JUMP STATEMENTS

The break Statement

The *break* statement provides an early exit from *for*, *while*, and *do*, just as from *switch*. A *break* causes the innermost enclosing loop or switch to be exited immediately. When *break* is encountered inside any loop, control automatically passes to the first statement after the loop.

Consider the following example;

```
main( )
{
        int i = 1 , j = 1 ;
        while ( i++ <= 100 )
        {
                while ( j++ <= 200 )
                {
                if ( j == 150 )
                  break ;
                 else
                  printf ( "%d %d\n", i, j );
                }
        }
}
```

In this program when j equals 150, break takes the control outside the inner while only, since it is placed inside the inner *while*.

The *continue* Statement

The *continue* statement is related to *break*, but less often used; it causes the next iteration of the enclosing *for*, *while*, or *do* loop to begin. In the *while* and *do*, this means that the test part is executed immediately; in the *for*, control passes to the increment step. The *continue* statement applies only to loops, not to switch.

Consider the following program:

```
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 ) ;
                }
        }
}
```

The output of the above program would be...

1 2
2 1

Note that when the value of I equals that of j, the *continue* statement takes the control to the *for* loop (inner) by passing rest of the statements pending execution in the *for* loop (inner).

The *goto* statement

Kernighan and Ritchie refer to the *goto* statement as "infinitely abusable" and suggest that it "be used sparingly, if at all.
The *goto* statement causes your program to jump to a different location, rather than execute the next statement in sequence. The format of the *goto* statement is;

goto statement label;

Consider the following program fragment

```
if (size > 12)
        goto a;
goto b;

a: cost = cost * 1.05;
flag = 2;
```

*b: bill = cost * flag;*

Here, if the *if* conditions satisfies the program jumps to block labelled as *a:* if not then it jumps to block labelled as *b:*.

Exercise questions:

1. WAP to input the 3 sides of a triangle & print its corresponding type.
2. WAP to input the name of salesman & total sales made by him. Calculate & print the commission earned.

TOTAL SALES	RATE OF COMMISSION
1-1000	3 %
1001-4000	8 %
6001-6000	12 %
6001 and above	15 %

3. WAP to calculate the wages of a labor.

TIME	WAGE
First 10 hrs.	Rs 60
Next 6 hrs.	Rs 15
Next 4 hrs.	Rs 18
Above 10 hrs.	Rs 25

4. WAP to calculate the area of a triangle, circle, square or rectangle based on the user's choice.
5. WAP that will print various formulae & do calculations:
   i. Vol of a cube
   ii. Vol of a cuboid
   iii. Vol of a cyclinder
   iv. Vol of sphere
6. WAP to print the following series
   i. S = 1 + 1/2 + 1/3 ……..1/10
   ii. P= (1*2) + (2 *3) + (3* 4)+…….(8 *9) +(9 *10)
   iii. Q= ½ + ¾ +5/6 +….13/14
   iv. S = 2/5 + 5/9 + 8/13….n
   v. S = $x + x^2 + x^3 + x^4$.....+ $x^9 + x^{10}$
   vi. P= $x + x^3/3 + x^5/5 + x^7/7$…….n terms
   vii. S= (13 *1) + (12 * 2)……(1 *13)
   viii. S = $1 + 1/(2^2) + 1/(3^3) + 1/(4^4) + 1/(5^5)$

ix. S = 1/1! + 1/2! + 1/3! ……+1/n!

x. S = 1 + 1/3! + 1/5!+……..n terms

xi. S = 1 + (1+2) +(1+2+3) + (1+2+3+4)……+(1+2+3…….20)

xii. S= $x + x^2/2! + x^3/3! + x^4/4!.....+x^{10}/10!$

xiii. P = $x/2! + x^2/3! +…….x^9/10!$

xiv. S = 1 – 2 + 3 - 4……. + 9 – 10

xv. S = $1 -2^2 + 3^2 - 4^2……. +9^2 - 10^2$

xvi. S = 1/(1 + 2) + 3/(3 + 5)……15/(15 + 16)

xvii. S = $1 +x^2/2! – x^4/4! + x^6/6!....n$

xviii. S = 1 + (1 + 2) + (1+2+3)……..(1+2+3+4…..20)

xix. S = $1 + x + x^2/2 + x^3/3…….+x^n/n$

xx. S = 1 * 3/ 2 * 4 * 5 + 2 * 4 / 3 * 5 * 6 + 3 * 5/ 4 * 6 * 7……..n * (n+2)/ (n+1) * (n+3) * (n+4)

7. WAP to input a no & print its corresponding table.
8. WAP to print the table from 1 to 10 till 10 terms.
9. WAP to input a no & print its factorial.
10. WAP to input a no & check whether it is prime or not.
11. WAP to input a no & print all the prime nos upto it.
12. WAP to input a no & print if the no is perfect or not.
13. WAP to find the HCF of 2 nos.
14. WAP to print the Pythagoras triplets within 100. (A Pythagorean triplet consists of three positive integers a, b, and c, such that $a^2 + b^2 = c^2$).
15. WAP to input a no & check whether its automorphic or not. (An automorphic number is a number whose square "ends" in the same digits as the number itself. For example, 5^2 = 2**5**, 6^2 = 3**6**, 76^2 = 5776, and 890625^2 = 793212**890625**, so 5, 6, 76 and 890625 are all automorphic numbers).
16. WAP to convert a given no of days into years, weeks & days.
17. WAP to input a no & check whether it's an Armstrong no or not. (An Armstrong no is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since $3^3 + 7^3 + 1^3$ = 371).
18. A cricket kit supplier in Jalandhar sells bats, wickets & balls. WAP to generate sales bill. Input form the console the date of purchase, name of the buyer, price of each item & quantity of each item. Calculate the total sale amount & add 17.5 % sales tax if the total sales amount >300000 & add 12.5 % if the total sales amount is >150000 & 7 % otherwise. Display the total sales amount, the sales tax & the grand total.
19. WAP to check whether a given number is magic number or not.

    (What is a magic number? Example: 1729

    - Find the sum of digits of the given number.(1 + 7 + 2 + 9 => 19)
    - Reverse of digit sum output. Reverse of 19 is 91

- Find the product of digit sum and the reverse of digit sum.(19 X 91 = 1729)
- If the product value and the given input are same, then the given number is a magic number.(19 X 91 <=> 1729)

20. Write a C program to calculate generic root of the given number. (To find the generic root of a no we first find the sum of digits of the no until we get a single digit output. That resultant no is called the generic no. Eg: 456791: 4+5+6+7+9+1=32. 3 +2 =5. So, 5 becomes the generic root of the given no)