Chapter # 10
Input and Output

Q. Define input and output. What do we mean by standard input and output? How is it performed in C?

Input

The process of giving something to the computer is known as **input**. The input is mostly given by keyboard. The term **standard input** refers to the input using keyboard. A program may need certain inputs from the user for working properly. C language provides many functions to get input from the user.

Some important functions for input are as follows:

- scanf()
- getch()
- gets()
- getche()

Output

The process of getting something from the computer is known as **output**. The output is mostly displayed on monitor. The term **standard output** refers to the output displayed on the monitor. The result of a program is the output of that program. C language provides many functions to display output to the user. Some important functions for output are as follows:

- printf()
- puts()

The functions used for input and output are stored in the header file **stdio.h**. If a program uses any input or output function, it is necessary to include this header file in the program.

Q. Define and explain "printf" function with examples.

printf Function

The **printf** function is used to display output on the monitor. It can display text, constants or values of variables on screen in specified format. It is most commonly used to display output in C language. The function **printf** is pronounced as **print-eff**.

Syntax

The syntax of **printf** function is as follows:

```
printf(Format String, argument_list);
```

Format String

The format string is given in double quotes. It is also called **control string**.

Format string may consist of the following:

Text	Any message to display on the screen
Format specifiers	specify how the values of variables will be displayed
Escape sequences	specify the format of output

Argument List

The argument list consists of constants, variables or expressions whose values are to be printed on the screen. Each argument in the list is separated by comma. The value of the argument is printed according to the corresponding format specifier given in format string. Different format specifiers are used for different types of values. The use of argument list is optional.

Using Format String without Argument List

The following example displays a simple message using printf function:

```
printf("Hello World");
```

Control String

In the above example, printf function contains only format string. There is no argument list, format specifier or escape sequence. It will display the message "Hello World" on the screen.

Using Format String with Single Argument

The following example uses an integer variable as argument and displays its value.

Format Specifier

```
printf("Your marks are %d", m);
```

Control String

In the above example, format string contains %d. It is related to m. It indicates that the value of m will be displayed as integer.

Using Format String with Multiple Arguments

Many variables can be used in one printf function as follows:

```
printf("Your marks are %d and grade is %c.", m, g);
```

Control String

The above example uses two format specifiers %d and %c. %d is related to m and %c is related to g. %d is used with integers and %c with characters. The sign % indicates the beginning of a format specifier.

Program 10.1

Write a program that displays a message and values of integer and character variables.

```
#include <stdio.h>
#include <conio.h>
void main()
{
 int n = 10;
 char ch = '*';
 clrscr();
 printf("Testing output...");
 printf("%d", n);
 printf("%c", ch);
 getch();
}
```

Output:
Testing output...10*

How above Program Works

The above program declares and initializes two variables n and ch. It displays message "Testing output..." on the screen. It then displays the values of variables n and ch. The output appears on the same line. The output of second statement starts where the output of first statement ends.

Program 10.2

Write a program that adds two floating point numbers and shows the sum on screen.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  float var1, var2, res;
  clrscr();
  var1 = 24.27;
  var2 = 41.50;
  res = var1 + var2;
  printf("%f + %f = %f", var1, var2, res);
  getch();
}
```

Output:
24.27 + 41.50 = 65.770004

Program 10.3

Write a program to calculate and print the area of a square with given height & width.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int height, width, area;
  clrscr();
  height = 5;
  width = 4;
```

Output:
Area of Square = 20

```
    area = height * width;
    printf("Area of Square = %d", area);
    getch();
}
```

Q. What is format specifier? Discuss different format specifiers in C language.

Format Specifier

Format specifier is used to specify the format according to which values will be read and displayed. It determines the following things:

- Data type of Variable
- Field Width
- Format of the value

Format specifier is started with the symbol %. Different format specifiers are used for different types of variables. These format specifiers are as follows:

- Integer format specifier
- Character format specifier
- Floating-point format specifier

Integer Format Specifier

Different format specifiers for integer values are as follows:

Format Specifier	Type
%d	Used for signed decimal integer value
%I	Used for signed integer value
%o	Used for unsigned octal value
%u	Used for Unsigned decimal integer value
%x	Used for unsigned hexadecimal values with lower case like a, b, c etc.
%x	Used for unsigned hexadecimal values with upper case like A, B, C etc.

Table 10.1: Conversion characters for integer values

Floating-Point Format Specifier

Different format specifiers for floating-point values are as follows:

Format Specifier	Type
%f	Used for signed float or double value
%e	Used for signed float or double in exponential notation
%g	Used for large floating point in exponential notation with no extra zeros

Table 10.2: Conversion characters for floating-point values

Character Format Specifier

Different format specifiers for character values are as follows:

Format Specifier	Type
%c	Used for character value
%s	Used for strings

Table 10.3: Conversion characters for character values

Q. How integer values are printed in C language?

The format specifier "%d" is used to display integer values in **printf** function. Suppose the value of integer variable **m** is 100. The following statement:

```
printf("Your marks are %d", m);
```

will display the following line on the screen:

```
Your marks are 100
```

Q. How floating point values are printed in C language?

The format specifier "%f" is used to display floating point values in **printf** function. Suppose the value of float variable **a** is 70.55.

The following statement

```
printf("Your average is %f", a);
```

will display the following line on the screen:

```
Your average is 70.550000
```

Q. Explain the use of field width specifiers in C language.

The number of columns used to display a value on the screen is called field width. A field width specifier describes the number of columns that should be used to print a value. It determines how the value will be displayed on the screen.

Syntax

The general syntax of using field width specifiers is as follows:

```
flag width.precision
```

Some values of flag are as follows:

Flag	Description
-	The output will be left-justified
+	Always displays sign with value
space	Display space if there is no sign

Table 10.4: Format Modifiers

Width It specifies total number of columns used to display value.

Precision It indicates number of columns used after decimal point.

Specifying Field Width

Field width specifier is used to specify number of columns in which the value is printed. It includes a positive value after % symbol that indicates the field width.

Some examples of field width are as follows:

- %6d specifies an integer for a field of width 6.
- %10c specifies a character for a field of width 10
- %8f specifies a floating point value in a field of width 8.

If the specified field width is not enough to print the value, the additional space is automatically used. If the value is less than the specified width in the format specifier, the value is printed right adjusted in the field.

Q. Explain the use of field width specifiers for integers in C language.

Specifying Field Width for Integers

The %d format specifier is used to display integer value in **printf** function. A value can be written between % and d in "%d" format specified. The value specifies the field-width or the number of columns to be used for displaying the value. The following statement indicates that four columns will be used to display the value of area:

```
printf("Area = %4d", area);
```

Suppose the value of area is 25. In this case, two spaces will be added before 25 as follows:

```
Area = □□25
```

The symbol □ represents a blank space in the above line. The value will appear right-justified and two spaces will appear before the actual value.

Column numbers →	1	2	3	4
Value →			2	5

Figure 10.1: Integer value with field width specifier

Specifying Left or Right Justified Output

The symbol – is used with % sign to display the output left-justified. The statement **printf("%-10d", n);** will display the value of **n** left justified in ten character space. If the value of **n** is 225, it will be displayed as follows:

Column numbers →	1	2	3	4	5	6	7	8	9	10
Value →	2	2	5							

Figure 10.2: Left-justified output with 10 field width

The following table shows how integer values are displayed using different format specifiers:

Value	Format specifier	Output	Value	Fromat specifier	Output
786	%4d	□786	-786	%4d	-786
786	%5d	□□786	-786	%5d	□-786
786	%6d	□□□786	-786	%6d	□□-786
786	%1d	786	-786	%2d	-786

Q. Explain the use of field width specifiers for floating point numbers in C language.

Specifying Width & Precision for Floating Point Values

The user can also specify both field width and number of decimal places for floating point numbers. The format specifier "%f" is used to display floating point values in printf function. A value can be written between % and f in "%f" format specified. The total field width should be large enough to display all digits before and after the decimal point.

For example, the total field width to display 15.245 should be six. A zero is also printed before decimal point for a number that is smaller than zero. For example, if the value is .5, it will appear as 0.5. The total field width should include a space for decimal point and for minus sing if the number is negative.

Syntax

The general format for the format specifier for floating point is as follows:

%m.nf

%	It specifies the beginning of a format specifier.
m	It represents the total field width including decimal point and fractional part.
n	It represents the desired number of decimal places.

Example

The statement will display a **float** value using space of eight characters.

```
printf("%8.2f", n);
```

The space of two columns is used for the digits after decimal point. The value of n will be rounded off to two decimal places. The second digit of fractional part is increased by 1 if the third digit of fractional part is 5 or more. Otherwise, the third is discarded. If the value needs more space than the width specifies, the additional space is automatically used.

Suppose the value of n is 20.346. In this case, the above statement will display 20.35. If the value of n is 20.3444, the statement will display 20.34.

The following table shows how floating point values are displayed using different format specifiers:

Value	Format specifier	Output	Value	Format specifier	Output
-25.41	%6.2f	□25.41	.123	%6.2f	□□0.12
3.14159	%5.2f	□3.14	3.14159	%4.2f	3.14
3.14159	%3.2f	3.14	3.14159	%5.1f	□□3.1
3.14159	%5.3f	3.142	3.14159	%8.5f	□3.14159
.6789	%4.2f	0.69	-0.007	%4.2f	-0.01
-.007	%8.3f	□□-0.007	-0.007	%8.5f	-0.00700
-.007	%.3f	-0.007	-3.14159	%.4f	-3.1416

Program 10.4

Write a program that displays average marks using field width of 5 characters and precision of 2 characters.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  float avg = 70.52;
  clrscr();
  printf("Average marks are: %5.2f", avg);
  getch();
}
```

Output:
Average marks are: 75.52

Program 10.5

Write a program that displays floating point value 152.3333 using field width of 10 characters and precision of 3 characters justified to left side.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  float v = 152.3333;
  printf("Value is: %-10.3f", v);
  getch();
}
```

Output:
Value is: 152.333

Q. How can you print character values in C language?

The format specifier %c is used to display character type values in printf function. Suppose the value of character variable g is 'A'. The following statement:

```
printf("Your grade is %c", g);
```

will display the following line on the screen:

```
You grade is A
```

In the above example, %c is replaced by the value of g in the output.

Program 10.6

Write a program that displays value of character variable grade.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char grade = 'A';
  clrscr();
  printf("Grade is: %c", grade);
  getch();
}
```

Output:
Grade is: A

Q. How string value is displayed in C language?

String variables are displayed using %s format specifier in printf function. Suppose the value of variable name is "Pakistan". The following statement will display this value on the screen:

```
printf("%s", name);
```

The above statement tells the computer to display the contents of name. The computer displays all characters one by one stored in string variable until null character is found. The null character indicates that the string has ended.

Program 10.7

Write a program that displays the value stored in a string variable name.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char name[ ] = "ICS";
  clrscr();
  printf("Name is %s ",name);
  getch();
}
```

Output:
Name is ICS

Program 10.8

Write a program that displays the name of a person right-justified and address left-justified by using 25 character spaces for each.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char name[ ] = "Usman";
  char address[ ] = "Faisalabad.";
  clrscr();
  printf("%25s \n",name);
  printf("%-25s ",address);
  getch();
}
```

Output:
Usman
Faisalabad

Q. What are escape sequences? Discuss different escape sequences in C.

Escape sequences are special characters used in format string to modify the format of output. These characters are not displayed in the output. These characters always begin with backslash "\". The backslash is known as **escape character**.

Different escape sequences used in C language are as follows:

Escape Sequence	Purpose
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Tab
\'	Single quote
\"	Double quote
\xdd	ASCII code in hexadecimal notation. Each d represents a digit.
\ddd	ASCII code in octal notation. Each d represents a digit.

This escape sequence is used to insert backspace in the output. For example:

printf ("Hello\bWorld"); will display
HellWorld

First of all, "Hello" is printed but **\b** deletes "o". Then "World" is printed. So "HellWorld" is displayed on the screen.

\f

This escape sequence is used to insert a blank paper in the printed output. It is called **form feed**. For example:

printf ("Hello\fWorld"); will display
HelloWorld

After printing "Hello", the computer will include a blank paper and then print the remaining output. It is used during printing.

\n

This escape sequence is used to insert new line in output. For example:

printf ("Hello\nWorld"); will display
Hello
World

First of all, "Hello" is printed and "\n" shifts the cursor to next line. Then "World" is printed. The output is displayed on two lines.

\r

This escape sequence is used to move the cursor at the beginning of current line. For example:

printf ("Hello\rWorld"); will display
World

First of all, "Hello" is printed but \r moves the cursor at the beginning of current line. Then "World" is printed that overwrites "Hello". So only "World" is displayed on the screen.

\t

This escape sequence is used to insert a TAB in the output. For example:

printf ("Hello\tWorld"); will display
Hello World

First of all, "Hello" is printed and **\t** inserts a TAB. Then "World" is printed. So "Hello World" is displayed on the screen.

\'

This escape sequence is used to display single quotes in the output. For example:

printf (" \'Hello World\' "); will display
'Hello World'

This escape sequence is used to display double quotes in the output. For example:

```
printf (" \"Hello World\" "); will display
"Hello World"
```

\\ Output by sigle slash (/) aey ge.

This escape sequence is used to display backslash in the output. For example:

```
printf ("C:\\ "); will display
C:\
```

Program 10.9

Write a program to display the following output using single printf statement.

```
*
* *
* * *
* * * *
```

```
#include <stdio.h>
#include <conio.h>
void main()
{
  clrscr();
  printf("*\n**\n***\n****");
  getch();
}
```

Program 10.10

Write a program to show following output using one printf statement:

```
1    2    3    4    5
6    7    8    9    10
```

```
#include <stdio.h>
#include <conio.h>
void main()
{
  clrscr();
  printf("1 \t 2 \t 3 \t 4 \t 5 \n 6 \t 7 \t 8 \t 9 \t 10");
  getch();
}
```

Program 10.11

Write a program to show following output using one printf statement:

```
Pakistan    is    my    country
Islamabad   is    its   capital.
```

```
#include <stdio.h>
#include <conio.h>
void main()
{
  clrscr();
  printf(" Pakistan\tis\tmy\tcountry\n Islamabad\tis\tits\tcapital.");
  getch();
}
```

Q. Define and explain "scanf" function with examples.

scanf Function

scanf function is used to get input from the user. The input is stored in a variable in a specified form.

Syntax

The syntax of scanf function is as follows:

```
scanf(format string, &variable1, &variable2, &variable3 ...);
```

Format String The format string is given in double quotes. It may consist of different format Specifiers that indicate the format according to which values will be entered.

Variables The variables are used to store the value entered by the user through keyboard. Each variable refers to a memory location. The symbol & indicates the memory location of variable in which the input is stored. It is called address operator.

Getting Single Input

The following example inputs a single value:

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

The above statement is used to get an integer input. The value entered by the user will be stored in a variable n.

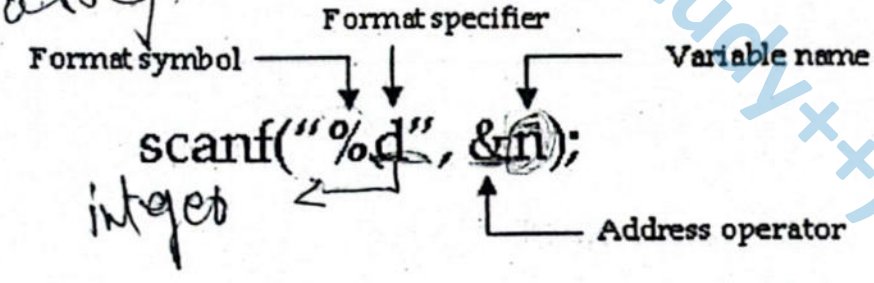

Figure 10.3: Scanf function

Getting Multiple Inputs

scanf function can be used to input many values at the same time as follows:

```
scanf("%d %d", &n, &m);
```

The above statement will get two inputs in n and m respectively. It is also possible to get different types of inputs in the same function as follows:

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

The above statement will get two inputs in n and avg. The first input is of type integer and the second is of type float.

The scanf function is normally used in combination with printf function. The purpose of using printf function before scanf function is to tell the user that computer is waiting for input. The following example explains it:

```
printf("Enter your marks: ");
scanf("%d", &marks);
```

In the above example, the first line displays message on the screen to tell the user about the input. It makes program easier to use by the user.

Program

Write a program to convert distance from kilometers into meters.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  float k;
  double m;
  clrscr();
  printf("Enter distance in kilometers:");
  scanf("%f", &k);
  m = k * 1000;
  printf("\n%f kilometers = %f meters", k, m);
  getch();
}
```

Output:

```
Enter distance in kilometers: 40.5
40.500000 kilometers = 40500.000000 meters
```

Working of the above Program

The above program declares two variables m and k. It displays a message to the user and gets input of distance in kilometers. The symbol & is used with variable k. The symbol refers to the address of the variable. The value entered by the user is stored in k. The program calculates the number of meters and displays the result on the screen.

Program 10.13

Write a program that inputs name, age and address from the user and displays it on the screen.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char name[25],address[30];
  int age;
  clrscr();
  printf("Enter your age:");
  scanf("%d", &age);
  printf("Enter your name:");
  scanf("%s", name);
```

Output:

```
Enter your age: 10
Enter your name: Usman
Enter your address: Faisalabad
Your name is Usman
Your address is Faisalabad
Your age is 10
```

```
    printf("Enter your address:");
    scanf("%s", address);
    printf("Your name is %s \n", name);
    printf("Your address is %s \n", address);
    printf("Your age is %d \n", age);
    getch();
}
```

Program 10.14

Write a program that inputs base and height from the user and calculates area of a triangle by using the formula Area = ½ * Base * Height.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float base, height;
    double area;
    clrscr();
    printf("Enter base:");
    scanf("%f", &base);
    printf("Enter height:");
    scanf("%f", &height);
    area = 0.5 * base * height;
    printf("Area = %5.2f", area);
    getch();
}
```

Output:
Enter base: 10.5
Enter height: 5.4
Area = 28.35

Program 10.15

Write a program that gets temperature from the user in Celsius and converts it into Fahrenheit using the formula F = 9/5 * C + 32.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float cel, faren;
    clrscr();
    printf("Enter temperature in celcius:");
    scanf("%f", &cel);
    faren = 9.0 / 5.0 * cel + 32;
    printf("Temperature in Fahrenheit is %5.2f", faren);
    getch();
}
```

Output:
Enter temperature in Celsius: 15.50
Temperature in Fahrenheit is 59.9

Program 10.16

Write a program that gets a three-digit number from the user and displays it in reverse order. For example if the user enter 123, it displays 321.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int n, a, b;
  clrscr();
  printf("Enter 3-digit number:");
  scanf("%d", &n);
  a = n / 100;
  n = n % 100;
  b = n / 10;
  n = n % 10;
  printf("Number in reverse order is %d%d%d", n,b,a);
  getch();
}
```

Output:
Enter 3-digit number: 512
Number in reverse order is 215

Program 10.17

Write a program that inputs 4 numbers and calculates the sum, average, and product of all the numbers.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int a, b, c, d, sum, product;
  float avg;
  clrscr();
  printf("Enter 4 numbers:");
  scanf("%d %d %d %d", &a, &b, &c, &d);
  sum = a + b + c + d;
  product = a * b * c * d;
  avg = sum / 4.0;
  printf("Sum of all numbers: %d \n", sum);
  printf("Product of all numbers: %d \n", product);
  printf("Average of all numbers: %5.2f", avg);
  getch();
}
```

Output:
Enter 4 numbers: 2 4 6 8
Sum of all numbers: 20
Product of all numbers: 384
Average of all numbers: 5.00

Program 10.18

Write a program that converts a person's height from inches to centimeters using the formula 2.54 * height.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int height;
  float height_in_cent;
  clrscr();
```

Output:
Enter height in inches: 20
Your height in centimeters is: 50.799999

```
    printf("Enter height in inches:");
    scanf("%d", &height);
    height_in_cent = height * 2.54;
    printf("Your height in centimeters is: %f", height_in_cent);
    getch();
}
```

Program 10.19

Write a program that inputs two numbers and exchanges their values. The program should display the values of variables before and after exchange.

```
#include <stdio.h>
#include <conio.h>
void main()
{
   int x, y, temp;
   clrscr();
   printf("Enter two numbers:");
   scanf("%d %d", &x, &y);
   printf("Value in x: %d \n", x);
   printf("Value in y: %d \n", y);
   temp = x;
   x = y;
   y = temp;
   printf("Value in x after exchange: %d \n", x);
   printf("Value in y after exchange: %d \n", y);
   getch();
}
```

Output:

```
Enter two numbers: 10 20
Value in x: 10
Value in y: 20
Value in x after exchange: 20
Value in y after exchange: 10
```

Program 10.20

Write a program to find the area of triangle when three sides a, b and c of the triangle are given. Use appropriate statements to input the values of a, b and c from keyboard. Formula for the area of triangle is Area = $\sqrt{s(s-a)(s-b)(s-c)}$ where s= (a + b + c) / 2.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
   float a, b, c;
   double s, area;
   clrscr();
   printf("Enter side A: ");
   scanf("%f", &a);
   printf("Enter side B: ");
   scanf("%f", &b);
   printf("Enter side C: ");
   scanf("%f", &c);
```

Output:

```
Enter side A: 10
Enter side B: 10
Enter side C: 10
Area of triangle is 43.30
```

```
    s = (a + b + c) / 2.0;
    area = sqrt(s * (s - a) * (s - b) * (s - c));
    printf("Area of triangle is %f", area);
    getch();
}
```

Program 10.21

Write a program that inputs radius of sphere from the user. Calculates its volume and surface area using the formula Area = $4\pi R^2$ and circumference = $4/3\pi R^3$ where π=3.14

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float r, area, cir;
    clrscr();
    printf("Enter radius: ");
    scanf("%f", &r);
    area = 4.0 * 3.14 * r * r;
    cir = (4.0/3.0) * 3.14 * r * r * r;
    printf("\nArea is %5.2f \n", area);
    printf("Circumference is %5.2f \n", cir);
    getch();
}
```

Output:
Enter radius: 5
Area is: 314.00
Circumference is: 523.33

Program 10.22

Write a program that inputs the radius of a circle and displays the circumference by using formula $2\pi R$. Store the value of π in a constant by using DEFINE directive.

```
#include <stdio.h>
#include <conio.h>
#define PI 3.141
void main()
{
    float r, area;
    clrscr();
    printf("Enter radius: ");
    scanf("%f", &r);
    area = 2.0 * PI * r;
    printf("Area = %5.2f", area);
    getch();
}
```

Output:
Enter radius: 5
Area is: 31.41

getch( ); getche( );

characters input

Q. What is character input? Which functions are used for character input?

C provides different function to input characters. The scanf function can be used for character input. It requires the pressing of Enter key at the end of input value. It is not useful in a situation where the input is to be entered without Enter key. For example, arrow keys are used to get character for the movement of an object in different games. The pressing of Enter key in this situation can affect the game.

C provides some other specialized functions for character input. Two important functions are as follows:

- getch()
- getche()

The above functions are part of **conio.h** library file. The world **conio** stands for **console input/output**.

Q. Explain "getch" function.

The word **getch** stands for **get character**. The **getch** function is used to input single character from the user. When this function is executed, it waits for any key to be pressed. The character entered by the user is not displayed on the screen.

The function **getch()** is defined in the header file **conio.h**. This header file must be included in the program to use this function.

Syntax

The syntax of getch function is as follows:

```
getch();
```

Another way to use this function is as follows:

```
variable = getch();
```

variable — It indicates the variable in which the character is stored. The use of variable is optional.

This function is normally used in programs to stop the execution of program temporarily. For example, the user can use it at the end of program so that the control remains on DOS screen until the user presses any key.

Program 10.23

Write a program that inputs a character using getch() function and displays it on the screen.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char c;
  clrscr();
  printf("Enter character: ");
  c = getch();
  printf("\nYou entered %c", c);
  getch();
}
```

Output:

```
Enter character:
You entered a
```

Q. Explain "getche" function.

getche Function

The word **getche** stands for **get character**. The letter **e** in **getche** stands for **echo**. The getche function is used to input single character from the user. When this function is executed, it waits for any key to be pressed. The function echoes (displays) the character on the screen entered by the user.

The function **getche()** is defined in the header file **conio.h**. This header file must be included in the program to use this function.

Syntax

The syntax of **getche** function is as follows:

```
getche();
```

Another way to use this function is as follows:

```
variable = getche();
```

variable — It indicates the variable in which the character is stored. The use of variable is optional.

Program 10.24

Write a program that displays the ASCII code of the character typed by the user.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char c;
  clrscr();
  printf("Enter character: ");
  c = getche();
  printf("\nThe ASCII code of %c is %d ", c, c);
  getch();
}
```

Output:

```
Enter character: a
The ASCII code for a is 97
```

The above program uses **getche()** function to input a character. The function stores the entered character in variable **c** without waiting to press **Enter**. The **printf** statement displays the character and its ASCII code on the screen.

Q. Explain the working of "gets" function.

The **gets** function is used to input string value from the user. The input is stored in a string variable. The user can enter any type of letters including alphabets, digits, special symbols and spaces. After typing the string from keyboard, the user presses **ENTER** key and the string is stored in the variable. The null character **\0** is automatically entered at the end of string.

Syntax

The syntax of gets function is as follows:

```
gets(variable);
```

variable It indicates the string variable in which the string is stored.

Example

The following statement inputs a string from user and stores it in a string variable str:

```
printf("Enter a string: ");
gets(str);
```

Suppose the user types "Pakistan" on the screen and then presses ENTER key. The string will be stored in str as follows:

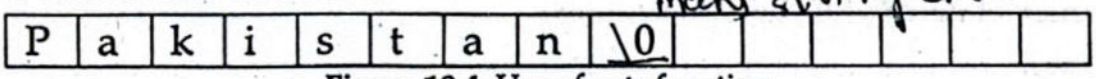

Figure 10.4: Use of gets function

Q. Explain "puts" function.

The puts function is used to display string on the screen. It can display a string constant or string variable.

Syntax

The syntax of puts function is as follows:

```
puts(parameter);
```

parameter It indicates the string variable in which the string is stored. In case of string constant, it is written in double quotes.

Example

The following example displays the contents of a string variable str on screen:

```
puts(str);
```

The following example displays a string constant on the screen:

```
puts("Programming makes life interesting.");
```

Program 10.25

Write a program that inputs a string and displays it on the screen.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char book[50];
  clrscr();
  printf("Enter name of your favorite book: ");
  gets(book);
  printf("Your favorite book is: ");
  puts(book);
  getch();
}
```

Output:

```
Enter name of your favorite book: Holy Quran
Your favorite book is: Holy Quran
```

Q. Explain "clrscr" function.

The clrscr function is used to clear the screen. It is an abbreviation of "clear screen". When this function is executed, the screen is cleared and the cursor blinks on the top-left corner. This function is available in the header file **conio.h**.

Syntax

The syntax of **clrscr** function is as follows:

```
clrscr();
```

Q. Explain the use of "sizeof" operator.

The sizeof operator is used to find the size of any data value. It gives the number of bytes occupied by that value.

Syntax

The syntax of using this operator is:

```
sizeof(operand);
```

The operand can be a variable or a constant.

Examples

Following are some examples of using sizeof operator:

```
sizeof(n);
sizeof(10);
sizeof(4.5);
sizeof("Pakistan");
```

Program 10.26

Write a program that displays the sizes of different data types.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  clrscr();
  printf("size of char = %d \n", sizeof(char));
  printf("size of int = %d \n", sizeof(int));
  printf("size of float = %d \n", sizeof(float));
  printf("size of long = %d \n", sizeof(long));
  printf("size of double = %d \n", sizeof(double));
  printf("size of long double = %d \n", sizeof(long double));
  getch();
}
```

Output:

```
The size of char : 1
The size of int: 2
The size of float : 4
The size of long : 4
The size of double : 8
The size of long double : 10
```

Q.7. a) Show output displayed by the following program when data entered is 10 and 15.

```
#include<stdio.h>
void main()
{
        int m,n;
        printf("Enter two numbers (separated by comma):");
        scanf("%d %d",&m,&n);
        m=m+10;
        n=5*m;
        printf("m=%d\t\t\t n=%d\n",m,n);
}
```

Output: m = 20 n = 100

c) Write a program that displays ASCII code of the character typed by user using scanf.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char c;
  clrscr();
  printf("Enter character: ");
  scanf("%c", &c);
  printf("The ASCII code of %c is %d ", c, c);
  getch();
}
```

Q.8. a) Show how the value -17.246 would be printed using the formats %8.4f, %8.3f, %8.1f, %8.0f, and %0.2f.

Value = -17.246 (□ means blank)

Format	Output
%8.4f	-17.2460
%8.3f	□-17.246
%8.2f	□□-17.25
%8.1f	□□□-17.2
%8.0f	□□□□-17.
%.2f	-17.24

b) Assuming x (type double) is 21.335 and y (type int) is 200, show the output of the following statements.

```
printf( "x is %6.2f \t y is  %4d\n", x, y);
printf( "y is %d \n", y);
printf( "x is %.1f \n", x);
```

x is □21.34 y is □□200
y is 200
x is 21.3

c) If the variable a, b and c are 307, 408.558 and -12.31, respectively, write a statement that will display the following line: (for clarity, the symbol □ show a blank space)

□□307□□□□□408.56□□□□-12.3

```
printf("%5d%11.2f%8.1f",a,b,c);
```

Q9. Write a program that asks the user to enter the radius of a circle and then compute and display the circle's area. Use the formula Area=PI*radius*radius. Where PI is the contant value of 3.14159. (Note: Define a constant macro PI with #define directive)

```
#include <stdio.h>
#include <conio.h>
#define PI 3.141
void main()
{
  float r, area;
  clrscr();
  printf("Enter radius: ");
  scanf("%f", &r);
  area =  PI * r* r;
  printf("Area = %f", area);
  getch();
}
```

Q10. Write a program that stores the values 'A', 'U', 3.456E10 and 50 in separate memory cells. Your program should get the first three values as input data but use an assignment statement to store the last value.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char x, y;
  float d;
  int n;
  clrscr();
  printf("Enter first value: ");
  scanf("%c", &x);
  printf("Enter second value: ");
  scanf("%c", &y);
  printf("Enter third value: ");
  scanf("%f", &d);
  n = 50;
  getch();
}
```

Q.11. Write a program that that converts a temperature in degrees. Farhenheit to degrees Celsicus. For conversion, use the following formula C = 5/9 (F – 32).

```
#include <stdio.h>
#include <conio.h>
void main()
{
  float cel, faren;
  clrscr();
  printf("Enter temperature in farenheit: ");
  scanf("%d", &faren);
  cel = 5.0 / 9.0 * ( faren – 32);
  printf("Temperature in celcius is %f", cel);
  getch();
}
```

Q.12. Write a program that takes a positive number with a fractional part and rounds it to two decimal places. For example, 25.4851 would round to 25.49 and 32.4431 would round to 32.44.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  float n;
  clrscr();
  printf("Enter a positive floating point number: ");
  scanf("%f", &n);
  printf("Number rounded to two decimal places = %.2f", n);
  getch();
}
```

Short Questions

Q.1. Define standard input.

The process of giving something to computer is known as input. The input is mostly given by keyboard. The term standard input refers to the input via keyboard.

Q.2. List some important functions for input.

scanf() gets() getch() getche()

Q.3. What is meant by standard output?

The process of getting something from computer is known as output. The output is mostly displayed on monitor. The term standard output refers to the output displayed on monitor. The result of a program is the output of that program.

Q.4. List some important functions for input and output.

printf() and puts(). The functions used for input and output are stored in the header file **stdio.h**.

Q.5. What is the use of "scanf" function? Write its syntax.

The scanf function is used to get input from the user. The input is stored in a variable. The syntax of scanf function is as follows:

scanf(control string, &variable1, &variable2, &variable3 ...);

Q.6. What is the use of "printf" function? Write its syntax.

The printf function is used to display output on the monitor. It can display text, constants or values of variables on screen in specified format. The syntax of printf function is printf(Format String, argument_list);

Q.7. What is control string in printf function?

The control string is given in double quotes. It may consist of Text, Format specifiers, Escape sequences and Parameters.

Q.8. *What is use of format specifiers in C language?*

Format specifier is used to specify the format according to which values will be read and displayed. It determines the data type of variable, field width and format of the value.

Q.9. Which symbol is used to start format specifiers?

Format specifiers are started with the symbol %.

Q.10. List out different types of format specifiers?

Different format specifiers are Integer format specifier, Character format specifier and Floating-point format specifier.

Q.11. List out different format specifiers for integer values.

Different format specifiers for integer values are %d, %i, %o, %u, and %x.

Q.12. List out different format specifiers for floating-point values.

Different format specifiers for floating-point values %f, %e and %g.

Q.13. List out different format specifiers for character values.

Different format specifiers for character values are %c and %s.

Q.14. State the purpose of %c format specifier.

The format specifier %c is used for single character values.

Q.15. State the purpose of %d format specifier.

The format specifier %d is used for signed decimal integer value.

Q.16. What is the use of field width specifiers in C language?

The number of columns used to display a value on screen is called **field-width**. A field width specifier describes the number of columns that should be used to print a value. It determines how the value will be displayed on the screen.

Q.17. Which characters are used to modify the format of output?

Escape sequences are special characters used in format string to modify the format of output. These characters are not displayed in the output. These characters always begin with backslash "\". The backslash is known as **escape character**.

Q.18. Name four escape character provided by C

Different escape sequences used in C language are \a, \b, \f, \n, \r and \t.

Q.19. Write the purpose of using \n escape sequence?

This escape sequence is used to insert new line in output.

Q.20. Write the purpose of using \t escape sequence?

This escape sequence is used to insert a TAB in the output.

Q.21. What is the use of "getch" function?

The **getch( )** function is used to input single character from user. It is an abbreviation of "get character". When it is executed, the character entered by user is not displayed on screen.

Q.22. What is the use of "getche" function?

The **getche** function is used to input single character from user. It is an abbreviation of "get character". When it is used, the character entered by the user is displayed on the screen.

Q.23. Compare getch and getche functions?

getch and **getche** are used to input single character. When getch is used, the character entered by user does not appear on screen. When getche is used, character appears on screen.

Q.24. Which function is used to clear the screen in C?

The **clrscr** function is used to clear the screen. It is an abbreviation of "clear screen". When this function is executed, the screen is cleared and the cursor blinks on the top-left corner. This function is available in the header file **conio.h**.

Q.25. Why is the ampersand (&) used in scanf function?

The ampersand (&) refers to the memory location of the variable in which the input is stored. It is placed before variable name. It is also called address operator.

Q.26. List out the names of different functions used for character input.

The functions used for character input are scanf(), getch() and getche().

Q.27. Give single printf statement to print the following message:

Number of students:

Number of teachers:

Ans: printf("Number of students: \n Number of teachers:");

Q.28. Write C statement to print the value of an integer n:

Ans: printf("%d", n);

Q.29. Given the strings FirstName and LastName, print them as LastName, FirstName.

Ans: printf("%s, %s", LastName,Firstname);

Q.30. Write C statement to print the value of double variable b with two digits showing after the decimal point.

Ans: printf("%.02lf",b);

Q.31. Write C statement to print the value of unsigned long x.

Ans: printf("%lu",x);

Q.32. Write C statement to input two values in two floating point variables a and b.

Ans: scanf("%f %f", &a, &b);

Q.33. Differentiate between printf and scanf functions.

The printf function is used to display different values on the screen. It can display text, constant or values of variables in specified format. The scanf function is used to get different types of inputs from the user. The input is stored in variables in specified format.

Q.34. Differentiate between gets and puts functions.

The gets function is used to input string values from the user. The input is stored in a string variable. The puts function is used to display string values on the screen. It can display string constant or string variable.

Q.35. Determine the output of the following code segments:

a.

```
int i=212
printf( "i=%5d",i);
printf( "i=%1d",i);
```

Ans: □□212212

b.

```
float f=3.14159
printf( "f=%5.1f",f);
printf( "f=%3.2f",f);
```

Ans: f=□□3.1 f=3.14

c.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
  float f=3.14159;
  printf("f=%3.2f",f);
  getch();
}
```

Ans: f=3.14

d.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
  int a,b,c;
  a=5;
  b=10;
  c=a+b;
  printf("The sum of a + b = %d",c);
}
```

Ans: The sum of a + b = 15

Q.36. What will the following code display?

```
printf("Book \t reading \n");
printf("is a \t very good");
printf("\n habit");
```

Answer:

```
Book    reading
is a    very good
habit
```

Q.37. Predict the output of the following?

a.

```
char Word1, Word2, Word3
Word1 = 'A';
Word2 = 'B';
Word3 = 'C';
printf("%c %c Word3", Word1, Word2);
```

Ans: A B Word3

b.

```
void main() {
  printf("Welcome \n");
  printf("Thanks");
}
```

Ans: Welcome
Thanks

c.

```
main() {
   printf ("555 \n");
   printf("55");
}
```

Ans:
555
55

d.

```
int i=10;
printf("%3d",i);
```

Ans:
□10 (□ indicates blank space)

e.

```
float f=3.14159;
printf("f=%4.2f",f);
```

Ans: f = 3.14

f.

```
char W1, W2, W3;
W1 = 'A'; W2 = 'B'; W3 = 'C';
printf("%c %c %c", W1, W2,W3);
```

Ans: A B C

Q.38. Predict the output of following code if user enters 3, 2 and 4?

```
int x, y, z;
printf("Enter three numbers: ");
scanf("%d %d %d", &x, &y, &z);
y = x + z;
x = z + 1;
printf("%d %d %d ", x, y, z);
```

Ans: 5 7 4

Q.39. Predict the output of the following code:

a.

```
printf("Hello\n world\n Pakistan");
```

Ans:
Hello
world
Pakistan

b.

```
{
  printf("*\n**\n***\n");
}
```

Ans:
*
**
***

c.

```
float f=3.24;
printf("value=%f", f);
```

Ans:
value=3.240000

d.

```
int i=20;
printf("% 6d", i);
```

Ans:
□□□□□20 (□ indicates space)

e.

```
void main()
{
int n, a, b;
n = 400;
a = n % 100;
b = n / 10;
n = n % 10;
printf("%d\n", a);
printf("%d\n", b);
printf("%d\n", n);
getch();
}
```

Ans:

0
40
0

f.

```
void main()
{
int x, y, z;
x = 10;
y = 20;
z =30;
x=x+y;
y=y+z;
z=x-y;
printf("Result = %d %d %d\n", x, y, z);
getch();
}
```

Ans:

Result = 30 50 -20

g.

```
void main()
{
printf("55\t");
printf("666");
printf("\n 777");
```

Ans:

55 666
777

h.

```
int x, y, z,res;
printf("Enter three numbers: ");
scanf("%d %d %d", &x, &y, &z);
// suppose the user inputs 3, 2, 4
res = x + y * z;
x = z + 1;
printf("%d ", res);
```

Ans:

11

i.

```
main()
{
    printf ("55 \t");
    printf("555");
}
```

Ans: 55 555

h.

```
main()
{
    int a=21;
    printf("%d",a%2);
}
```

Ans: 1

Q.40. Find the output of the following code segment?

```
int x=10;
int y=20;
int temp;
temp =x;
x=y;
y=temp;
printf("%d\n",x);
printf("%d\n",y);
```

Answer:
20
10

Q.41. Predict the output of the following code if age=25.

```
int month,age,days;
printf("Enter your age");
scanf("%d",&age);
month=age *12;
```

Answer:
age in months = 300
age in days = 9125

```
days=age * 365;
printf("age in months=%d\n";month);
printf("age in days=%d";days);
```

Q.42. What will be the output of the following statement?

```
printf("I \n love \n Pakistan");
```

Answer:
I
love
Pakistan

Q.43. Trace the output of the following programs.

a.

```
void main()
{
int n, a, b;
n = 400;
a = n % 100;
b = n / 10;
n = n % 10;
printf("Result=%d%d%d",n,b,a);
getch();
}
```

Ans: Result=0400

b.

```
void main()
{
int x, y, z;
x = 3;
y = 2;
z = 4;
printf("%d  %d %d", x+y, y+5, x+z);
getch();
}
```

Ans: 5 7 7

c.

```
#include<stdio.h>
#include<conio.h>
void main()
{
char ch1 = 'x', ch2 = 'y', ch3 = 'z';
printf("%c \t %c \t ch1", ch2, ch3);
getch();
}
```

Ans: y z ch1

d.

```
void main()
{
char ch1 = 'A', ch2 = 'B', ch3 = 'C';
printf("ch2 \n %c \n ch1",ch3);
}
```

Ans:
ch2
C
ch1

e.

```
int a = 9;
a = a % 4;
printf("\n %d is Result", a);
```

Ans: 1 is Result

f.

```
{
    float f=3.2413;
    printf("f=%3.3f",f);
}
```

Ans: f = 3.241

Q.44. Trace out errors in the following code:

```
float area, r
clrscr();
printf("Enter radius);
scanf("%f"; r);
```

Ans: The semicolon is missing at the end of first statement. The closing double quotation is missing in the third statement. Comma must be used instead of semicolon in the last statement before r.

Q.45. Trace out the errors from following piece of code:

```
#include<STDIO.H>
#include<conio.h>
```

```
Void main()
{
  Printf("Pakistan");
}
getch();
```

Ans: The word "void" must be in lowercase in the third statement. The word "printf" must also be lowercase. The closing braces must be written after getch(); function.

Q.46. Trace the errors in the following code:

```
#include(stdio.h)
{
 Main();
}
```

Ans: The header file **stdio.h** must be written as <stdio.h>. The main() function must be written before the braces. The first letter of main() function must be lowercase. The semicolon after main() function is not allowed.

Q.47. Trace error in the following C code.

```
# include <stdio.h>
Void main()
{
Prin (" C Language is interesting");}
```

Ans: The word **Void** must be written as **void**. The first word in the braces must be **printf** instead of **Prin**.

Q.48. Find the error in the code.

```
#include<STIO.H>
void main() {
int a;
a=20;
printf("%f",a);
getch(); }
```

OUTPUT
INPUT

Ans: The name of header file is **STDIO.H**. The symbol **%f** must be replaced with **%d** for integer output. The **getch()** requires to include **conio.h** header file.

Q.49. Trace the error.

```
#includes<stdio.h>
Main()
{
 Print("Helloworld");
}
```

Ans: The word **Main** must be written as **main**. The word **Print** must be written as **printf**. The program will show a warning to return a value. Write **void** before main function or use **return** statement at the end to return a value.

Q.50. Find the error in the code.

```
#include<stdio.h>
Void main();{
x=20
y=40
z=x+y
```

```
printf(%d",z)
getch();]
```

Ans: The word **Void** must be written as **void**. The semicolon after **main()** is invalid. The first four statements in body of **main()** function must end with semicolon. The variables x, y and z must be declared before use. The function body must end with } instead of].

Q.51. Trace the errors

```
int b=8;
int c=0
x=number
print  f("%f",x).
```

Ans: The second and third statements must end with semicolon. The variable number in third line must be declared before use. The word **print f** must be written without space as **printf**. The period at the end of fourth statement must be replaced with semicolon. The symbol %f in the last statement must be replaced with %d for integer output.

Q.52. Find the error in program code.

```
int x=20, y=35
x=y++
y=++y
printf("%d f %f\n", x,y);
```

Ans: The first three statements must end with semicolon. The symbol %f must be replaced with %d for integer output. The letter f after the symbol %d in the last statement is not required.

Q.53. Trace the error of the following code.

```
void main()
{
Int A=10
Printf(%d",a)
}
```

Ans: The word Int must be written as **int**. The two statements in braces must end with semicolons. The word **Printf** must be written as **printf**. The symbol %d must be enclosed in double quotations as "%d".

Q.54. Trace out error from the following code:

a.

```
// suppose x=5, y=10, z=8
int x,y,z
printf("Enter three no");
scanf("%d %d %d ",&x,&y,&z);
y=x+z+y;
x=z+1
printf(" %d %d %d",x,y,z);
```

Ans: The first and fifth statements must end with semicolon.

b.

```
main [ ];
{
  float n;
  printf("%f" , n);
}
```

Ans: The brackets [ ] after main must be replaced with parenthesis ( ) and semicolon at the end of line is not required.

Q.55. Trace out errors in the following code:

```
int x=20
x+4=x;
```

printf("value of x is %d",x)

Ans: The first and third statement must end with semicolon. The second statement is invalid because an expression cannot be written on the left side of assignment operator.

Q.56. Find error in the following code:

```
inlude<stdio.h>
Main()
{
  Int i=10;
  Printf("%d",i);
}
```

Ans: The word "inlude" must be replaced with "include" in first statement. The word "Main" must be written in lowercase. The word "Int" in the variable declaration must be written in lowercase. The word "Print" in the last statement must be written in lowercase.

Q.57. Find the errors in the following code:

```
Main();
{
 PRINT F("Hello");
}
```

Ans: The word "Main" must be written in lowercase. The use of semicolon after main function is invalid. The word "PRINTF" must be written in lowercase.

Q.58. Find out error in the following code:

```
main();
{
 Printf("Hello"))
 }}
```

Ans: The use of semicolon after main function is invalid. The word "Printf" must be written in lowercase. The printf function must be closed with single closing parentheses. There must be only one closing braces in the ending statement.

Q.59. Find out errors from the following code:

```
Char C
Clrscr();
Print ("enter character");
c=getch();
printf ("\n",C);
```

Ans: The word "Char" must be written in lowercase and the first statement must end with semicolon. The name of function in second statement must be written in lowercase. The word "Printf" must be written in lowercase. \n in last statement must be replaced with %c.

Q.60. Rewrite the following code after correction:

```
void main()
{
 float a=3.14;
printf("%d",a);
 getch();
}
```

Ans:

```
void main()
{
 float a=3.14;
printf("%f",a);
 getch();
}
```

Q.61. Trace the errors in the following programs.

a.

```
void main() {
p = 3.141593
int x, y;
x = 10;
y = 30;
z = x + y;
printf("Result=", z);
getch()    }
```

Ans: The variable p is not declared. The semicolon missing after first assignment statement and getch() function.

b.

```
include<stdio.h>
main()
{
 int  a=10;
 prinft("%d",a)
}
```

Ans: The # sign is missing before include. The void keyword is missing before main. The semicolon after printf statement is missing.

c.

```
{
   int i=5;
  print f("%d", i);
}
```

Ans: The space before f is not allowed.

d.

```
#include<STDIO.H>
#include<CONIO.H>
void Main(){
printf("Pakistan is Islamic country")
}
```

Ans: The first letter of main is capital instead of lower. The semicolon is missing after printf statement.

e.

```
#include (stdio.h)
void main()
{
   float n;
   scanf("%d", &n);
   printf("The Number is %f", n);  }
```

Ans: The header file is written in parenthesis. %d is used instead of %f for input with scanf.

f.

```
int n;
printf("Enter any number");
scanf("%c", &p);
printf("Number=", n)
```

Ans: The variable p is not declared. %c is not used for integer variable. The semicolon is missing in the last line.

g.

```
#include<stdio.h>
#include<conio.h>
void main();
{
 Printf("%S", pak);
 getch();  }
```

Ans: The semicolon after main() is not required. The first letter of printf is capital. The %S is used instead of %s. The variable pak is not declared.

h.

```
#include<stdio.h>
void main();
printf("%c","pak");
}
```

Ans: The semicolon after main function is not required. %c is used with string data. The opening parenthesis are missing after main().

Q.62. Write a single C statement to print the following output:

Hello to
The world of
C Programming

Ans: printf ("Hello to \n The world of \n C Programming");

Multiple Choice

1. Which of the following function is used to input data in C programs?
   a. scanf b. printf c. get() d. getch()
2. The functions used for input and output is stored in:
   a. stdio.h b. conio.h c. math.h d. None
3. The function that is used to display output on screen is called:
   a. printf b. scanf c. pow d. None
4. The function getche( ) is defined in:
   a. stdio.h b. conio.h c string.h d. math.h
5. The format specifier %u is used for:
   a. integer b. unsigned short
   c. unsigned float d. unsigned long int
6. Which of the following is displayed by printf function?
   a. Text b. constant or values of variable
   c. A and b d. None
7. How many variables can be used in one printf function?
   a. One b. Two c. Many d. None
8. Which of the following things are determined by format specifier?
   a. Data type b. Field width
   c. Format of the value d. All of these
9. Format specifier is started with symbol:
   a. ? b. % c. & d. None
10. Which of the following format specifiers is used for integer data type?
    a. %d b. %s c. %c d. %f
11. Which of the following format specifiers is used for character data type?
    a. %s b. %c c. %d d. %f
12. Which of the following format specifiers is used for float data type?
    a. %s b. %f c. %c d. %d
13. Which of the following format specifier used for string?
    a. %f b. %d c. %c d. %s
14. Which of the following is a possible value of flag?
    a. - b. + c. space d. All
15. Which of the following is NOT legal element of control string in printf statement?
    a. Ordinary characters b. Escape sequence
    c. Format specifier d. Variables
16. The format specifier for output are:
    a. %d for int and %f for float
    b. %c for char
    c. %s for string
    d. All
17. The general form of format specifier for floating point value:
    a. %m.nf b. m.n% c. m%.nf d. m.n%f
18. Which of the following symbol is used with % sign to display output left-justified
    a. - b. / c. \ d. None

19. The statement printf ("%10d",n) will display the value of n :
a. Right justified in ten-character space
b. Left-justified in ten-character space
c. Center-justified in ten character space
d. None

20. The statement printf ("%-10d",n) will display the value of n :
a. Right justified in ten-character space
b. Left justified in ten-character space
c. Center-justified in ten character space
d. None

21. An ampersand before the name of a variable denotes
a. Actual Value
b. Variable Name
c. Address
d. Data Type

22. Which escape sequence can be used to begin a new line in C?
a. \a
b. \b
c. \m
d. \n

23. Which escape sequence can be used to beep from speaker in C?
a. \a
b. \b
c. \m
d. \n

24. Which escape sequence can be used to insert a Tab in C?
a. \a
b. \b
c. \t
d. \n

25. Which escape sequence can move the cursor at the beginning of current line in C?
a. \a
b. \b
c. \m
d. \r

26. The escape sequence for backslash is?
a. \
b. \b
c. \ \
d. \t

27. Which function is used to input data in C-program?
a. scanf
b. getch()
c. gets()
d. All

28. getch() stands for:
a. go character
b. get character
c. give character
d. All

29. %f is used for:
a. double
b. float
c. integer
d. short

30. Which character signifies the beginning of an escape sequence?
a. {
b. \
c. //
d. /

31. Which escape sequence is used to erase a character left to the cursor position?
a. \b
b. \del
c. \f,
d. \c

32. Which character signifies the beginning of an escape sequence?
a. {
b. \
c. //
d. /

33. Which of the following is not a valid escape code?
a. \t
b. \r
c. \y
d. \f

Answers

1. a	2. a	3. a	4. b	5. b	6. c
7. c	8. d	9. b	10. a	11. b	12. b
13. d	14. d	15. d	16. d	17. a	18. a
19. a	20. b	21. c	22. d	23. a	24. c
25. d	26. c	27. d	28. b	29. b	30. b
31. a	32. b	33. c			