CONTROL STRUCTURES IN C

ITERATIONS

- If we want to execute a set of statements repeatedly, we use iterative structures also called as loops.
- There are three types of loops in C:
  - for
  - while
  - do..while

FOR LOOP

- Syntax:

```
for(initialization; condition; updation)
{
//statements;
}
```

- Initialization: A loop variable can be used to start the loop by giving it some initial value.
- Condition: To check whether the loop must stop/continue. If condition is false, the control goes outside the loop.
- Updation: Loop variable is updated in some way so that the condition finally becomes false at some point.
- Note: In a way, the loop variable in a for loop accounts for ***number of times loop must execute.***

FOR LOOP

- Example:
- To print “Hello” 10 times
- Initialization: A variable (by convention called i) is initialized to 1
- Condition: Check if i has reached 10. If yes, stop, else continue. Can you guess the condition?
    - i==10
    - i!=11
    - i>10
    - i<=10
- Updation:i must increment by 1 each time.

  So i=i+1 or i++

FOR LOOP

- Example:
- To print “Hello” 10 times

```
for(i=1;i<=10;i++)
{
printf(“Hello”);
}
```

QUICK EXERCISE

- Predict the output:

```
int i;
for(i=1;i<=5;i++)
{
printf("Hello %d\n",i);
}
```

Output:

```
Hello 1
Hello 2
Hello 3
Hello 4
Hello 5
```

QUICK EXERCISE

- Predict the output:

```
int i;
for(i=2;i<=6;i++)
printf("Hello\n”);
printf(“i=%d”,i);
```

Output:

```
Hello
Hello
Hello
Hello
Hello
i=7
```

QUICK EXERCISE

- Predict the output:

```
int i;
for(i=1;i<=5;i++);
printf("Hello\n”);
printf(“%d”,i);
```

<u>Output:</u>

```
Hello
6
```

QUICK EXERCISE

- How many times do each of the below loops execute?

1. for(i=1;i<=5;i++)
2. for(i=0;i<=5;i++)
3. for(i=0;i<5;i++)
4. for(i=5;i<=5;i++)
5. for(i=5;i<=1;i++)
6. for(i=5;i>=1;i++)
7. for(i=5;i>=1;i--)
8. for(i=1;i<=n;i++)
9. for(i=0;i<=n;i++)

QUICK EXERCISE

- How many times do each of the below loops execute?

1.	for(i=1;i<=5;i++)	5 times (i = 1,2,3,4,5)
2.	for(i=0;i<=5;i++)	6 times (i = 0,1,2,3,4,5)
3.	for(i=0;i<5;i++)	5 times (i = 0,1,2,3,4)
4.	for(i=5;i<=5;i++)	1 time (i=5)
5.	for(i=5;i<=1;i++)	0 times (Loop condition false first time)
6.	for(i=5;i>=1;i++)	Infinite times (i = 5,6,7,8,9…)
7.	for(i=5;i>=1;i--)	5 times (i =5,4,3,2,1)
8.	for(i=1;i<=n;i++)	‘n’ times (i = 1,2,3,4,.....n)
9.	for(i=0;i<=n;i++)	n+1 times(i = 0,1,2,3...n)

FOR LOOP

- <u>POINTS TO REMEMBER:</u>
- We can eliminate initialization and updation (by taking them inside or outside the loop), but we cannot eliminate the condition.
- Example:

```
int i=1;
for( ;i<=5; )
{
i++;
}
```

- Each for loop can only have exactly two semi-colons(;)
- We can have multiple initializations and multiple updations, separated by a comma(,)
- Can you guess how can we write multiple conditions?
  - Using appropriate logical operators

QUICK EXERCISE

- How many times do each of the below loops execute?

1. for(i=1,j=7;i<=5 && j>=5;i++,j- -)
2. for(i=1,j=7;i<=5 | | j>=5;i++,j- -)

QUICK EXERCISE

- How many times do each of the below loops execute?
- for(i=1,j=7;i<=5 && j>=5;i++,j- -)

3 times

i=1 , j=7

i=2 , j=6

i=3 , j=5

- for(i=1,j=7;i<=5 | | j>=5;i++,j- -)

5 times

i=1 , j=7

i=2 , j=6

i=3 , j=5

i=4, j=4

i=5, j=3

SUCCESSIVE ARITHMETIC OPERATIONS

- Whenever we want to do an arithmetic operation in a series/sequence such that the previous result is carried forward for the next operation.
- Example: Arithmetic/Geometric series
- The general rule of writing it is :
  - result = result (+,-,*,/%) term

SUCCESSIVE ARITHMETIC OPERATIONS

- Suppose we want to do,

  2+4+6+8

  1. result = result +2
  2. result = result +4
  3. result = result +6
  4. result = result +8
- What should be the first value of result????

  result = 0. Else, we may get garbage value.
- Can we use for loops for the above?

SUCCESSIVE ARITHMETIC OPERATIONS

- Suppose we want to do,

1+2+3+4+5

```
int i, sum = 0;
for(i=1;i<=5;i++)
{
sum = sum + i;
}
printf("%d", sum);
```

Iteration no.	i	sum Initial value=0
1	1	1
2	2	3
3	3	6
4	4	10
5	5	15

SUCCESSIVE ARITHMETIC OPERATIONS

- Suppose we want to do,

2+4+6+8+10

```
int i, sum = 0;
for(i=2;i<=10;i=i+2)
{
sum = sum + i;
}
printf("%d", sum);
```

Iteration no.	i	sum Initial value=0
1	2	2
2	4	6
3	6	12
4	8	20
5	10	30

SUCCESSIVE ARITHMETIC OPERATIONS

- Suppose we want to do,

  1+2+3+4+5+….n   where ‘n’ is user input

```
int i, sum = 0;
scanf(“%d”, &n);
for(i=1;i<=n;i++)
{
sum = sum + i;
}
printf(“%d”, sum);
```

QUICK EXERCISE

- Write a program to do the following series operations:
  - result = 1-2-3-4-5
  - result = 5-4-3-2-1
- Write a program to calculate a * b without using a “*” operator.