FORMATTED INPUT/OUTPUT FUNCTIONS

scanf() Function

scanf() function is used to read/input values of variables using the standard input device (keyb
It has the following form.

```
scanf("format string", &V1, &V2...&Vn);
```

where V1, V2...Vn are variables whose values are to be read from the keyboard.

"format string" is the control string which represents the format specification (refer Tabl

The symbol & (ampersand) represents the memory address where the variable value is to be st

Example

1. `scanf("%d %d", &a, &b);` to read value of **int** variables a and b.
2. `scanf("%f %d", &x,&n);` to read **float** value of x and **int** value of n.
3. `scanf("%c", &sex);` to read **char** value for variable sex.
4. `scanf("%s", sname);` to read a string of **char** variable sname.

When these functions are executed, the computer will wait for the values of the variable listed **scanf** which are to be entered using the keyboard.

Table 1 *Format Specifiers*

Data Type	Format	Meaning
int	%d	Represents a decimal integer value
	%u	Represents an unsigned integer value
	%o	Represents an unsigned octal value
	%x	Represents an unsigned hexadecimal value
float/double	%f	Represents a floating point value
	%e	Represents a floating point value in decimal or exponential form
char	%c	Represents a single character value
	%s	Represents a string of value of characters

Note: **%hd, %ld, %lf** are used for **short int, long int** and **long double** respectively.

printf() Function

prinf() function is used to print/display values of variables using the standard output device (monitor). It has the following form.

```
printf("format string", V1, V2 ... Vn);
```

where V1, V2 ... Vn are variables whose values are to be displayed in the monitor. "format string" is the control string which represents the format specification (refer Table 1)

Example

1. `printf(" %f ", s);`
2. `printf("\n sum = %6.2f ", s);`
3. `printf("\n %d factorial is %d", k, kfact);`

When these functions are executed, the computer prints/displays the values of the variables listed in **printf()**. Consider values s = 2.8, k = 5 and kfact = 120.

```
2.800000
sum = 2.80
5 factorial is 120
```

Note that **\n** used in **printf** is an escape sequence (refer Table 2) to print the results in the new line.

%6.2f represents the formatted print with total width 6 and number of decimal digits 2. Imagine the following.

		2	.	8	0

It is recommended to use a formatted print with required number of decimal digits for **float** type variables to display the results in decimal form.

ESCAPE SEQUENCES

Escape sequences are control characters used to move the cursor and print characters such as ? , " , \ and so on.

Table 2 *Escape Sequences*

Character Constant	Meaning
\a	Audible bell (alert)
\b	Backspace
\f	Form feed

Character Constant	Meaning
\n	Move to new line
\r	Carriage return (Enter)
\t	Horizontal tab
\v	Vertical tab
\\	Print back slash
\?	Print question mark
\'	Print single quote
\"	Print double quote
\0	Null character

Note that escape sequences "\n" and "\0" are commonly used.

ASSIGNMENT STATEMENT

An assignment statement is used to assign value to a variable. It has the following form.

```
Variable = Value or expression;
```

Example

```
m = 25;
y = x*x+4*x-6;
s = s + x;
```

MULTIPLE ASSIGNMENT STATEMENT

A multiple assignment statement is used to assign a value to more than one variable. It has the following form.

```
V1 = V2 ... = Vn = Value or expression;
```

where V1, V2 ··· Vn are variables.

Example

```
m = n = 3;
np = nn = nz = 0;
x = y = (a * a + b * b)/2;
```

Example 1

Write a C program to find the sum and product of given two numbers.

Solution

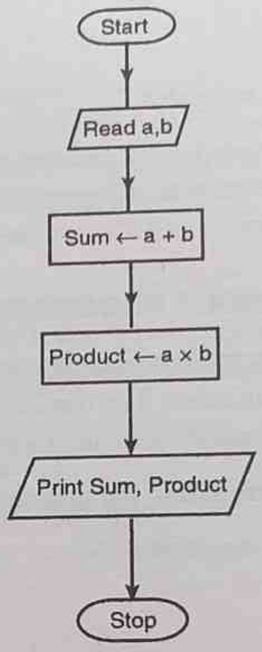
Two numbers, say A and B, are given (input) and the result sum (A+B) and product (A*B) of two numbers are to be calculated.

It is recommended to use the flowchart or algorithm to write the correct sequence of instructions in the program.

Algorithm

1. Read a, b
2. Sum ← a + b
3. Product ← a * b
4. Print Sum, Product
5. Stop

Flowchart




```
/* program to find sum and product */
#include <stdio.h>
main()
{ int a,b,sum,product;
  scanf("%d %d",&a,&b);
  sum = a+b;
  product = a*b;
  printf("%d %d",sum,product);
}
```

When this program is executed, the computer will wait for the values of variables a and b, which has to be entered by the user.

Now the computer calculates the sum and product of the two numbers and the results are displayed.

Example 2

Write a C program to convert temperature in °F to °C using the formula $°C = \frac{5}{9}(°F-32)$

Solution

In this problem, the input variable is F which represents temperature in °F and the output variable C which represents temperature in °C.

```
/* program to convert temperature from degree F to degree C */
#include <stdio.h>
#include <conio.h>
main()
{ float f,c;
  clrscr();
  printf("\n Degree fahrenheit ? ");
  scanf("%f",&f);
  c = 5.0/9.0 *(f-32);
  printf("\n Degree centigrade = %6.2f",c);
  getch();
}
```

Note that 5.0/9.0 will produce the real value 0.555556, but 5/9 will produce the integer value zero which leads to a wrong answer.

The variable F and C are declared **float** as it is understood that the result in this problem is always a floating point value.

When this program is executed, the input value is entered and the result is displayed as follows:

Input value → Degree fahrenheit ? 105 ↵

Output value → Degree centigrade = 40.56

Example 3

Write a C program to find the area of a triangle using three sides.

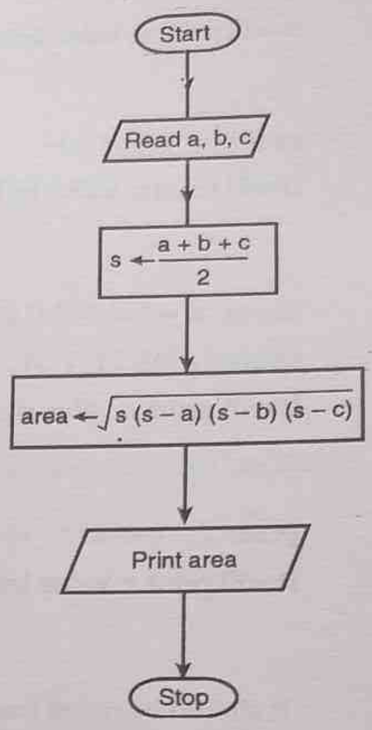
Solution

We know that area of a triangle = $\sqrt{s(s-a)(s-b)(s-c)}$

where a, b and c are the three sides of the triangle and $s = \frac{a+b+c}{2}$

```
/* program to find area of a triangle */
#include <stdio.h>
#include <math.h>
#include <conio.h>
main()
{   float a,b,c,s,area;
    clrscr();
    printf("\n Enter three sides : ");
    scanf("%f,%f,%f",&a,&b,&c);
    s = (a + b + c)/2;
    area = sqrt(s*(s - a)*(s - b)*(s - c));
    printf("\n Area of triangle = %6.2f Sq.units",area);
    getch();
}
```

Flowchart



Note that **<math.h>** is included to link the library function **sqrt()** with the program.

When this program is executed, the input values are entered and output is obtained as follows.

```
Enter three sides : 5,4,6 ↵

Area of triangle = 9.92 Sq.units
```

Note that the input values are separated by comma (,) which is given using **scanf()** format string.

Caution: Values entered for sides a, b and c should form a triangle. For example, 5, 1 and 3 for three sides will never make a triangle.

MORE ABOUT FORMATTED OUTPUT FUNCTIONS

Output values can be printed with specific width using commands for formatted output. Consider the following examples.

```
int n = 28 ;
printf("%5d",n);
```

It will print n value right justified with three preceding spaces.

			2	8

SHORT QUESTIONS AND ANSWERS

1. What is the purpose of `scanf()` function? How it is used within a C program?

   A `scanf()` function is used to read values of variables from the keyboard. It is used with the variable list and its format string in parentheses.

   Example

   ```
   scanf("%f %d", &x, &n);
   ```

2. What is the use of `"\n"`?

   `"\n"` is used to generate or move the cursor control to a new line.

3. Assuming i = 3, give the output for

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

   new line generated

   5 5 3

   new line generated

   Note that the evaluation takes place from right to left as per the operator precedence and associativity. Then the values are printed.

4. Explain the significance of the following control specifiers:

   (a) `%c` (b) `%o` (c) `%x` (d) `%d`

   `%c` is used to read/print a character value.

   `%o` is used to read/print an unsigned octal number.

%x is used to read/print an unsigned hexadecimal number.

%d is used to read/print a signed integer value.

5. What is a header file in C? List any two header files.

   A C program is written using many functions which are available under many header files. The required header file has to be included whenever such functions are used in the program.

   Example

   ```
   #include <math.h>
   #include <conio.h>
   ```

6. The line from which a program execution begins is _____.

   main()

7. The sign # of the compiler directive must appear at the _____ of a line.

   beginning

8. Write four different C statements each adding 1 to integer variable x.

   ```
   x = x + 1;
   x += 1;
   x++;
   ++x;
   ```

9. How do you generate an alarm/beep using **printf**?

   `printf("\a");` generates an alarm/beep sound.

10. What is the difference between **getchar()** and **getche()** ?

    getchar() and **getche()** are used to read a character from the keyboard. **getchar()** can be used to read characters one at a time which can be edited on the screen. As soon as the enter key is pressed the characters are assigned to a **char** array. But **getche()** will read a character without expecting the enter key press.

11. What is an escape sequence? What is its purpose?

    An escape sequence is a control character used to move the cursor and print a character (e.g. \ / and so on) using **printf()** function.

    For example,

    ```
    printf("\n\" %c\"", ch);
    ```

    may display the output in a new line like "m"

12. Explain the need for the following:

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

    The header file **<stdio.h>** is used to include and link standard input output functions in program.

The header file <math.h> is used to include and link mathematical functions like sqrt(), fabs() and so on in a C program.

13. Explain with examples the syntax of scanf() and printf() functions.

scanf() function is used to read values of variables from the keyboard. It has the following form.

```
scanf("format string", &V1, &V2 ··· &Vn);
```

where V1, V2 ··· Vn are variables.

Example

```
scanf("%f %d", &x, &n);
```

prinf() function is used to display/print values of variables in the monitor. It has the following form.

```
printf("format string", V1,V2 ··· Vn);
```

where V1, V2, ··· Vn are variables.

Example

```
printf("%d factorial is %d", k, kfact);
```

14. What is the difference between C character and C string ?

A C character can hold a single character which is enclosed in single quote (' ').

Example

```
char ch = 'm';
```

A C character string can hold a string of characters which is enclosed in double quotes (" ").

Example

```
char st[20] = "COMPUTER WORLD";
```

15. Name any four functions available in stdio.h.

scanf(), printf(), getchar(), putchar()

16. The file name stdio.h is an abbreviation for _____ file.

standard input output header

17. What is the difference between 'a' and "a" in C ?

'a' is a single character constant but "a" is a string of characters constant. The computer assigns a null character "\0" to the end of this constant.

18. How can the minimum field width for a data item be specified using the printf() function?

The minimum field width for a data item is given within the format string in printf() function. Consider the following example to print the values right justified along the specified width.

```
int rno = 20346;
char sname[15] = "SACHIN";
float tot = 78.5;
printf("%5d%15s%6.2f", rno, sname, tot);
```

Example

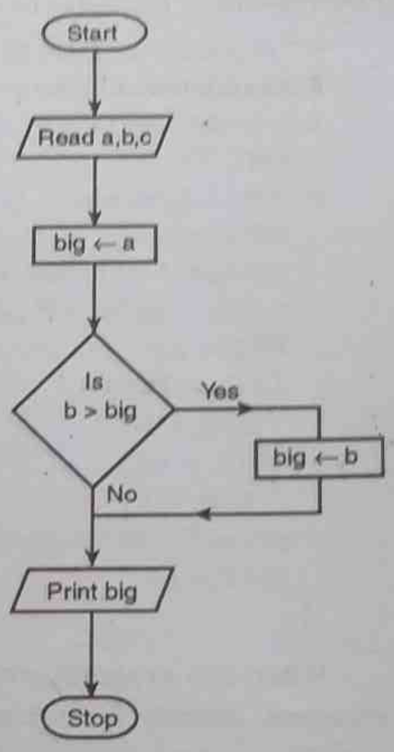
```
if ( a > b )
    big = a;
else
    big = b;
```

Example 1

Write a C program to find the biggest of given two numbers.

Solution

There are many logics (methods) available to obtain the biggest of the two numbers. Consider the following method which uses a variable big to store the biggest value. This method can be easily extended for any number of variables (say n) by considering a loop.




```
/* Program to find biggest of two numbers */
#include <stdio.h>
#include <conio.h>
main()
{ int a,b,big;
  clrscr();
  printf("\n Enter two numbers : ");
  scanf("%d %d",&a,&b);
  big = a;
  if(b > big)
    big = b;
  printf("\n Biggest number is %d",big);
  getch();
}
```

When this program is executed, the user has to enter the two numbers. Initially a is assumed as big. It is then compared with b to obtain the biggest number.

```
Enter two numbers : 5 8
Biggest number is 8
```

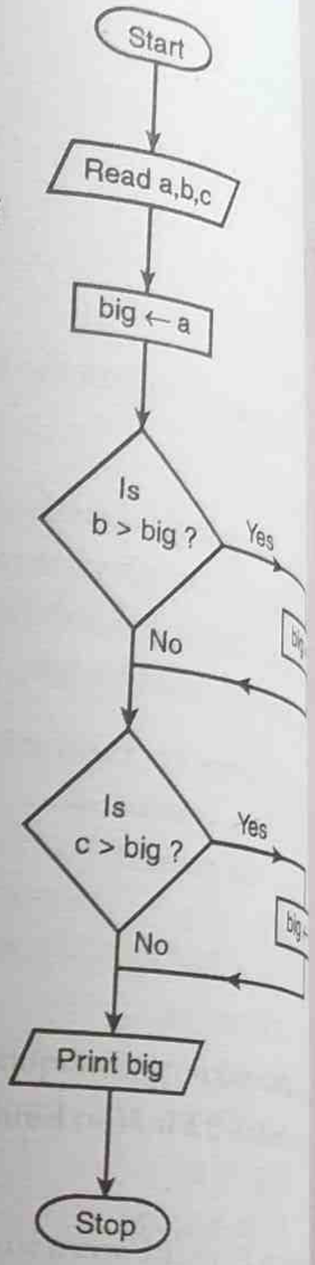
Example 2

Write a C program to find the biggest of given three numbers.

Solution

The method discussed in Example 1 is extended to find the biggest of three numbers.

```
/* Program to find biggest of three Numbers*/
#include <stdio.h>
#include <conio.h>
main()
{ int a,b,c,big;
  clrscr();
  printf("\n Enter three numbers : ");
  scanf("%d %d %d",&a,&b,&c);
  big = a;
  if(b > big)
    big = b;
  if(c > big)
    big = c;
  printf("\n Biggest number is %d",big);
  getch();
}
```



When this program is executed, the user has to enter the three numbers. Initially a is assumed as big, then it is compared with b to get biggest of a and b. Finally c is compared with existing big to get the biggest number.

```
Enter three numbers: 5 13 8
Biggest number is 13
```

NESTED if-else STATEMENT

An **if** statement may have another **if** statement in the **<true block>** and **<false block>**. This compound statement is called nested **if** statement. Any number of **if** statements can be nested. It has the following form.

Example 4

Write a C program to find the value of y using

$$y(x,n) = \begin{cases} 1+x & \text{when } n = 1 \\ 1+x/n & \text{when } n = 2 \\ 1+x^n & \text{when } n = 3 \\ 1+nx & \text{when } n > 3 \text{ or } n < 1 \end{cases}$$

Solution

The given conditions are considered and a nested **if** statement may be written to use the required relation to find the value of y.

```
/* Program to find value of Y */
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{  int n;
   float x,y;
   clrscr();
   printf("\n Enter value to x and n : ");
   scanf("%f %d",&x,&n);
   if(n == 1)
      y = 1 + x;
   else
      if(n == 2)
         y = 1 + x/n;
      else
         if(n == 3)
            y = 1 + pow(x,n);
         else
            y = 1 + n*x;
   printf("\n Value of y(x,n) = %6.2f",y);
   getch();
}
```

When this program is executed, the user has to enter the values of x and n. The value of y is obtained by considering the nested **if** statement and output is printed as follows.

statement.

Example 5 Write a C program to find the value of y using

$$y(x,n) = \begin{cases} 1+x & \text{when } n = 1 \\ 1+x/n & \text{when } n = 2 \\ 1+x^n & \text{when } n = 3 \\ 1+nx & \text{when } n > 3 \text{ or } n < 1 \end{cases}$$

Solution

This problem is similar to Example 4 except that **switch** statement is used here in the place of **if** statement.

```
/* Program to find value of Y  */
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{
   int n;
   float x,y;
   clrscr();
   printf("\n Enter value to x and n : ");
   scanf("%f %d",&x,&n);
   switch(n)
   {
      case 1 :  y = 1 + x;
                break;
      case 2 :  y = 1 + x/n;
                break;
      case 3 :  y = 1 + pow(x,n);
                break;
      default : y = 1 + n*x;
                break;
   }
   printf("\n Value of y(x,n) = %6.2f",y);
   getch();
}
```

When this program is executed, the user has to enter the values of x and n. The value obtained in the **switch** statement and the result is printed as follows.

```
Enter value to x and n : 0.42 5
Value of y(x,n) = 3.10
```

Example 7

Write a C program to find the roots of a quadratic equation $ax^2 + bx + c = 0$ for all possible combinatio of a, b and c.

Solution

A quadratic equation will have two roots which are obtained using the following expression.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where $b^2 - 4ac$ is called discriminant.

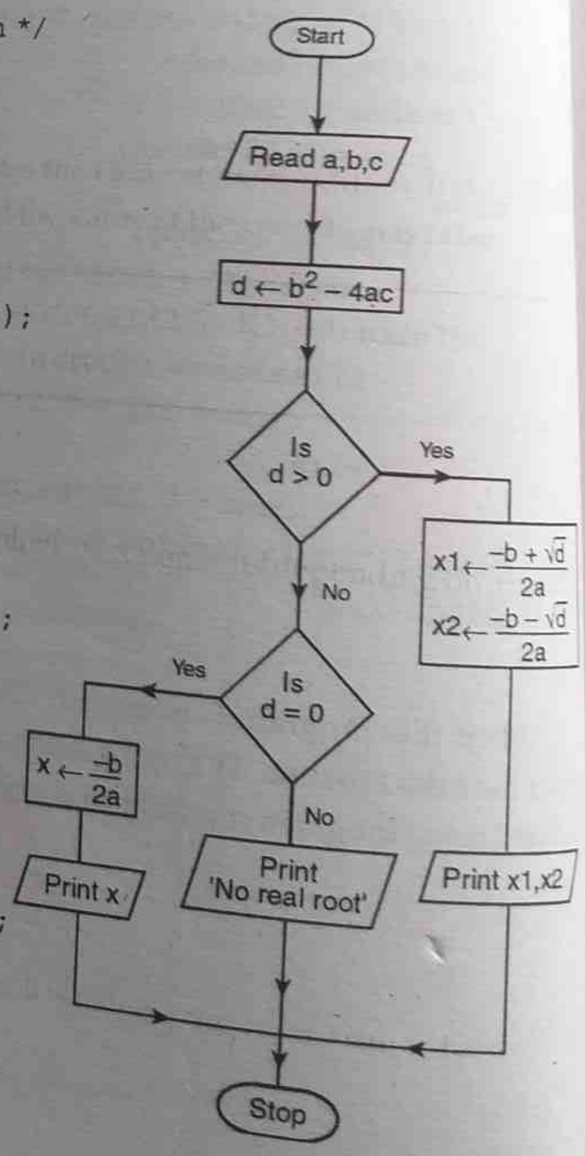
Note that when $b^2 - 4ac > 0$, the roots are real and unequal

$b^2 - 4ac = 0$, the roots are real and equal, i.e. $x = \frac{-b}{2a}$

$b2 - 4ac < 0$, the roots are imaginary, i.e. $x = \frac{-b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a} i$

Also note that the value of the coefficient $a \neq 0$.

```
/* Program to find roots of a Quadratic Equation */
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{ float a,b,c,d,x1,x2,x;
  clrscr();
  printf("\n Enter coefficients a,b and c : ");
  scanf("%f %f %f",&a,&b,&c);
  d = b*b - 4*a*c;
  if(d > 0)
  {
    x1 = (-b + sqrt(d))/(2*a);
    x2 = (-b - sqrt(d))/(2*a);
    printf("\n Roots are real and unequal");
    printf("\n %6.2f  %6.2f",x1,x2);
  }
else
    if(d == 0)
    {
      x = -b/(2*a);
      printf("\n Roots are real and equal");
      printf("\n %6.2f",x);
    }
    else
```



```
        printf("\n No Real roots,roots are complex");
    getch();
}
```

When this program is executed, th