

```

    s = s + i;
    }
    printf ("Sum of numbers = %d", s);
    }

```

## Lesson 8 Jump Statement

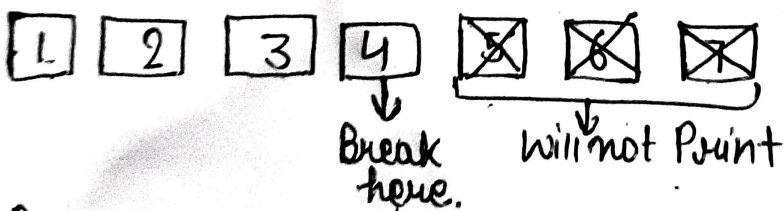
Unit 2

Q. What are Jump statements?  
 Ans → Jump statements in C are used to interrupt flow of the program or escape a particular section of the program. There are many more operations they can perform within the loops, switch statements, and functions. There are four types of Jump Statements.

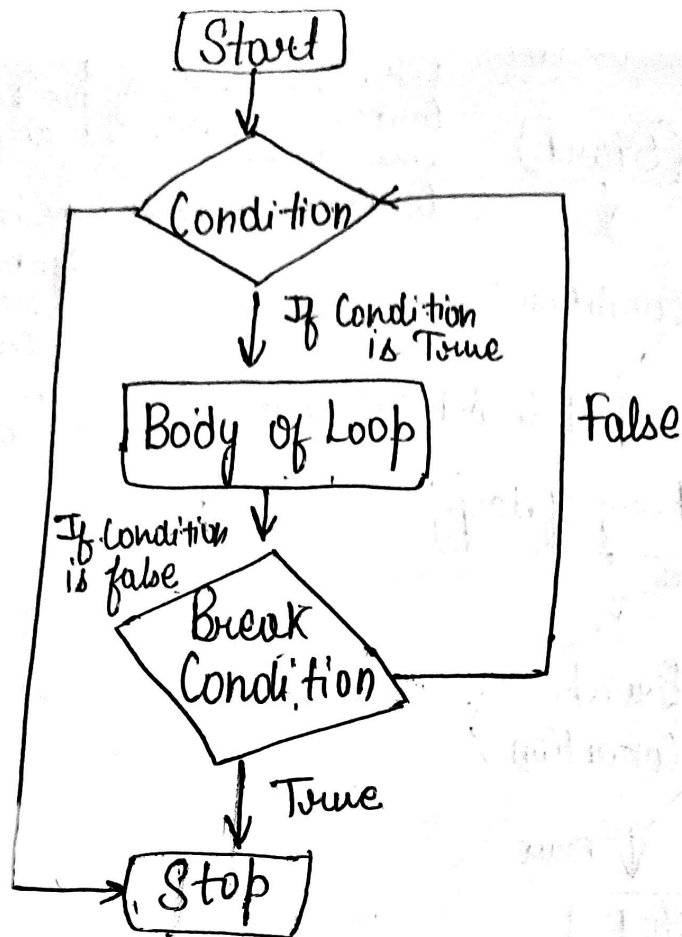
- Break
- Continue
- goto
- return.

### ★ Break Statement in C

Break statement exists the loops like for, while, do-while immediately, brings it out of the loop, and starts executing the next block. It also terminates the switch statement.



### Break Statement Flow Chart



### • Break Statement example

```

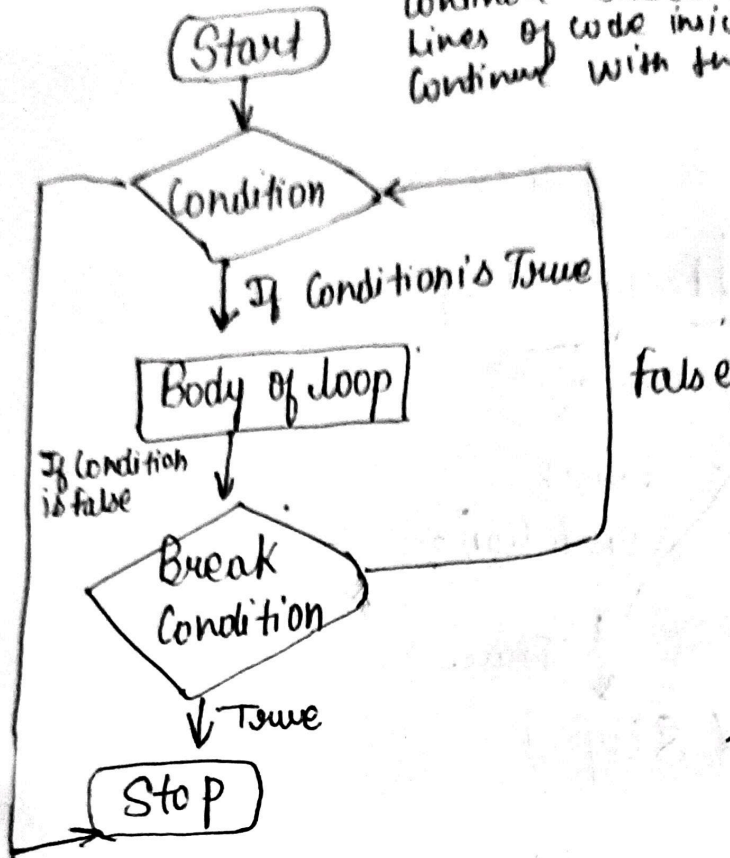
#include <Stdio.h>
void main()
{
  int a=1; // Initialize value to a
  while (a <= 10) // run loop unless the value is 10
  {
    if (a == 3) // If the value is 3
      break; // break the loop
    printf("Print = %d \n", a);
    a++;
  }
  printf("Outside loop"); // Print statement outside
  // the loop
}
  
```



## \* Continue Statement in C

The Continue Statement is used to bring the program control to the beginning of the loop. The Continue Statement skips some lines of code inside the loop and continues with the next iteration.

Syntax:-  
// Loop statement  
Continue;  
// Some lines of the code which is to be skipped.



Continue is a jump statement in C. It skips the specific iteration in the loop. It is similar to the break statement, but instead of terminating the whole loop, it skips the current iteration and continues from the next iteration and so on in the same loop. It brings the control of a program to the beginning of the loop.

Using the continue statement in the nested loop skips the inner loop's iteration only and doesn't affect the outer loop.



## Jump Statements

### Return Statement in C

#### Continue Statement Example

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

```
int main()
```

```
{ int i;
```

```
for (i=1; i<=7; i++);
```

```
{ if (i==3) // Continue Condition
```

```
{ continue ; // Continue Statement
```

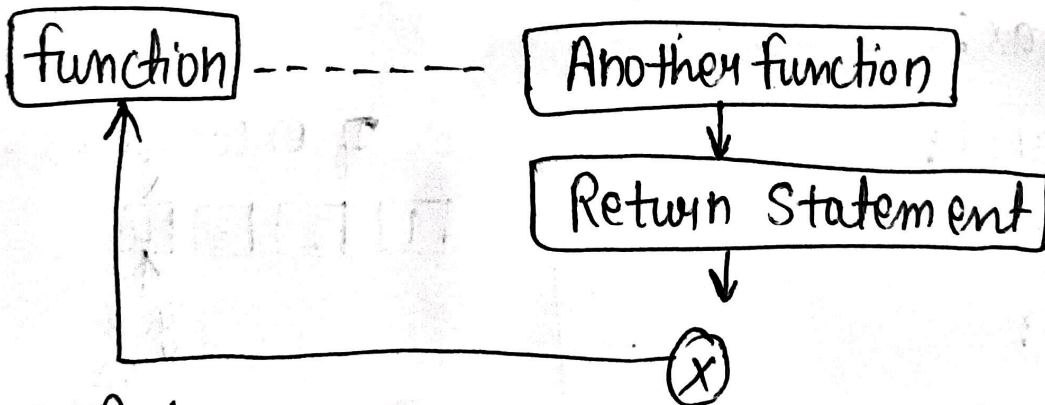
```
}
```

```
printf ("Value = %d \n", i);
```

```
}
```

```
}
```

### Return Statement in C



The Return Statement is a type of jump statement in C which is used in a function to end it or terminate it immediately with or without value and return the flow of program execution to the start from where it is called.

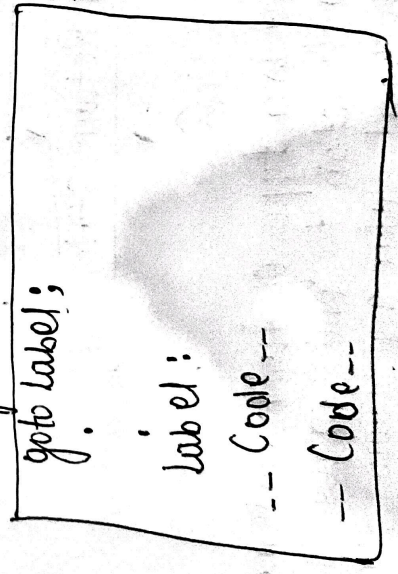
Return Statement Example

```
int Sum ()  
{  
    int a=0;  
    int num;  
    for (num = 0; num <= 10; num++) {  
        a += num;  
    }  
    return a;  
}  
void main()  
{  
    int a = Sum();  
    print f("Sum of digits from 1 to 10 = %d \n", a);  
    return ;  
}
```

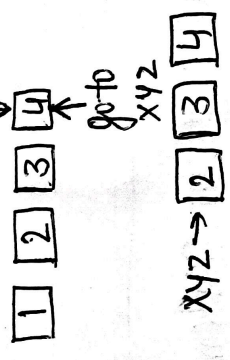
★ Goto Statement

Syntax of goto statement

Syntax:



If encountered first this





Goto statement is same what similar to the continue statement in the jump statement in C, but the continue statement can only be used in loops, whereas goto can be used anywhere in the program, but what the continue statement does it skips the current iteration of the loop and goes to the next situation. Iteration, but in the goto statement we can specify where the program control should go after skipping.

The concept of label is used in this statement to tell the program control where to go. The jump in the program that goto takes is within the same function. Below is the diagram; the label is XYZ,

\* Goto Statement Example

```

int main() {
    int a;
    printf("Enter a positive int: ");
    scanf("%d", &a);
    if (a % 2 == 0) // logic of even
        goto Even; // goto statement 1
    else
        goto odd; // goto statement 2
Even: // Label 1
    printf("Number is Even\n");
    exit(0);
Odd: // Label 2
    printf("Number is Odd\n");
    return 0;
}
}

```

## ★ ADVANTAGES of Jump Statement in C.

- You can control the Program flow or <sup>alter</sup> after the follow of Program.
- Skipping the unnecessary code can be done using jump statements
- You can decide when to break out the loop using break statements
- You get some sort of flexibility with your code using jump statements.

## ★ DISADVANTAGES of Jump Statement in C

- Readability of the code is disturbed because there are jump from one part of the code to another.
- Debugging becomes a little difficult.
- Modification of the code become difficult.