**Use of getch(), getche() and getchar() in C –**

Function : getchar()

getchar() is used to get or read the input (i.e a single character) at run time.

During the program execution, a single character is get or read through the getchar(). The given value is displayed on the screen and the compiler wait for another character to be typed. If you press the enter key/any other characters and then only the given character is printed through the printf function.

Function : getche()

getche() is used to get a character from console, and echoes to the screen.

Function : getch()

getch() is used to get a character from console but does not echo to the screen.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DIFFERENCE SCANF() AND GETS()

the main difference between gets() and scanf() function is that while getting any input as a stream of characters or as a string using scanf() , the end termination character is a white space (Space / Tab or Enter) for scanf() , so upto blank space it will be saving it in the memory. It ignores the characters coming after white space.

In gets() , it allows with the blank spaces and the termination condition is an enter key (\n) and accepts the whole line as input.

**Write example**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Difference between exit and break in c programming**

The major difference between break and exit() is that break is a keyword, which causes an immediate exit from the switch or loop (for, while or do), while exit() is a standard library function, which terminates program execution when it is called.

The general syntax of the exit() function is : void exit(int return\_code) , eg exit(0);

**Also write example of Break**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Advantage and disadvantage of Switch over if else:**

The if-else ladder is of type strict condition check, whereas switch is of type jump value catching.

Limitations of switch over if-else:

The variable expression is not allowed in cases. For eg case i+2: is not allowed in switch, but is valid on if-else.

Switch can not be used on float/ double type variable but if-else can be used on them.

Advantages of switch over if-else:

A switch statement works much faster than equivalent if-else.

It is because compiler generates a jump table for a switch during compilation. Consequently, during execution, instead of checking which case is satisfied, it only decides which case has to be executed.

It is more readable in compare to if-else statements.

It is more manageable for having higher level of indentation than if. If there are large number of comparisons for a condition in program, using switch is easier to for coding over if-else.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**use of header files in c**

A header file is a file containing C declarations and macro definitions to be shared between several source files. We request the use of a header file in our program by including it, with the C preprocessing directive ‘#include '.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Entry Checking and Exit Checking Loop :

Depending on the position of the control statement in the loop, a control structure can be classified into two types; entry controlled and exit controlled. They are described below.

Entry checking loop: The types of loop where the test condition is stated before the body of the loop, are known as the entry checking loop. So in the case of an entry checking loop, the condition is tested before the execution of the loop. If the test condition is true, then the loop gets the execution, otherwise not. For example, for loop and while loop are entry checking loop. In the given figure, the structure of an entry checking loop is shown.

**Draw the Attached Picture**

Exit checking loop : The types of loop where the test condition is stated at the end of the body of the loop, are know as the exit checking loops. So, in the case of the exit checking loops, the body of the loop gets execution without testing the given condition for the first time. Then the condition is tested. If it comes true, then the loop gets another execution and continues till the result of the test condition is not false. For example, do....while loop is an exit checking loop. The structure of an exit checking loop is given in the figure.

**Draw the Attached Picture**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_