C PROGRAMMING

A general-purpose programming language developed by Dennis Ritchie and Ken thompson to construct utilities for the UNIX operating system at AT&T Bell laboratory originally developed in 1972.

History of the C programming Language.

The language was named C because of its features derived from an already existing programming language B which never got anywhere close to C in terms of popularity and acceptance.

In 1978 Brian Kernighan & Dennis Ritchie produced the first publically available description of C now known as the K&R standard. (C78)

C became popular in the 1980's as the most widely used programming language.

C has been standardized by ANSI since 1989 and by International Standard Organisation (C90)

The C standard was further revised in late 1990's to include inline functions, and new datatypes - this is refered to as C99 standard

New features were introduced in another revision to the standard (C11) in 2011.

The current standard for C language is C17 introduced in the year 2018. It only made technical corrections to C11.

Features of C Programming Language

1. Procedural Language

- Modeled on the concept of the procedure call where a procedure contains a series of computation to be executed

2. Fast and Efficient

The C language is compiled to provide very low level access to memory.

The C language constructs map very efficiently to machine instructions which makes running C programs very fast.

C is considered as a middle level language which allows you to efficiently manage hardware and memory constructs.

3. Modular

C programming can be written in small chunks called functions and made available via libraries

The standard C libraries offer a wide variety of functions for common operations that you might perform.

4. Statistically Typed.

Variable types are checked at compile time rather than runtime.

Camlin Page
Date / /

statically typed languages are more efficent as faster as type checking is performed before execution.

Rich Build in Operators.

Common operations are already available to the programmer and need not be defined from scratch.

Arithmetic, logical, relational, assignment and binary operations are all available as a part of the standard c implementation.

Libraries for common Functions

Implementation of common functions, macros type definitions and other utilities such as string manipulations, mathematical computations, I/O processing, memory management and other services are available.

Importance of c programming language.

C still powers the kernels of operating system such as windows Linux and macOS.

Mobile OS kernels such as IOS and Android are also written in C.

Databases such as oracle, MySQL, MY SQL Server are coded in C

Embedded systems programing for IOT uses C.

Conversion of c program to Executable code

Compiled code

A compiled code is a code translated by a series of step one off. It translates the entire source code at a time

Programs that are compiled exactly once can be executed several times.

Interprited code.

It is the translation of a source code that is read and translated line by line.

Programs need to be interpritted each time they are executed.

- C is a compiled language.

Four Main Phases to Go from Source to Executable Code.

1. Preprocessor.

Performs some preparatory operations on the source code before compilation.

2. Compiler.

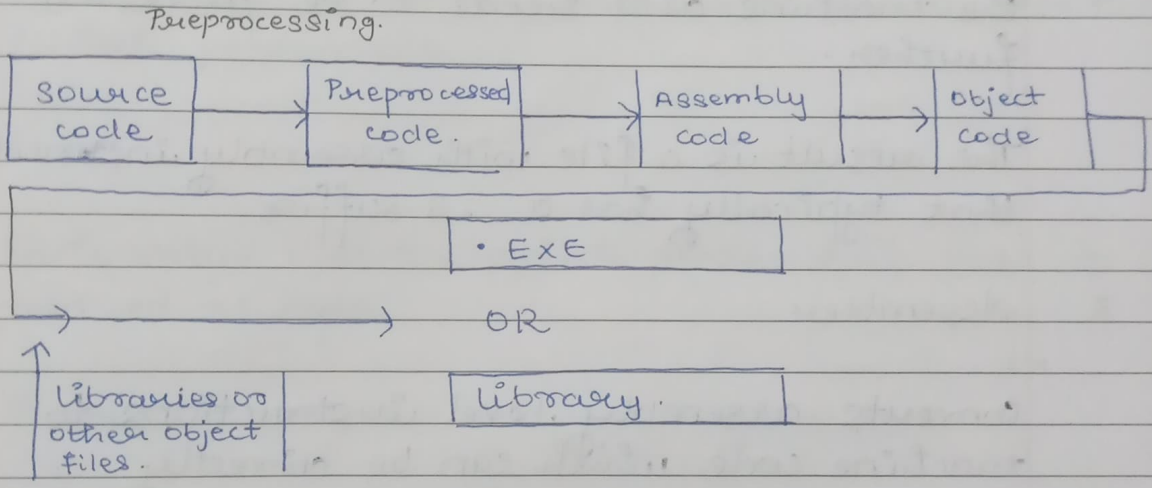
Converts the high level source code (which has been pre-processed) to assembly code.

Assembler.

Converts assembly instructions to machine code that can be executed on the computer.

Linker.

Brings different bits of machine code together to build an executable code.



1. Preprocessing.

Defines behaviours that are carried out on the code before compilation.

C code contains preprocessor directives identified by the #prefix.

At the end of preprocessing all the preprocessed directives have been replaced by actual C code; usually stored in a file with a .i suffix

Removing comments

- expanding macros
- Include code that is compiled conditionally (Ex: #if, #else, #ifdef #endif)

2. Compilation.

Converting source code in C to assembly language - a very low level programming language.

Assembly code is not directly understood by the machine and needs to be processed further.

The result is a file with assembly instructions that typically has a .s suffix.

3. Assembler.

Converts assembly level instructions to machine code which can be directly executed on the computer.

The output of an assembler is usually stored in files with .o extension (also called object code).

4. Linking.

Brings together the machine code for different files into one executable

Also includes linking the executable with external libraries and framework.

Keywords of C Programming language.

1. Auto

Declares an automatic variable. Auto is a storage class, variable declared with this keyword is recreated each time function is executed. local to a function

2. Break.

Terminates the innermost loop. Terminates the switch statement

3. Case

Each value that the switch statement test is termed as case.

4. Char

Represents a character or letter storage, It is an integer type. Due to its storage of number for characters.

5. Const.

Declares a variable as constant. Initialises a value.

6. Continue.

Forces the next iteration of loop to take place, skipping any code in between.

7. default.

Used in a switch statement to indicate the control path when no other case is selected.

8. do.

Used with while keyword. It indicates a loop.

9. Double.

datatype storing high value float point [decimal] data.

10. Else.

Used with if, it indicates initiation of a statement when if becomes false.

11. Enum.

Datatype enabling variable to be a set of pre-defined constant.

12. Extern.

Declares a global variable that is without any memory assigned.

13. float.

Datatype for decimal values.

14. for.

loops the statement after testing.

15. goto.

Jumps from one block to another.

16. If

Checks the condition & executes the statement if true.

17. Int.

datatype for integer values.

18. long.

datatype for high integer values.

19. register.

declaring a register variable which stores variable in CPU register instead of memory for faster accessability.

20. return.

ends execution of a function & returns control to the calling function.

21. short.

datatype integer stores same value as int

22. signed.

datatype used with int to store negative and positive values [optional].

23. sizeof

compile time unary operator used to compute the size of its operand [in bytes]

24. static.

scope of static variable is throughout the program.

25. struct.

composite datatype that defines physically grouped list of variables under one name in a block of memory.

26. switch.

test cases that are true.

27. Typedef

It replaces the name of an existing datatype with the name of provided datatype.

28. union.

datatype that allows to store different datatype with the name of same memory location.

29. Unsigned.

datatype specifier to make variable store only non-negative values.

30. Void.

datatype with no value.

31. Volatile.

Qualifier informing the compiler that variable value can change anytime without a task by source code.

32. while.

executes statements if condition true.

Escape Sequences.

\a	Audible signal
\b	Backspace
\t	Tab
\n	Newline
\v	Vertical Tab.
\f	Newpage \ clearscreen.
\r	Carriage return
\\	double backslash
\'	Single quote mark
\"	double quote mark.
\0	Null.

Operators

- Arithmetic operators.

		Syntax
multiply	*	a * b
devide	/	a / b
addition	+	a + b
substraction	-	a - b
Modulus	%	a % b

- Relational Operator.

compares two values or expressions. Relational operator returns either true or false value depending on whether the conditional relationship between the two operands holds or not.

less than	<
Greater than	>
less than or equal	<=
greater than or equal	>=

- Equality Operators.

Equality operator compare operands for strict equality or inequality.

Returns one if both operands equal	= =
Returns one if operands not having same value	!=

- Logical Operator.

C language supports three logical operators they are.

- Logical AND (&&)

It is a binary operator, which simultaniously evaluates two values or relational expressions.

A	B	A && B
0	0	0
0	1	0
1	0	0
1	1	1

- Logical OR (||)

Returns a false value if both operand are false.

A	B	A || B
0	0	0
0	1	1
1	0	1
1	1	1

- Logical NOT (!)

Logical not takes a single expression and produces a zero if expression evaluates to a non zero value and produces a 1 if expression evaluates to a non-zero

A	!A
0	1
1	0

Unary operators

Unary operators work upon or act on a single operand. C language supports three unary operators.

Unary Minus (-)

This operator negates the value of its operand.

```
int a, b = 10;
a = -(b);
```

Result : -10.

Increment & Decrement.

Increment (++)

It increases the value of its operand by 1

Decrement (--)

It decreases the value of its operand by 1

Conditional operator.

Expression 1 is evaluated first. If it is true then only expression 2 is evaluated and becomes the result of expression otherwise expression 3 is evaluated.

```
exp1 ? exp2 : exp3
```

Bitwise operator.

It performs operations at bit level.

Bitwise AND

Like boolean AND(&&) bitwise AND (&) perform operations at bit level instead of bytes. The first operand's bit is ANDed with corresponding bit in the second operand.

2015 Page No.
Date / /

Assignment - 1

1. Describe about history of C programming

The language was named C because its features were derived from an already existing language called B. In 1978 Brian Kernighan & Dennis Ritchie produced the first publically available description of C now known as the K&R standard.

It was developed by Dennis Ritchie & Ken thompson to construct utilities for the Unix operating system at AT&T bell laboratories in 1972 later on It was standardised by ANSI in 1989 and became [C89] then after adoption by ISO it became (C90) in 1999 there after (C99) in 1999, then (C11) in 2011 then (C17) in 2017 latest version is expected in 2023.

---

2. Describe about identifiers and keywords.

Identifiers

Identifiers are names defined by user of a C programm. These

names can be of variables, functions arrays, structure, labels etc...
An identifier can be composed of letters such as uppercase lowercase etc
- Starting letter can be either a alphabet or Punderscore.
- Keywords cannot become identifiers.
- length should no more be than 31 characters.

Keywords

A keyword is a predefined word in C language and have special meanings to the compilers. There are 32 keywords in C programming language.

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

---

3. What do you mean by constant and variable? How it will be declared?

Constants

The variable which have a fixed value that cannot be modified are known as constants.

1. Example #define val 10
   here val = a variable
   10 is the fixed value.
2. const float pi = 3.14.

Variables

A variable is a container or a storage place for a value which can be changed or modified in the course of program. They are of two main types.

1. Local variables
2. Global variables.

```
#include <stdio.h>
#include <conio.h>
int a = 50, b = 40;  // global variable
int main()
{
 int c=10, d=20.  // local variable.
printf ("a = %.d and b= %.d", a, b);
printf ("c = %.d and d = %.d", c, d);
}
return 0;
```

4. What do you mean by Data Algorithm? Describe its properties. Define flow chart.

An algorithm is a set of commands that must be followed to perform calculations or other problem solving operations

It is a finite set of instruction carried out in a specific order to perform a particular task

Properties.

1 Non Ambiguity.

Each instruction should be clear & precise, thus it should not denote any conflicting meaning.

2. Range of Input.

Range of input should be specified otherwise it can go in infinite stat

3. Multiplicity.

Same algorithm can be represented in several different ways. We can write in simple english or pseudo codes

Speed: Algorithm should be fast in producing output.

Finiteness: It should be properly terminated after performing required calculations.

Flowchart

The pictorial & structural approach of an algorithm is known as a flowchart. It consists of step by step process in a sequential order.

5. Represent $(1460.125)_{10}$ in single and double precision formats.

Single Precision Format

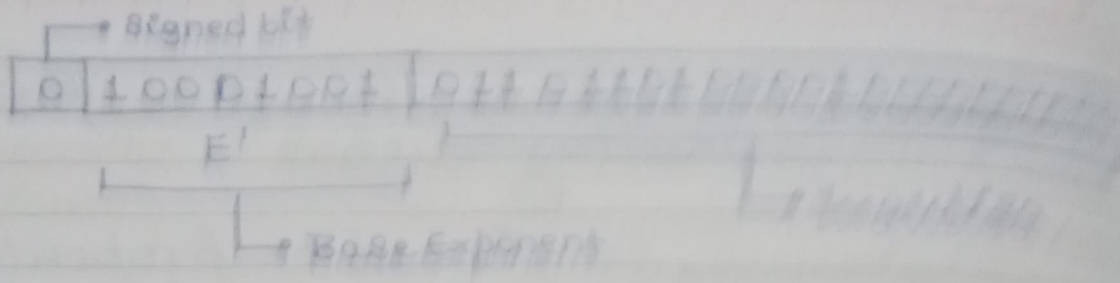
$(1460.125)_{10} = (10110110\ 100.001)_2$

$= 1.0110110110100001 \times 2^{10}$

$E' = 127 + E$

$= (137)_{10}$

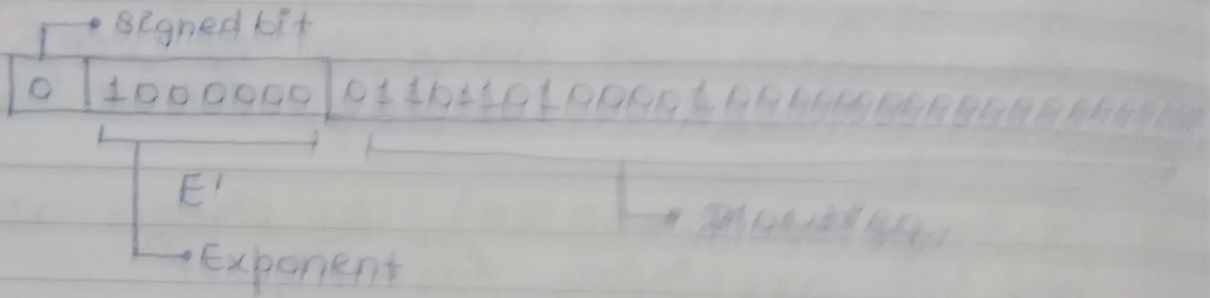
$= (10001001)_2$

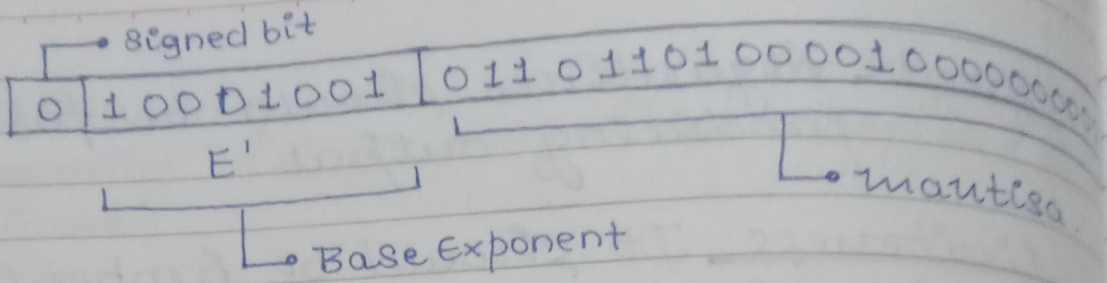


Double Precision Format,

$(1460.125)_{10} = (1.0110110100.001)_2 \times 2^{10}$

$E' = 1023 + E = (1032)_{10} = (10000001000)_2$

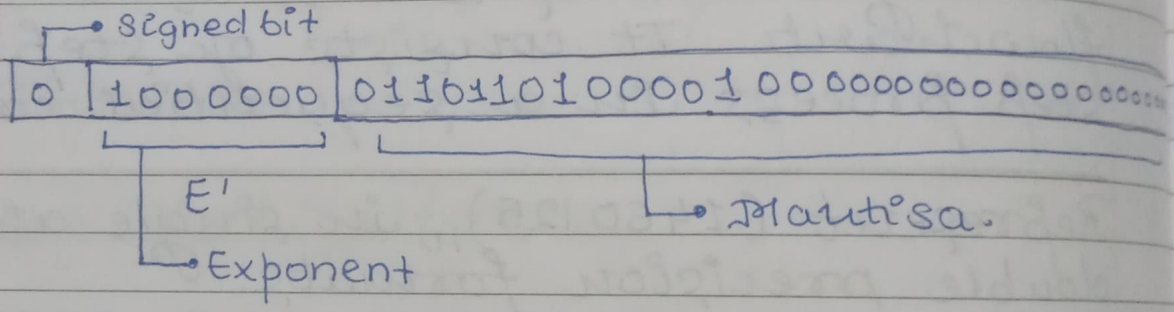




Double Precision Format.

$(1460.125)_{10} = (1.0110110100001 \times 2^{10})_2$

$E' = 1023 + E = (1033)_{10} = (10000001001)_2$



6. What do you mean by datatype? describe about different datatypes in C programming.

Datatype defines the set of values that a data item can take and operation that can be performed on an item

Different datatypes in C programming are

Basic datatypes : float, int, double, char.
Derrived datatypes: Union, structure, array etc.
Enumerated datatypes: Enum.
Void datatypes : Empty value.
Bool type : true or false.

Datatype modifiers. long, short, signed unsigned.

---

7. What do you mean by storage class? explain different storage class with suitable examples.

Storage classes are used to describe the features of variable or functions. These features basically include the scope, visibility, life time to trace existance of a variable during a program.

1. auto

This is a default storage class for a variables. It basically declairs a automatic variable which is recreated each time function is executed, it is local to a function

2. Extern.

This storage class defines a variable declairs a variable that is def either elsewhere in another function or in another file of the project.

3. Static.

Static storage class emplies that the variable is having a prop of preserving their value even after they are out of their sc

4. Register

This storage class tries to store the variable declared after it into the register of microprocessor for faster execution if a free register is available.

---

8 What is switch statement? write syntax and explain how it is different from if statement.

A switch is a multiway decision statement that is simplified version of an if-else-if block

Syntax:

```
Switch (variable)
{
  case value 1 :
     statement block 1;
     break;
  case value 2 :
     statement block 2;
     break;
  default :
     statement ;
  break;
}
```