Jump Statements:

Break Statement:

- The keyword break allows us to jump out of a loop instantly, without waiting to get back to the conditional test.
- When a keyword break is encountered inside any loop, control automatically passes to the first statement after the loop.
- A break is ususally associated with an if.
- **Example:** Write a program to determine whether a number is prime or not.

```
main()
{
        int num,i;
        printf("Enter a number:");
        scanf("%d", &num);
        for(i=2;i<num;i++)
        {
                if(num%i = =0)
                {
                        printf("Not Prime Number");
                        break;
                }
        }
        if(i= =num)
                printf("Prime Number");
}
```

- **Output:**
```
Enter a number:17
Not Prime Number
```
- In case of nested loops the keyword **break**, breaks the control only from the inner loop in which it is placed.

Continue Statement:

- The keyword "**continue**" allows us to take the control to the beginning of the loop, bypassing the statements inside the loop, which has not been executed.
- When the keyword "**continue**" is encountered inside any loop, control automatically passes to the beginning of the loop.
- A "**continue**" is usually associated with an "if".
- **Example:**

```
main()
{
        int i,j;
        for(i=1;i<=2;i++)
        {
                for(j=1;j<=2;j++)
                {
                        if(i = = j)
                                continue;
                        printf("\n%d %d\n",i,j);
                }
        }
```

```
}
```

- **Output:** 1 2
  2 1

exit() Function:

- It is a standard library function which terminates the execution of the program.
- It terminates the entire program forcing a return to Operating System.
- **Syntax:** void exit(int return-code);
- **Example:** exit(0);
- Zero as the argument indicates normal program termination.