

### Character set of C programming.

→ The characters that can be used to form words, numbers and expressions depend upon the computer on which the program is run. The characters in C are grouped into the following categories:

1. Letters
2. Digits
3. Special Characters
4. White spaces

The entire character set is given in table:

1. Letters	<ul style="list-style-type: none"> <li>• Uppercase (A...Z)</li> <li>• lowercase (a...z)</li> </ul>	
2. Digits	All decimal digits (0-9)	
3. Special characters	Comma (,) Period (.) Semicolon (;) Colon (:) Question mark (?) Apostrophe (') Quotation mark ("") Exclamation mark (!) Vertical bar ( ) Slash (/) Backslash (\) Tilde (~) Underscore (_) Dollar sign (\$)	Percent sign (%) Ampersand (&) Caret (^) Asterisk (*) Minus sign (-) Plus sign (+) Opening angle bracket (<), or less than sign Closing angle bracket (>), or greater than sign Left parenthesis (() Right parenthesis ()) Left bracket ([)



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

#### 4. White spaces

right bracket (})

left brace ({)

right brace (})

number sign (#)

Blank space

Horizontal Tab

Carriage return

New line

Form feed

Q. What is variable? Write down the rules of writing variables in C program.

→ Variable: A variable is a data name that may be used to store a data value.

- (1) variable may take different values in different
- (2) times during execution.

A variable name can be chosen by the programmer in a meaning full way. Some examples of such names are

Average

height

Total

Counter-1

Class-strength



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

Variable names may consist of letters, digits and the underscore (\_) character.

### Rules of writing variable name:

1. A variable can have alphabets, digits and underscore.
2. A variable name can start with the alphabet and underscore only. It can't start with a digit.
3. No whitespace is allowed within the variable name.
4. A variable name must not be any reserved word or keyword like int, goto etc.
5. Uppercase and lowercase are significant.

Some examples are of variables names:

Variable name	Valid?	Remark
First_tag	Valid	
char	not valid	char is a keyword
Price \$	not valid	Dollar sign is illegal.
group one	not valid	Blank space is not permitted
average_num. bert	valid	
int_type	valid	keyword may be part of a name
mark	valid	



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

41 How can a program to read data from keyboard?

→ In C programming, scanf() is one of the commonly used function to take input from the user. The scanf() function reads formatted input from the standard input such as keyboards. The general format of scanf is as follows:

```
scanf("control string", &variable1,  
      &variable2, ...);
```

Here, the control string contains the format of data being received. The ampersand symbol (&) before a each variable name is an operator that specifies the variable's address. We must always use this operator. For example,

```
scanf("%d %d", variable1, variable2);
```

Since the control string ("%d") specifies that an integer value is to be read from the keyboard, we have to type the value in integer form.



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

Q1. What is a variable and what do you mean by the value of a variable?

→ Variable: A variable is a data name that may be used to store a data value. A variable may take different values in different times during execution.

C variable is a named location in a memory where a program can manipulate the data.

This location is used to hold the value of the variable. The value of the C variable may get change in the program. C variable might be belonging to any of the data type like int, float, char etc.

Q2. How do variables and symbolic names differ?

→ Symbolic name: A symbolic name is a constant that given to some numeric constant or a character constant or string constant or any other constant. #define is used for defining symbolic constants.

Variable: A variable is a data name that may be used to store a data value. Variable names may consist of letters, digits and underscore (\_) character.

Declaration of a symbolic name just defines a name that can be used in a program.



Sub: \_\_\_\_\_

Day

--	--	--	--	--	--	--	--

Time: \_\_\_\_\_

Date: / /

Since the symbolic names are constant, their value can't be changed in the program. On the other hand, variable value can be changed within the program.

Q What do you mean by the overflow and underflow of data?

→ Overflow and underflow happen when we assign a value that is out of range of the declared data type of the variable.

If the value is too big, we call it **overflow**, if the value is too small, we call it **underflow**.

The largest value that a variable can hold also depend on machine. C does not provide any warning or indication of integer overflow. It simply gives incorrect result.

Variable: A variable is a data value that may be used to store a data value. A variable may consist of letters and digits and a underscore character.

Declaration of a symbolic name just before its use in a program.



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

Q. Explain C Tokens

→ In a passage of text, individual words and punctuation marks are called tokens. Similarly, in a C program, individual units are known as C tokens. C has six types of tokens and they are:

Keywords: Every C-word is classified as either keyword or an identifier. All keyword has fixed meaning and that is can't be changed.

Keywords serve as basic building blocks for program statement. All keyword must be written in lowercase. Example: auto, double, int, return etc

Identifiers: Identifiers refer to the names of variables, functions and arrays. These are user defined names and consist of a sequence of letters and digits, with a letter as a first character. Both uppercase and lowercase letters are permitted, although lowercase letters are commonly used. The underscore character is also permitted in identifiers. It is usually used as a link between two words in long identifiers.

Constants: Constants in C refer to fixed values that do not change during the execution of a program. Constants may be numeric constants or character constants. Constants may belong to any of the data types.



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

string: The string can be defined as the one dimensional array of character terminated by a null ('0'). The character array or the string is used to manipulate text, such as word or sentences. Each character in the array occupies one byte of memory and last character always be zero.

Operators: An operators is a symbol that tells the compiler to perform specific mathematical or logical functions. C operators are, Arithmetic (+, -, \*, /, %), Relational (<, >, <=, >=, ==, !=), Arithmetic Operator (+, -, \*, /, %), Relational operator (<, >, <=, >=, ==, !=), Logical operator (&, ||, !), Assignment operator (=, +=, -=, \*=, /=, %=), Increment and decrement operator (++ , --), Conditional operator (? , !), Bitwise operator (&, |, ^, ~, <<, >>), Special operator (sizeof, &).



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

Special Symbols: The special symbols that are used in C/C++ with some special meaning and some specific function are called as special symbols.

Examples: Brackets [ ], Braces { }, comma (,), semicolon (;), Parenthesis (), Asterisk (\*), Assignment operator (=), Preprocessor (#).

List of Keyword:

→ Every C word is classified as either keyword or an identifier. All keywords have fixed meanings and these meanings can not be changed.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

\* All keywords must be written in lowercase.



All C compilers support five fundamental data types namely integer (int), character (char), floating point (float), double precision floating point (double) and void.

Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

#### Q1 Rules for Identifiers :

1. First character must be an alphabet or underscore.
2. Must consist of only letters, digits or underscore.
3. Only first 31 characters are significant.
4. Cannot use a keyword.
5. Must not contain white space.

#### Q1 Explain Data Types of C.

→ Each variable in C has an associated data type. Each data type requires different amounts of memory and has some specific functions/operations.

ANSI C supports these <sup>three</sup> four classes of data types and they are:

1. Primary data types
2. Derived data types
3. User-defined data types

Following are the five primary data types which are commonly used in C:

- int : We can use int for declaring an integer variable. An int variable stores an integer. Integers are whole numbers with a range of values supported by a particular



machine. Range of int in 32 bit is  $-2,147,483,648$  to  $2,147,483,647$ . int are usually stored in 2 bytes on internal storage.

integer types	Range (16 bit)
int	$-32,768$ to $32,767$
unsigned short int	$0$ to $65,535$
long int	$-2,147,483,648$ to $2,147,483,647$
unsigned long int	$0$ to $4,294,967,295$
long long int	$-(2^{63})$ to $(2^{63}) - 1$
unsigned long long int	$0$ to $18,446,744,073,709,551,615$

• **char:** The most basic data type in C. It stores a single character and requires single byte of memory in almost all compilers. Characters are usually stored in 1 bytes on internal storage.

• **float:** It is used to store real numbers (numbers with floating point value) with single precision. Floats are usually stored in 4 bytes on internal storage.

• **double:** It is used to store real numbers (numbers with floating point value) with double precision. Double are usually stored in 8 bytes on internal storage.



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

Types	Range
float (32 bit)	$1.2 \text{E} - 38$ to $3.4 \text{E} + 38$
double (64 bit)	$1.2 \text{E} - 38$ to $3.4 \text{E} + 38$
long double (80 bit)	$3.4 \text{E} - 4932$ to $1.1 \text{E} + 4932$

• **Void**: Void is an empty data type that has no value. We use void data type in functions when we don't want to return any value to the calling function.

2. Derived data types: Derived data types are those that are defined in terms of other data types. Array, pointers, struct and union are the derived data types.

3. User defined data types: Those data types which are defined by the user. Structures, Union, Typedef, enum, class are the user define data types.



Q. What is Constant? Write down the types of constants.

→ Constants in C refer to fixed values that do not change during the execution of a program. C supports several types of constants.

Integer Constants: An integer constant refers to a sequence of digits. It can be an octal integer or a decimal integer or even a hexadecimal integer. The integer constant used in a program can also be of an unsigned type or a long type.

Example: Decimal integer constant: 55  
 Hexa-Decimal integer constant: 5AB  
 Octal integer constant: 023

Real Constants: Real constants are also known as floating point constants. This type of constant must contain both the parts decimal as well as integers. These quantities are represented by numbers containing fractional parts are like 17.548. Sometimes, the floating point constant may also contain the exponential part.

Example: We represent the floating point value 3.14 as  $3F+14$  in its exponential form.

#define symbolic\_name value\_of\_constant



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

Single character constant: A single character constant (or simply character constant) contains a single character enclosed within a pair of single quote mark.

Example: '5', 'x', '1', 'a', 'z' [ 'a' → 97  
'z' → 122 ]

String constant: The string constants are the collection of various special symbols, digits, characters and escape sequences that get enclosed in double quotation mark.

The definition of a string constant occurs in a single line. "This is cookie", "Hello" are the example.

Q Explain Symbolic constant.

→ We can use unique constants in a program. Before defining a constant we must ensure modifiability and understandability.

Because sometimes we need to modify these constant and obviously the purpose of constant is understandable by the reader of the program. That's why the name of the constant is important. A constant defined as follows:

```
#define symbolic_name value_of_constant;
```



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

Note that:

1. '#' must be the first character of the line and the word 'define' must be written in lowercase. No space between the # and the word define.
2. Symbolic names have the same form as variable name.
3. There is no semicolon after the statement.

Example: `#define pi 3.1416`

`#define pass-mark 33`