            current);
    } while (input != 0);
}
```

Main Program

```
#include <stdio.h>

int current=0, input, i;
int x[10] = {1,2,3,4,5,-2,-3,6,-4,0};

int main() {
    for (i = 0; i <= 9; i++) {
        input = x[i];
        do {
            current += input;
            input = x[++i];
        } while (input != 0);
    }
    printf("%d", current==12);
}
```

Test Program

Part 4: Terminal output is on the next page.

## 4 Terminal output

```
Sum of numbers: 1
Insert No. people:2
Sum of numbers: 3
Insert No. people:3
Sum of numbers: 6
Insert No. people:4
Sum of numbers: 10
Insert No. people:5
Sum of numbers: 15
Insert No. people:-2
Sum of numbers: 13
Insert No. people:-3
Sum of numbers: 10
Insert No. people:6
Sum of numbers: 16
Insert No. people:-4
Sum of numbers: 12
Insert No. people:0
Sum of numbers: 12
```

Termianl: Main program

---

1

Termianl: Test program

# Experiment III

---

## 1 Problem description

- By using **for loop**, write a program that sums up value from 100 to 120. Also, find average value.
- Also, verify your program.

## 2 Program design

- The for loop is like while but in the limit that we can set .
- So we set 4 variables. count[For counting], sum[For sum number], i[For the for loop],
- From problem we have to start from 101 to 120 and we can step by 1. That mean, the condition in for is `i=101;i<=120;i++`.
- For verify and testing. The output from sum and avg have to be 2210 and 110.0000000
  - If testing return 1, the program is correct.
  - If testing return 0, the program is incorrect.

## 3 Program text

```
#include <stdio.h>

int count=0;
int sum,i;
float avg;

int main(){
    for (i=101;i<=120;i++)
    {
        printf("i=%d\n",i);
        sum += i;
        count++;
    }
    avg = sum/count;
    printf("Sum=%d Avg=%f",sum, avg);
}
```

Main Program

```
#include <stdio.h>

int count=0;
int sum,i;
float avg;

int main(){
    for (i=101;i<=120;i++)
    {
        sum += i;
        count++;
    }
    avg = sum/count;
    printf("%d",sum==2210&&avg==110);
}
```

Test Program

Part 4: Terminal output is on the next page.

#### 4 Terminal output

```
i=101  
i=102  
i=103  
i=104  
i=105  
i=106  
i=107  
i=108  
i=109  
i=110  
i=111  
i=112  
i=113  
i=114  
i=115  
i=116  
i=117  
i=118  
i=119  
i=120  
Sum=2210 Avg=110.000000
```

Termianl: Main program

```
1
```

Termianl: Test program

---



# Experiment IV

---

## 1 Problem description

- By using **nested for loop**, write a program that shows combinations of throwing 2 dices and sum of marks is equal to 7.
- Ex. 3+4, 4+3, 2+5, 5+2,....
- Also, verify your program.

## 2 Program design

- The nested for loop is for loop but have other for loop in it.
- So we set 2 variables. i and j for for loop.
- From problem, we want the combination that equal to 7 that mean we have to set if-else. the condition in for is i=1;i=6;i++ and j=1;j=6;j++ and the condition in if else is i+j==7.

## 3 Program text

```
#include <stdio.h>

int i,j;

int main(){
    for (i=1;i<=6;i++){
        for (j=1;j<=6;j++){
            if (i+j==7)
            {
                printf("%d+%d\n",i,j);
            }
        }
    }
}
```

Part 4: Terminal output is on the next page.

#### 4 Terminal output

```
1+6  
2+5  
3+4  
4+3  
5+2  
6+1
```

---

*End of Lab Experiment Week 4*

# Experiment Extra I

---

## 1 Problem description

- According to table below, change condition of while loop in lab4-1.c,

Q 1.1: run and record number of iteration that "Hello" is displayed on screen as well as giving the reason in the last column.

Trial No.	Condition	No. of iteration	Reason
1	true		
2	1		
3	false		
4	0		
5	'a'		
6	22.55		
7	i!=5		
8	i<8		
9	i>5 && i<15		
10	i>5 ——— i<15		

Q 1.2: Write a flowchart of trail no. 7 this program.

## 2 Problem program text

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

int main(void){
    int i=1;

    while(22.55){
        printf("Hello\n");
        i++;
    }

    return 0;
}
```

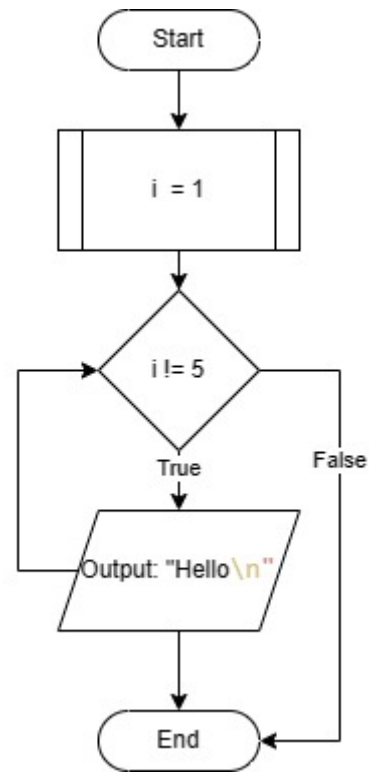
### 3 Answer

Ans 1.1: Table of Experiment I

Trial No.	Condition	No. of iteration	Reason
1	true	inf.	When the condition in while() is true, It will continue looping until condition is false
2	1	inf.	Same reason as No.1, 1 in boolean is true
3	false	None.	The condition in while() is false. so the loop is break
4	0	None.	Same reason as No.3, 0 in boolean is false
5	'a'	inf.	The condition in while() is 'a' which is always true
6	22.55	inf.	Same reason as No.5, The condition is 22.55 is always true
7	i!=5	4	So the condition is when i not equal to 5 and i is 1. So, the loop will be run only 4 times.
8	i<8	7	Same as No.7, The condition is true when i is less than 8. When i is equal 8, the loop will break
9	i>5 && i<15	None.	The condition is when i more than 5 and less than 15. i is equal to 1 so it will be false and true equal to false. So, The loop will break.
10	i>5 — i<15	inf.	The condition is when i more than 5 or less than 15. There are no case that i is less than 5 and more than 15. So, The condition will always turn

**Part 4: Flowchart design is on the next page.**

#### 4 Flowchart design



Flowchart trial No.7 [Ans 1.2]

# Experiment Extra II

---

## 1 Problem description

- According to table below, change condition of while loop in lab4-2.c,

Q 2.1: run and record number of iteration that "Hello" is displayed on screen and record value of 'i' of each iteration in the table.

Trial No.	Exp1	Exp2	Exp3	No. of iteration	Value of variable 'i' of each iteration
1	i=0	i<10	i++		
2	i=0	i<=10	i++		
3	i=1	i<=10	i++		
4	i=1	i<=10	i+=2		
5	i=10	i>=0	i--		
6	i=10	i>=0	i-=2		

Q 2.2: Write a flowchart of trail no. 4 of this program.

Q 2.3: Explain how to modify this program in order to calculate sum and average of values from 101 to 250.

## 2 Problem program text

```
#include<stdio.h>

int main(void)
{
    int i;
    int count = 0;
    int sum = 0;
    for(i=10;i>=0;i--)
    {
        printf("Hello %d\n",i);
        count++;
        sum = sum+i;
    }
    printf("%d %f", count, (float)(sum/count));

    return 0;
}
```

### 3 Answer

Ans 2.1: Table of Experiment II

Trial No.	Exp1	Exp2	Exp3	No. of iteration	Value of variable 'i' of each iteration
1	i=0	i<10	i++	10	Increasing in loop from 0 to 9
2	i=0	i<=10	i++	11	Increasing in loop from 0 to 10
3	i=1	i<=10	i++	10	Increasing in loop from 1 to 10
4	i=1	i<=10	i+=2	5	Increasing in loop in odd number from 1 to 9 {1,3,5,7,9}
5	i=10	i>=0	i--	11	Decreasing in loop from 10 to 0.
6	i=10	i>=0	i-=2	6	Decreasing in loop in even number from 10 to 0 {10,8,6,4,2,0}

Ans 2.3: In this case, we can modify programming in 2 ways.

Case 1: modify only for loop: In this case we can change condition to i=101;i=250;i++

Case 2: modify for loop to count in odd number: In this case similar like case above but we change from i++ to i+=2 and multiply be 2 in count after breaking loop for lower time and faster compiling code.

```
#include <stdio.h>

int main(void)
{
    int i;
    int count = 0;
    int sum = 0;
    for(i=101; i<=250; i++)
    {
        printf("Hello %d\n", i);
        count++;
        sum = sum+i;
    }
    printf("%d %f", count, (float)(sum/count));

    return 0;
}
//150 175.000000
```

Case 1

```
#include <stdio.h>

int main(void)
{
    int i;
    int count = 0;
    int sum = 0;
    for(i=101; i<=250; i+=2)
    {
        printf("Hello %d\n", i);
        count++;
        sum = sum+i;
    }
    printf("%d %f", count*2, (float)(sum/count));

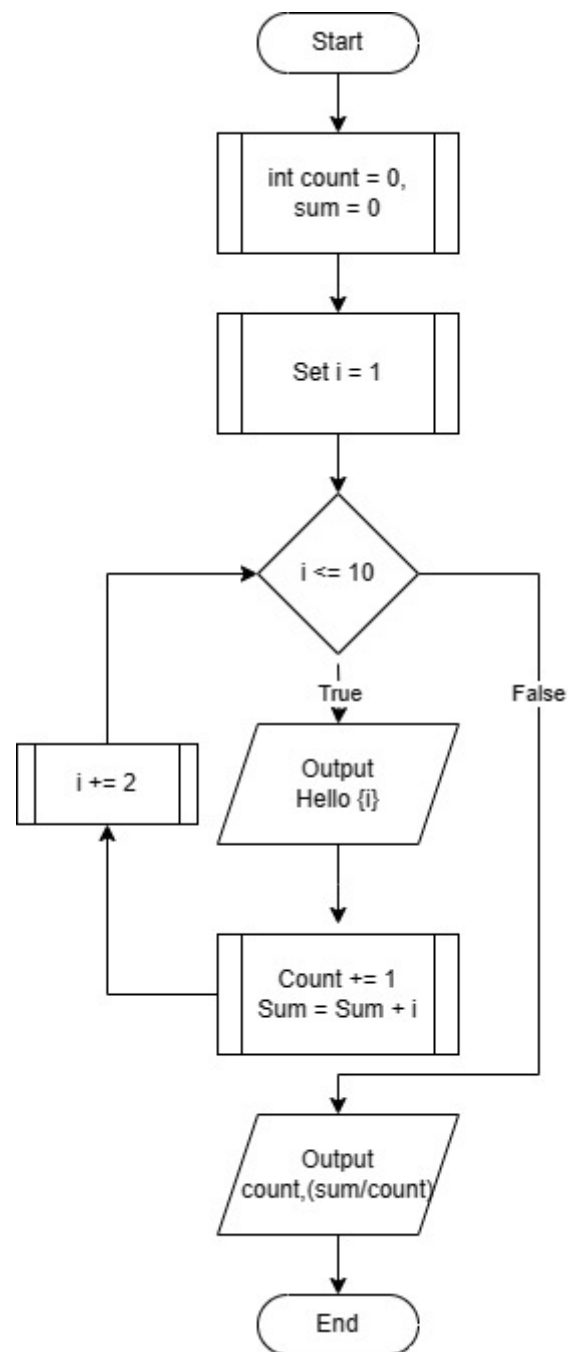
    return 0;
}
//150 175.000000
```

Case 2

All 2 case output same number.

**Part 4: Flowchart design is on the next page.**

#### 4 Flowchart design



Flowchart trial No.4 [Ans 2.2]



# Experiment Extra III

---

## 1 Problem description

- According to table below, change value of hidden\_number in lab4-3.c,

Q 3.1: run and put in your guess as in the table. Record number of iteration that "Guess my number:" is displayed on screen as well as giving the reason.

Trial No.	hidden_number	Guess number	No. of iteration	Reason
1	5	5		
2	5	7,12,2,19,4,16,5		
3	7	7		
4	7	8,2,9,10,20,6,3,1,7		

Q 3.2: Write a flowchart of this program.

## 2 Problem program text

```
#include<stdio.h>

int main(void)
{
    int hidden_number;
    int user_guess;
    hidden_number = 5;
    user_guess = 0;

    while(user_guess!=hidden_number)
    {
        printf("Guess my number :");
        scanf("%d", &user_guess);
    }
    printf("Good guess !!\n");
    return 0;
}
```

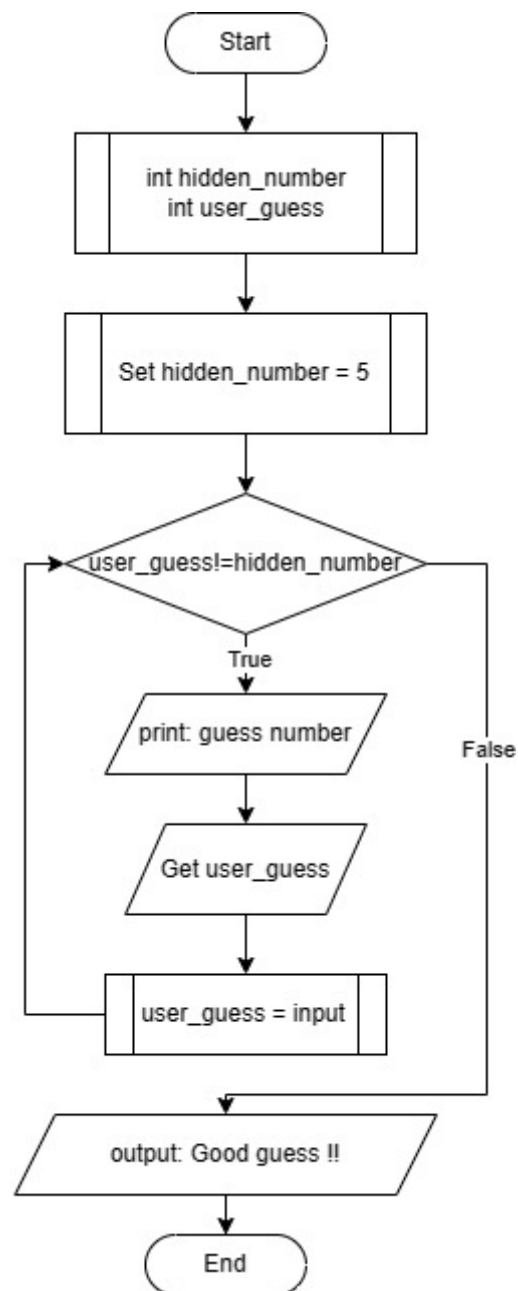
### 3 Answer

Ans 3.1: The condition is when user\_guess not equal to hidden\_number, get the number from input to user\_guess and loop again.

Trial No.	hidden_number	Guess number	No. of iteration	Reason
1	5	5	1	In this case, hidden_number is not equal to user_guess. So it will loop and when user input same number (that is 5). It will break. That mean it run in loop only 1 time.
2	5	7,12,2,19,4,16,5	7	In this case, hidden_number is not equal to user_guess. So it will loop. When user input number that not equal to hidden_numeber. It will continue looping. Until user input number equal to hidden_number. The loop will break.
3	7	7	1	Same reason as No.1
4	7	8,2,9,10,20,6,3,1,7	9	Same reason as No.2

**Part 4: Flowchart design is on the next page.**

#### 4 Flowchart design



Flowchart of programming process [Ans 3.2]

# Experiment Extra IV

---

## 1 Problem description

- According to table below, change statement in lab4-4.c,

Q 4.1: run and record number of iteration and value of sum. Then, give the reason.

Trial No.	statement	No. of iteration	Sum	Reason
1	;			
2	continue;			
3	break;			

Q 4.2: Write a flowchart of trial no. 1.

## 2 Problem program text

```
#include<stdio.h>

int main(void)
{
    int x;
    int sum = 0 ;

    for(x=1;x<=9;x++){
        if(x==5){
            break;                //Change this statement
        }
        else
        {
            sum +=x;
            printf("+%d ", x);
        }

    }
    printf("= %d", sum);

    return 0;
}
```

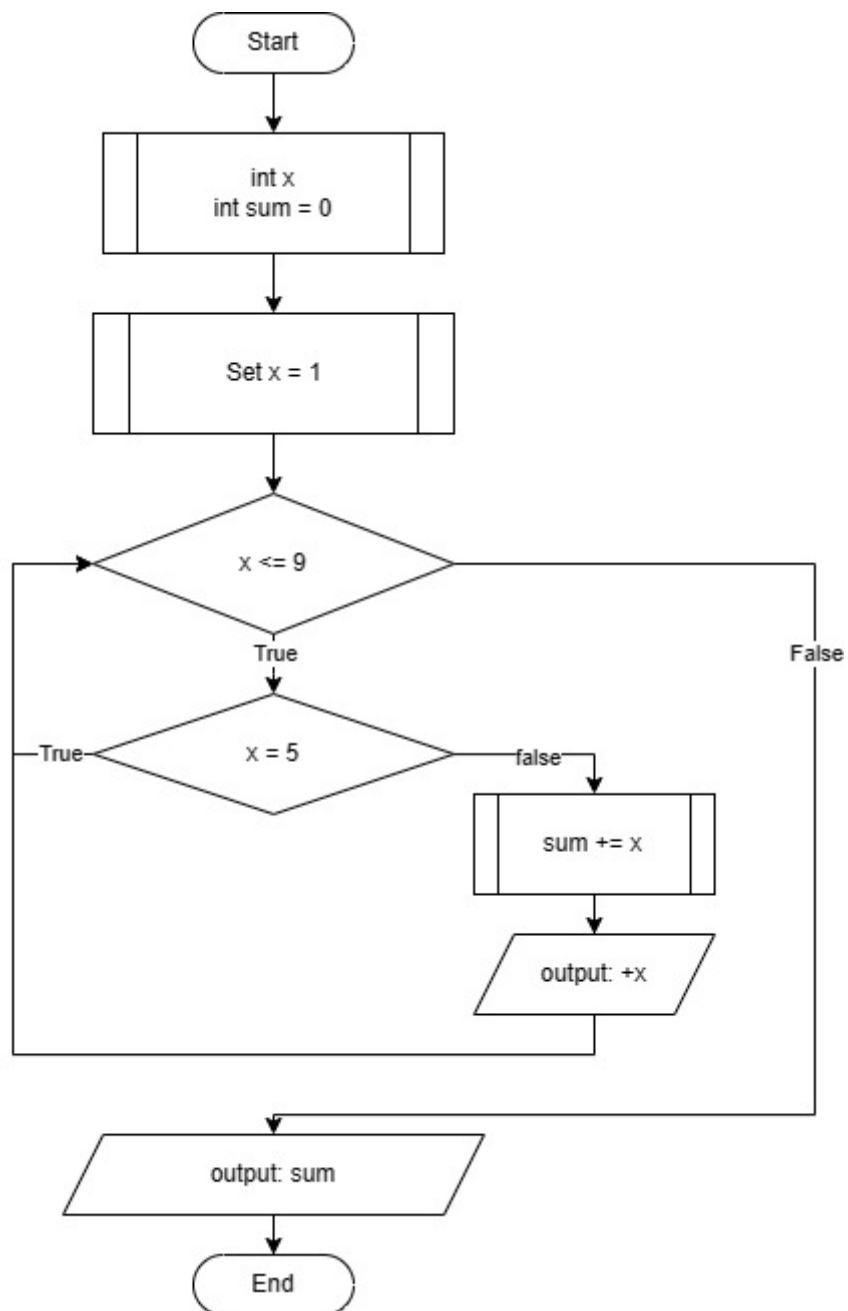
### 3 Answer

Ans 4.1: Table of Experiment IV

Trial No.	statement	No. of iteration	Sum	Reason
1	;	8	40	When x equal to 5, The if condition will be true. When it true, there no operation in if case. So the loop will continue.
2	continue;	8	40	Same as No.1, the continue function make loop continue looping.
3	break;	4	10	When x equal to 5, The if condition will be true. When it true, it will run break function and break (out) looping.

**Part 4: Flowchart design is on the next page.**

#### 4 Flowchart design



Flowchart of trial No. 1 [Ans 4.2]