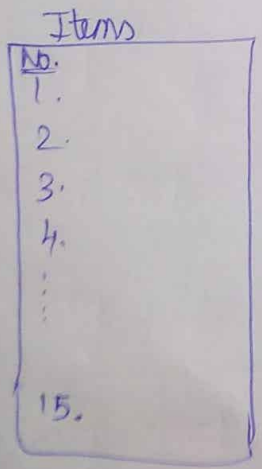
C_33 ⇒ Introduction to Loops in C

Loops:

* Why to use loops? Scenario → going to supermarket again and again to buy items; this is waste of time and energy instead of this, we prepare a list of items in prepared manner and go to super market at a single time and buy items.

* When we get all items; Eg 15 items we go to Billing counter and put bill.



* Print Jenny one time.

```
printf("Jenny\n");
```

* Print Jenny 5 times.

```
printf("Jenny\n");
printf("Jenny\n");
printf("Jenny\n");
printf("Jenny\n");
printf("Jenny\n");
```

* Print Jenny 100 times.

→ It is not possible to write code by Counting the lines by written everytime printf statement.

→ In this scenario we can repeat the printf statement using Loops.

→ So, Loops are used to repeat Some set of programs or statements in a program.

```
loop starts
1. printf("Jenny");
2.
3.
.
.
100. printf("Jenny");
loop ends.
```

→ For this 'i' variables are used.

i=1 loop starts
i=100 loop ends.

→ Increase i value → i++.

Loop terminate (i = 101 ✗) i=100 ✓ i<=100 condition

3 steps for loops

(1) Initialize "i"

(3) Increase "i" value → i++ (updation)

(2) Terminate "i" → give condition

1. i = 1	2. i++	3. i <= 100

Loops types in C

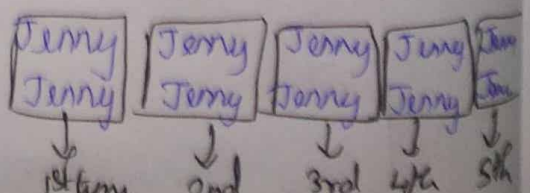
1. Entry Controlled loops.
2. Exit Controlled loops.

Entry Controlled loops

* Terminate condition is put at the start of the loop.
* First Condition is checked and the body of loop executed.

```
loop start (i=1) (i<=5)
    printf ("Jenny");
    printf ("Jenny");
loop end
```

⇒ O/P

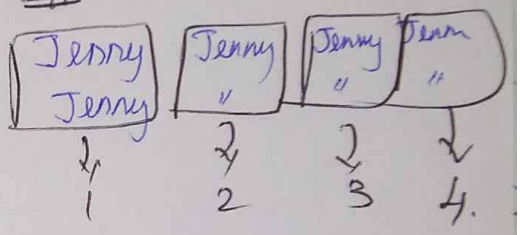


Exit Controlled loop

* Terminate condition is put at the end of the loop.
* First time control will enter into the loop, it will check no condition; but execute set of statements even though at the end of loop the placed condition is true or false.

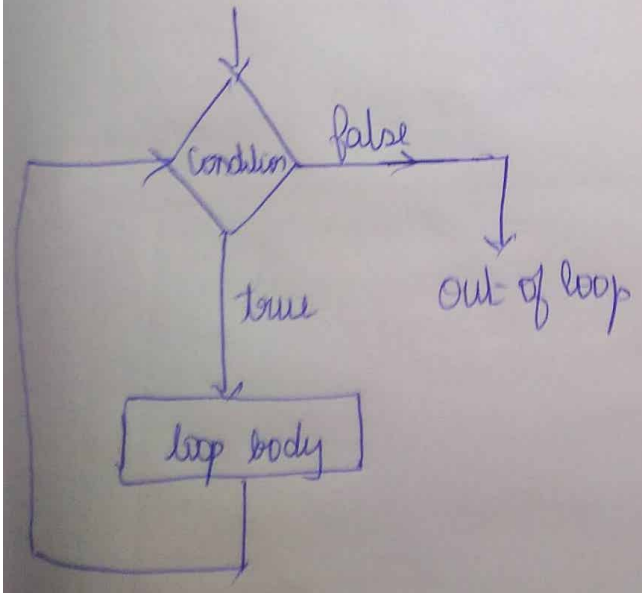
```
loop start (i=1)
    printf ("Jenny");
    printf ("Jenny");
(i<=4)
loop end
```

O/P :-



Flow chart

Entry controlled Loop



Exit controlled Loop.