∴ Important Question Bank :-

Short Answer Questions : (2 marks each) (50 words)

(i) Define Unilateral and Bilateral elements.

ns:- **Unilateral element :-** If the characteristics/ Property of an element changes with change in direction of current is called Unilateral element. eg:- Diode.

Bilateral element :- If the characteristics/ property of an element does not change with change in direction of current is called Bilateral element. eg:- Resistors

(ii) Define Active and Passive elements.

ns:- **Active elements :-** The elements that are capable of delivering electrical energy are called Active elements. eg:- Voltage source, Current source.

Passive elements :- The elements that receive (dissipates/ stores) electrical energy are called Passive elements. eg:- R, L, C.

(iii) Define Linear and Non Linear elements.

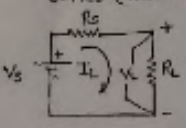
Ans:- **Linear elements :-** The elements that follow ohm's law, principle of superposition and Homogeinity are called Linear elements. There exists sign of $\propto$, $\frac{d}{dt}$ and $\int$ between V and I. eg:- R, L, C.

Non-Linear elements :- The elements that do not follow ohm's law, principle of superposition and Homogeinity are called Non linear elements. The graph between V and I is non-linear eg:- Diode, Transistor.

Q.2- Define Ideal and practical voltage and current source.

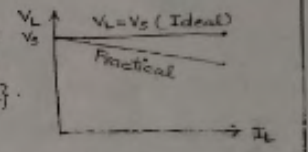
Ans:- Ideal voltage source provides constant voltage to the load inrespective of current flowing in it. It's internal resistance $R_s = 0$.

Practical voltage source has some finite resistance in series (internal R_s).



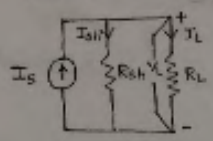
$V_L = V_s$ (Ideal)

$V_L = V_s - (I_L R_s)$ {Practical}.



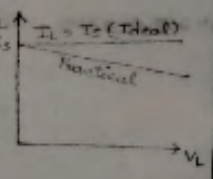
Ideal current source provides constant current to the load inrespective of voltage across it. It's internal resistance $R_{sh} = \infty$.

Practical current source has some finite internal resistance in shunt.



$I_L = I_s$ (Ideal)

$I_L = I_s - I_{sh}$ {Practical}



Unit : 1

Q1 Discuss the major components of a digital computer with suitable block diagram. Also discuss the function of each component? (2015-16, 2016-17, 2017-18, 2018-19, 2019-20, 2020-21)

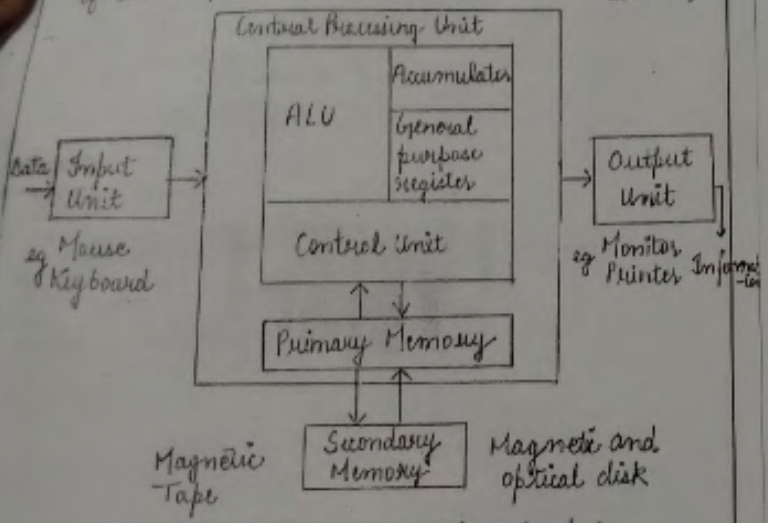


fig: Block diagram of digital Computer

1) Input unit :- It links the external environment with the computer system.
- All input device must convert the input data into the binary codes.
- Example: Keyboard, Mouse etc

2. Storage Unit: The storage unit of a computer holds data and instruction that are entered through the input unit, before they processed.
- It stores programs, data as well as intermediate results and results for output.
- Its main function is to store information.

a) Primary storage: Main Memory
b) Secondary storage: Auxiliary Memory
c) Cache Memory:
- It is a high speed, expensive memory unit.
- It placed between the processor & primary memory to reduce the mismatch between the speed of two unit.

3. Central Processing Unit:
- The control unit and arithmetic logical of computer are together known as central processing unit (CPU).
- The CPU is brain of computer.
- Perform: calculation, take decisions & control all unit

a) Control unit
- It manages and coordinates the entire computer system and synchronizes its working.
- Referred as "Central Nervous System" or "Brain of the computer".

b) Arithmetic & logical Unit :

It perform arithmetic and logical operations such as addition, subtraction, AND, OR, NOT etc

4) Output Unit :-

- It performs the reverse operation of that of an input unit.
- It supplies information obtained from processing to outside world.
- Example : Monitor, Printer etc

Q: Draw the memory hierarchical structure of a computer system ?

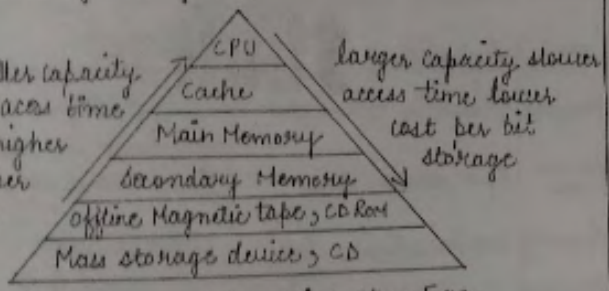
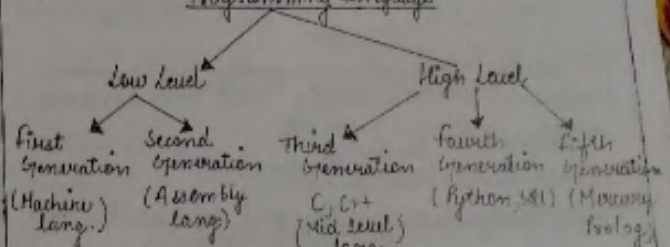


fig: Memory Hierachical structure

Q Brief the generations of the programming language with example or Differentiate between low level language and high level language. (2018-19, 2020-21)

Ans:



Low Level Languages

- It is a machine friendly language.
- Faster than high level language
- Memory efficient
- No need of translator except assembler for AL

Disadvantage

- Difficult to learn.
- They are Machine dependent and are not portable
- Debugging & maintenance is difficult.
- They are more error prone

eg: Machine & Assembly language

High Level Language

- It is programmer friendly language.
- comparatively slower
- Less efficient
- Compiler & Interpreter is needed

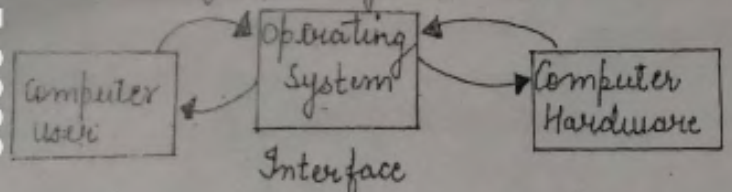
Advantage

- Easy to learn.
- They are machine independent and portable.
- Debugging & maintenance is comparatively easier
- They are less error prone.

eg: C, C++, Python etc

Q What is operating system? What are the various component of operating system. Describe the functionalities of operating System? (2015-16, 2018-19)

Ans:



Interface

An operating system (OS) is a system program which provides an interface between user and hardware.
Example: Windows, LINUX etc

Components of Operating System (OS):

- Process Management: Creating and deleting both user and system process.
- Main Memory Management: Keep track of which part of memory are currently being used and by whom.
- Secondary Memory Management: Allocation & deallocation of memory.
- Input/Output Management: It includes buffering caching & spooling.
- File Management: Creating & deleting files and directories.

Function of (OS):

- Program Execution: The system is able to load a program into memory & to run that program.

- Input/Output Operation: A running program require input and output through a input device.
- File System Manipulation: To create, delete, read & write file.
- Resource Allocation: Program needs resource to complete the task, it allocates resource to the running processes.
- Security: Protection against unauthorized access is ensure by OS.
- Accounting: It keeps the track of computer resources

Q: Give difference between android and window OS?

	Basis	Android	Window
1)	Graphics	Relatively optimized	Fully Fledged
2)	source code	Open source code	close source code
3)	Security	comparatively less	Slightly more
4)	Application	for mobile	for PC
5)	Developer	Google	Microsoft

Q What is algorithm? write the characteristics of an algorithm?

Ans: A step by step method of solving a problem or making decision is termed as algorithm.

Characteristics of Algorithm:

1) Input:- An algorithm has zero (0) or more inputs.
2) Output:- An algorithm has one or more outputs.
3) Finiteness:- An algorithm must terminate after a finite number of step.
4) Definitness:- Each step of an algo must be precisely defined.
5) Effectiveness:- An algorithm is generally expected to be effective.

Example: on next page

Advantage:

- Easy to understand.
- Every step have its own logic.
- It helps to broken down the problem into smaller pieces or steps.
- No dependency on any programing Language.

Disadvantage:

- Time consuming.
- No standard format.
- Algo. is not a computer program it is rather a concept of how a program should be.

Q What is flowchart? Draw a flowchart to find maximum of three numbers and sum and average of three number also. (2016-17, 2015-16)

Ans: It is a pictorial representation of the sequence of logical steps of a program.

Notations of Flow Chart:

Start/Stop	used to show starting & ending of flow chart.
Process	It indicates the logic.
Input/output	used for denoting input & output
Decision	used in decision statement
Arrow	show relationship b/w different shapes.

Advantage:

- Excellent way of representing the logic of program.
- Easy & Efficient to analyze problem using Flow chart.

Disadvantage:

- Flow chart can be complex when the logic is quite complicated
- Drawing flow chart is time consuming

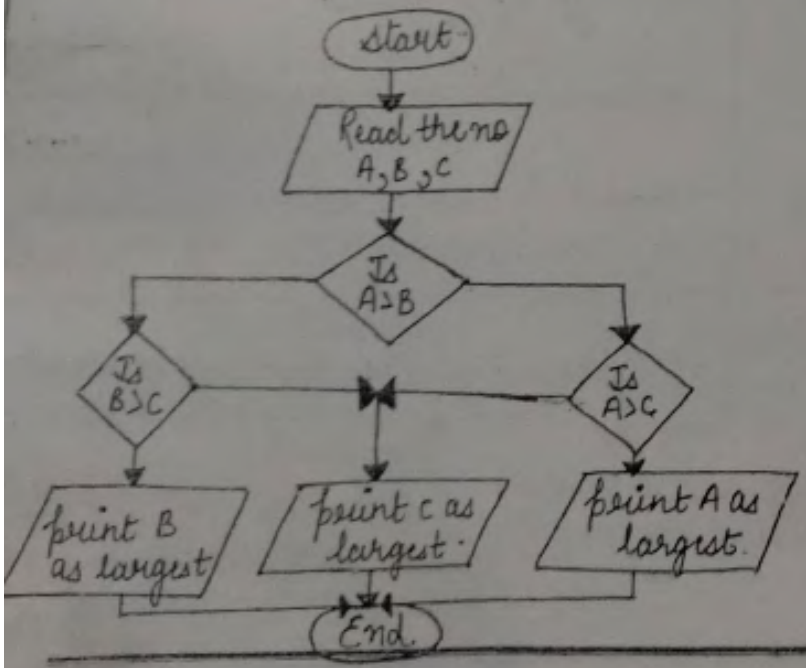
Examples of Flow chart & Algorithm

i) To find maximum of three numbers

Algorithm

① Start
② Read the three numbers to be compared as A, B & C.
③ Check if A is greater than B
3(i) If true, then check if A is greater than C.
3(a) If true, print A as greatest number
3(b) If false, print C as greatest number
3(ii) If false then check if B is greater than C.
3(a) If true, print B as greatest number
3(b) If false, print C as greatest number
④ End

Flow chart

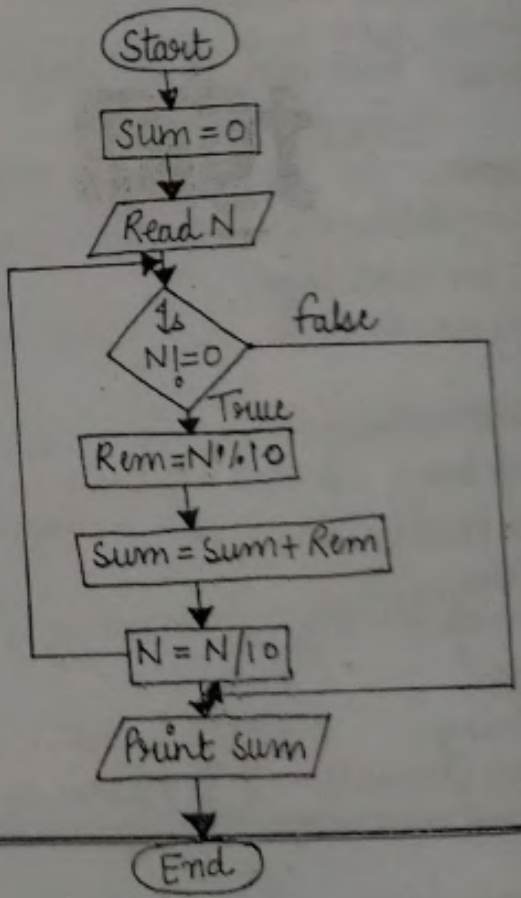


(ii) To find sum of digit of an integer number.

Algorithm:

① Start
② Input N
③ Sum = 0
④ while(N != 0)
Rem = N % 10
Sum = Sum + Rem
N = N / 10
⑤ print Sum
⑥ End

flow chart:

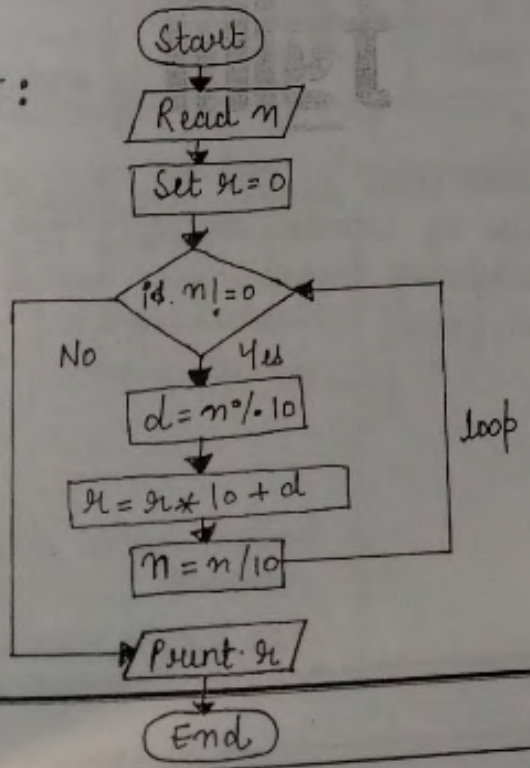


(iii) To reverse an integer number by the user.

Algorithm:

1. Start
2. Input number (n)
3. reverse (r) = 0
4. Check number (n != 0) True
5. digit (d) = n % 10
6. reverse (r) = r * 10 + d;
7. n = n / 10
8. repeat step 3 until (n != 0) False
9. write reverse number
10. End

Flowchart:



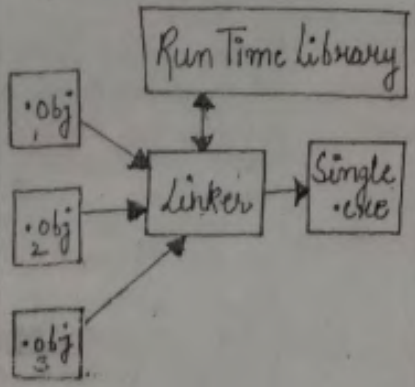
Q Difference between (i) Compiler and Interpreter
(ii) Linker and Loader

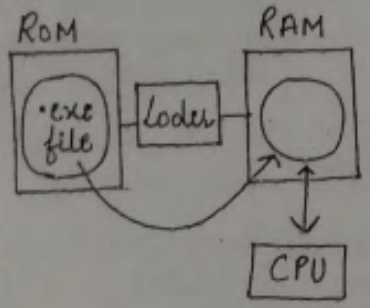
Basis	Compiler	Interpreter
Definition	It scans the entire program and translate it into machine code (as whole).	Translates program one statement at a time (line by line).
Error display	Display the syntax errors as a whole.	Display the syntax error line by line.
Execution time	The overall execution time is faster.	The overall execution time is slower.
Debugging	Debugging is hard	Debugging is easy
.exe file	.exe file is created	.exe file is not created
example	Programing Lang: C, C++ uses compiler (Turbo C, Gcc)	Programing Lang: python; java uses interpreter.

Basis	Linker	Loader
Basic	It generates the executable module of a source program.	It loads the executable module to the main memory.
Input	It takes object code generated by an assembler	It takes executable module generated by a linker.
function	It resource the unresolved function with the help of run time library and combine multiple object file into a single executable file.	It allocates the address to an executable module in memory for execution

Type: Linkage Editor, Dynamic Linker

Approach: Absolute Loading, Relocatable loading and dynamic run time Loading.





Q: What is the difference between .obj and .exe file?

Ans: • An object file is a file generated after compiling the source code.

• While an executable file is a file generated after linking a set of object file together using a linker.

Q: Write difference between syntax, run-time, logical error & Input data error?

Syntax Error	Runtime Error	Logical Error
• The error which arises due to voilation of any rule of C language.	• The error which occur during execution of program.	• The error which occur due to usage of wrong formula or logic
• These error are identified by compiler	• These error are identified during linking / run time	• These error are not identify by compiler or linker
• missing of semi-colon (;)	• print instead of printf	• Like using 2*3.14*r as area of circle

Input Error:

The error which occur during the data entry process, because of entering wrong input values during execution of program are called as input data error.

Q Explain basic structure of C Program?

Ans:
- Documentation (Documentation section)
- Link section (Copy of header files is inserted into code befor compilation)
- Definition (#define allows us to use constants in our code. It replace all the constant with its value in the code.)
- Global Declaration (Includes declaration of global variable, function etc)
- Main () function (The execution starts from the main function.
  - It is mandatory to include main() function in every C program.
- User defined function if any.

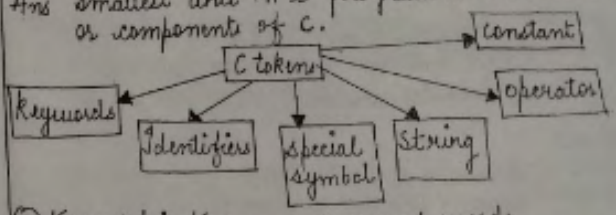
Example

```
// Addition of two numbers   // Documentation
#include<stdio.h> // Link section

void add (); // Global Declaration
void main() // main() function
{
add();
}
void add()   // user defined function
{ int a,b,c;
scanf("%d%d",&a,&b);
c=a+b;
printf("sum=%d",c);
}
```


Q Define tokens in C language? or components of C?

Ans smallest unit in a program is know as C tokens or components of C.



(1) Keywords:
- Known as reserved words
- Meaning is already defined in C library

eg: int, char, float etc.

(2) Identifier: is a group of character that is used for naming the variable, array, function, pointer etc.

eg: a, _sum, a_1 etc

(3) Operator:
- Special symbols used to perform specific function.
- Meaning is already know to a.c compiles
- There are 45 operator

Types:
- → Unary operator (Works on one operand eg. ++a, -a)
- → Binary operator (Works on Two operand eg. a+b, a*b etc)
- → Ternary operator (Works on Three operand eg ? :)

(4) String: It is the collection of character, digits or special symbols enclosed in double quotes terminated by null character like "Hello".

(5) Special symbols: Have a special meaning and cannot be used for another purpose eg: [], (); # etc

⑥ Constant: Constant are those whose value remain unchanged.

Types → Integer constant (-3, 4, 2894 etc)
→ Real constant (-2.53, 3.5E28)
→ character constant ('a', 'b' etc)

Example:

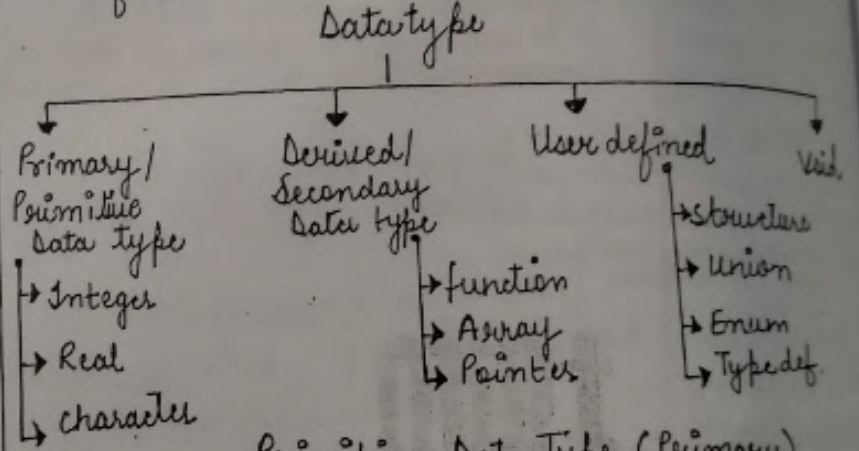
```
void main()
{ int r;
  float ac, cc;
  clrscr();
  printf("Enter Radius");
  scanf("%d", &r);
  ac = 3.14 * r * r;
  cc = 2*3.14 * r;
  printf("area = %f circumference = %f", ac, cc);
  getch();
}
```

C Tokens

1.	Keywords	void, int, float
2.	Identifiers	r, ac, cc, printf, scanf
3	Constant	3.14, 2
4	Operators	=, *
5	String	"Enter Radius"
6	Special Symbol	{ }, ;

Q Define data type in C. Discuss in terms of memory size, format specifier and range? / Primitive Data

Ans: Data type is used to declare the variable. It determine the type of value and the range of value that can be stored inside a variable.



Primitive Data Type (Primary)

Data Type	format specifier	Size	Range
• char	%c	1 byte	-128 to +127
• int	%d	2 byte	-32768 to 32767
• short int	%d	2 byte	-32768 to 32767
• long int	%Ld	4 byte	-2,147,483,648 to 2,147,483,647
• unsigned int	%u	2 byte	0 to 65,535
• unsigned short int	%u	2 byte	0 to 65,535
• Unsigned long int	%Lu	4 byte	0 to 4,294,967,295
• float	%f	4 byte	3.4E-38 to 3.4E+38
• double	%lf	8 byte	1.7E-308 to 1.7E+308
• long double	%Lf	10 byte	3.4E-4932 to 1.1E+4932

UNIT-2

Ques What is an operator ? List all the operators used in C. Give example.

Ans: Operators are special symbols whose meaning is already know to C compiler. There are 45 operators in C classified as :-

① Unary Operators :-
Operators that need only one operand to complete its task is termed as unary operator.
eg : (!) logical not (~) complement

② Binary operators :-
Operator that need two operand to complete its task is termed as Binary operators.
ex: (+) Addition , (*) Multiplication

③ Ternary Operators :-
operator that need three operand to perform its task is termed as conditional operator.
ex : Condition ? True Block : False Block

General Categories of Operator

① Arithmetic Operators : Used to carryout basic arithmetic operations

operator	Meaning	Example
+	Addition	1+2 = 3
−	Substraction	3-2 = 1
*	Multiplication	2*2 = 4
/	Division	2/2 = 1
%	Modulo Division	10%3 = 1

(2) Relational Operator:

The operator which are used to form conditions for comparing two operands are termed as relational operators.

Operator	Meaning	Example	Return value
<	is less than	3<5	1
<=	is less than or equal to	4<=2	0
>	is greater than	7>5	1
>=	is greater than or equal to	3>=5	0
==	Equal to	6==6	1
!=	not equal to	5!=5	0

(3) Logical Operators:

The operators which are used to combine the result of two or more conditions are termed as logical operators.

Operator	Meaning	Example	Return value
&&	Logical AND	(9>2)&&(6>4)	1
||	Logical OR	(9>2)||(3>4)	1
!	Logical Not	!4	0

Truth Table:

a	b	a&&b	a||b	!a	!b
1	1	1	1	0	0
1	0	0	1	0	1
0	1	0	1	1	0
0	0	0	0	1	1

0 → False
1 → True

Bitwise Operator:

The operators which are used to perform operation at bit level are termed as bitwise operator.
There are 6 bitwise operators used in 'C'.

Operator	Meaning	Example	Return Value
&	Bitwise AND	5&7	5
|	Bitwise OR	5|7	7
^	Bitwise XOR	5^7	2
~	Complement	~5	-6
<<	Left shift	4<<2	16
>>	Right shift	16>>1	8

Truth Table:

a	b	a&b	a|b	a^b
0	1	0	1	1
0	0	0	0	0
1	1	1	1	0
1	0	0	1	1

(~) Complement: (Replace all 0 by 1, 1 by 0)

Q Differentiate between precedence and associativity

Precedence	Associativity
• It describe the order of execution of operators in an expression having different priority.	It describe the order of execution of operators in an expression having same priority i.e LHS to RHS or vice versa
• The highest precedence operator is applied first, followed by the next highest and so on.	• It tells how operators of same precedence are grouped and how the expression will be evaluated.
• ex * has high precedence than +.	• Arithmetic operator has left associativity.

Q What is type conversion? Explain with example
or
Differentiate implicit and explicit conversion with example.?

Ans: When one data type converted into another data type by a compiler is called as type conversion.

→ Type conversion (Implicit Type)
→ Type casting (Explicit Type)

Implicit Type Conversion	Type casting / Explicit Type conversion
• It is automatic type conversion.	• It is manual type conversion.
• Performed in lower to higher data type only.	• It can be performed in any order.
• no need for a casting operator.	• casting operator is needed.
• It is done at the compile time.	• It takes place during the program design by programmer.
• float i; i = 5.0/2; = 2.5 Here 5.0 belong to double data type 2 belongs to int data type so 2 get converted into double before execution	• float i; i = (int) 5.0/2 = 2.0 5.0 belongs to double data type 2 belongs to int data type But 5.0 get converted into int before execution

Write short note on various decision / selection control instruction use in C? or
What is the difference between if else statement & conditional operator?
What is the difference between if else statement & switch statement?

Ans:

- The control instruction which help to check some instruction, condition and take decision about what to execute is know as decision / selection control instruction.

If else	Conditional operator	Switch
• It used where multiple statement has multiple choice. • In if else, else can be optional • Either if statement will be executed or else statement is executed. • It works with condition of any data type. • It is decision control instruction. Syntax: if (condition) { true block ; } else { false block ;	• It used where simple statement are within expression. • All 3 expression are necessary. • If exp.1 is true then exp.2 is executed otherwise exp.3 is executed. • It works with condition of any data type • It is decision control instruction. Syntax: exp 1 ? exp 2 : exp 3	• It used where single expression has multiple choice. break and • In switch, default can be optional. • Switch statement execute one case after another till a break statement encounter or the end of switch is reached. • It works only with character expression and integer datatype. • It is case based instruction. Syntax: switch (n) { case 1: stm x ; break ; case 2: stm y ; break ; default : stm z ;

Example :- WAP to check year is leap year a not.

```
#include<stdio.h>
#include<conio.h>
void main()
{ int y;
  clrscr();
  printf("\n Enter year :\n");
  scanf("%d",&y);
  if(y%400==0||(y%4==0&&y%100!=0))
  { printf("\n Leap year\n");
  }
  else
  { printf("\n Not Leap year\n");
  }
  getch();
}
```

WAP to find Largest of three number

```
#include<stdio.h>
#include<conio.h>
void main()
{ int a,b,c;
  clrscr();
  printf("Enter value of a,b&c");
  scanf("%d%d%d",&a,&b,&c);
```



```
if(a>b)
{ if(a>c)
  printf("\n%d is greater number\n",a);
  else
  printf("\n%d is greates number\n",c);
}
else
{ if(b>c)
  printf("\n %d is greates number\n",b);
  else
  printf("\n%d is greates number\n",c);
}
  getch();
}
```

WAP to check whether the given character is upper case, lower case, numeric or symbol. (2014-15, 2017-18)

```
#include<stdio.h>
#include<conio.h>
void main()
{ char n;
  clrscr();
  printf("\n Enter Character :\n");
```

```
scanf ("%c", &n);
if (n>=65 && n<=90)
printf ("\n Upper case character \n");
else if (n>=97 && n<=122)
printf ("\n Lower case character \n");
else if (n>=48 && n<=57)
printf ("\n Numeric character \n");
else
printf ("\n Symbol \n");
getch();
}
```

Q. Write the concept of switch statement. Write a program to draw the calculator using switch statement in C language? (2016-17, 2017-18)

Ans: The control statement that allows us to make a decision from the number of choices is called the switch case statement.

WAP to design calculator

```
#include <stdio.h>
#include <conio.h>
void main()
{ int a,b,c, ch;
clrscr();
printf ("\n Enter 2 Numbers :\n");
scanf ("%d%d", &a, &b);
```



```
printf ("\n Enter 1 for addition : \n");
printf ("\n Enter 2 for subtraction : \n");
printf ("\n Enter 3 for Multiply : \n");
printf ("\n Enter 4 for Division : \n");
printf ("\n Now Enter Your choice : \n");
scanf ("%d", &ch);
switch (ch)
{ case 1: c=a+b;
        printf("Sum is %d", c);
        break;
case 2: c=a-b;
        printf ("Sub is %d", c);
        break;
case 3: c=a*b;
        printf ("Mul is %d", c);
        break;
case 4: c=a/b;
        printf ("Div is %d", c);
        break;
default : printf ("\n Wrong input \n");
}
getch();
}
```

Q: What are the rules for using switch statement

Ans: Rules of switch statement are:

- The switch case must be constant or a constant expression.
- The case lable must be valid and unique.
- Case lable must end with colon (:) and each statement with semi colon (;)
- Case lable can be int or char constant but it can't be float.
- Using break and default is optional.

Q Find the output

```
main()
{ int x=4;
  x = x<<2
  printf("%d", x);
}
```

To Remember:

Use this formula to calculate

(•) << (left shift)
= num. * $2^{(\text{left shift})}$

(•) >> (right shift)
= num. / $2^{(\text{right shift})}$

Ans: output
16

Q Find value of y
y = -16 % -2

Ans: y = -0
Here % (Modulo Division) operator gives remainder.
And it take the sign of numerator

Q find output

```
int main()
{ int x=2;
  x = x<<1;
  printf("%d", x);
}
```

Ans: output
4

Q Find Output

```
main()
{ printf ("\n ab");
  printf ("\b si");
  printf ("\r ha");
}
```

Ans: Output

hai

Here answer is based on backslash characters.

Q Distinguish main() & void main(void)

main()	void main(void)
* No return type is given	* void is used as return type
* By default return type is int	* It will not return any value
* Need return() statement	* No need of return statement.

Q What do you mean by mixed operand/expression?

Ans: If operand in an expression contain both integers & real constant or variable this is termed as mixed operand.

UNIT-3

/ : Iteration statement

Q. What is looping control instruction? Describe the type of looping statement in 'C' with necessary syntax? or (2014-15, 2018-19)

Differentiate between do-while and while loop?

Ans: Looping can be defined as repeating the same process multiple times until a specific condition satisfies.

Control statement
- Decision
- Iterative / Loop
  - for loop
  - while loop
  - do while loop

1) **for Loop:**

It is used to execute a set of statements repeatedly until a particular condition is satisfies.

Syntax:

```
for (initialization ① ; condition ② ; updation ③ inc/dec )
{
    stm block or body of loop -④
}
stm x; -⑤
```

Order of execution:

① ② ④ ③ ② ④ ③ ② ⑤ => (out of the loop)

(② True, ② True, ② False)

B. Tech I Year [Subject Name: Programming for Problem Solving]

while loop	do while loop
① It is entry control loop.	It is exit control loop.
② Here condition is checked first & if condition is true only then the block will be executed.	In do-while, block is executed first & then the condition is checked so block will execute at least 1 time even condition found false.
③ In while loop, if the condition is found true for 'n' times the block will execute 'n' times.	In do while loop, if the condition is true for n time the block will execute (n+1) times.
④ We do not put semicolon after while (condition) statement.	We put semicolon after while (condition) statement.

⑤ Syntax (while loop):

```
initialization; -①
while( condition) -②
{
③ - stm block or body of loop;
④ - updation (inc/dec);
}
stm x; -⑤
```

Order of execution:

① ② ③ ④ ② ③ ④ ② ⑤

(② True, ② True, ② false → ⑤ (out of loop))

Syntax (do while loop):

```
initialization; -①
do
{
    stm block or body of loop; -②
    updation (inc/dec); -③
}
while (condition); -④
stm x; -⑤
```

Order of execution:

① ② ③ ④ ② ③ ④ ② ③ ④ ⑤

(④ True, ④ True, ④ false → ⑤ (out of loop))

Q: Write a program to check a number is prime number or not ?

```
#include <stdio.h>
#include <conio.h>
void main()
{ int n,i,f=0;
  clrscr();
printf ("\n Enter Number \n");
scanf ("%d",&n);
for (i=1; i<=n ;i++)
{ if (n%i ==0)
  { f++;
  }
}
if (f==2)
printf ("Prime number");
else
printf ("Not Prime");
getch();
}
```

Output:

```
Enter Number
7
Prime number
```


Q: WAP in C to generate fibonacci series

```
#include <stdio.h>
#include <conio.h>
void main()
{ int n,a=-1,b=1,c,i;
 printf ("\n Enter no of Terms \n");
 scanf ("%d",&n);
  for (i=1; i<=n ;i++)
  { c=a+b;
    printf ("%d",c);
     a=b;
     b=c;
  }
    getch();
  }
```

Output:

```
Enter no of Terms
5
0 1 1 2 3
```

WAP to calculate sum of fibonacci series

```
#include<stdio.h>
#include<conio.h>
void main()
{ int a=-1,b=1,n,c,i,s=0;
 clrscr();
printf("\n Enter number of Terms:\n");
scanf("%d",&n);
for(i=1;i<=n;i++)
 { c=a+b;
   s=s+c;
   a=b;
   b=c;
 }
  printf("sum of series = %d",s);
getch();
}
```

Output:

```
Enter number of Terms
5
sum of series = 7
```


Q Program to find sum of digit ?

```
#include<stdio.h>
#include<conio.h>
void main()
{ int n,m,d,s=0;
 clrscr();
printf("\n Enter Number\n");
scanf("%d",&n);
 m=n;
 while(m>0)
{ d=m%10;
  s=s+d;
  m=m/10;
}
 printf("Sum of %d = %d",n,s);
 getch();
}
```

Output

```
Enter Number
153
Sum of 153 = 9
```

Q Write a program to check whether a given number is Armstrong or not. (2015-16, 2016-17)

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{ int n, m, d, s=0;
  clrscr();
  printf("\n Enter Number \n");
  scanf("%d", &n);
  m=n;
  while(m>0)
  { d = m%10;
    s = s + d*d*d;
    m = m/10;
  }
  if(n==s)
  { printf("\n Armstrong \n");
  }
  else
  { printf("\n Not Armstrong \n");
  }
  getch();
}
```

Output:

Enter Number
153
Armstrong

Q Write a program in 'C' that will read a positive number from the keyboard & print it in reverse order?

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{ int n, m, d, r=0;
  clrscr();
  printf("\n Enter Number \n");
  scanf("%d", &n);
    m = n;
  while(m>0)
  { d = m%10;
    r = r*10 + d;
    m = m/10;
  }
  printf("Reverse of %d = %d", n, r);
  getch();
}
```

Output:

Enter Number
121
Reverse of 121 = 121

Q: Write a program to check number is pallindrome or not. The program should accept any arbitary number typed by user? (2016-17, 2017-18, 2018-19, 2019-20, 2021)

Ans:
```
#include <stdio.h>
#include <conio.h>
void main()
{ int n, m, d, r=0;
  clrscr();
  printf("\n Enter Number\n");
  scanf("%d", &n);
  m=n;
  while(m>0)
  { d=m%10;
    r=r*10+d;
    m=m/10;
  }
  if(n==r)
  { printf("\n Pallindrome\n");
  }
  else
  { printf("\n not Pallindrome\n");
  }
  getch();
}
```

Output:

Enter Number
121
Pallindrome

Q WAP to add the given terms of the following series by using loop $1! + 2! + 3! + \ldots\ldots + n!$?

Ans:
```
#include <stdio.h>
#include <conio.h>
void main()
{ int n, j, i, f, s=0;
  clrscr();
  printf("\n Enter Value of n\n");
  scanf("%d", &n);
  for(j=1; j<=n; j++)
  { f=1;
    for(i=j; i>1; i--)
    { f=f*i;
    }
    s=s+f;
  }
  printf("Sum of series = %d", s);
  getch();
}
```

Output:

Enter Value of n
3
Sum of series = 9

Q: WAP to print the following pattern ?

```
*
* *
```

Ans:
```
#include <stdio.h>
#include <conio.h>
void main()
{ int n,i,j;
clrscr();
printf("\n Enter Number of Rows : \n");
scanf("%d",&n);
for(i=1; i<=n ; i++)
{
for(j=1; j<=i;    j++)
{
printf("*");
}
printf("\n");
}
getch();
}
```

output:

```
Enter Number of Rows:
3
*
* *
* * *
```


Q: What is the difference between break & continue in 'C'?

Ans: Break

- It is used to come out of the loop.
- break will skip all the coming iteration
- It is used to come out of switch statement

example:
```
int i;
for(i=1;i<=3;i++)
{ if(i==2)
  break;
  printf("%d",i);
}
```

output: 1

Continue

- It is used to come in the begining of loop.
- while continue will skip only current iteration.
- It is not used in switch statement.

example:
```
int i;
for(i=1;i<=3;i++)
{ if(i==2)
  continue;
  printf("%d",i);
}
```

output: 1 3

Q: Give loop statement to print
-8 -6 -4 -2 0 2 4 6 8

Ans:
```
for (i=-8; i<=8; i=i+2)
{ printf ("%d", i);
}
```

Q: Give the for loop statement

Ans:
```
for (i=1; i<=32; i=i*2)
{ printf ("%d", i);
}
```


Q: What is the meaning of prototype of a function? ('AND') Explain function declaration and definition of a function with example? (2015, 16, 17, 18, 19, 20)

Ans: A function prototype is simply the declaration of a function that specifies function name, parameter and return type.

Declaration of function:

Syntax:
returntype functionname (parameter type); ← optional
eg 1) int fun (int, float);
2) void fun();

Calling of function:

Syntax:
Variable = functionname (parameter value); ← optional
↳ optional
eg 1) x = fun (a, b);
2) fun();

Definition of function:

Syntax: Return type functionname (parameter declaration) ← optional
{ body of function
return (value); → optional
}

Q Difference between Actual & Formal Argument/Parameter

Actual Argument	formal Argument (2014-15, 16-17, 17-18)
The variable which are used during function call by the calling function are know as actual parameters or argument.	The variable which are used in function definition to collect the value of actual argument in called function are know as formal parameters or argument.

Example to show Actual & Formal Argument/Parameter

```
void fun (int, int);    // function declaration
void main()
{
  body of main function
  fun (a,b);            // function calling
                  → Actual Argument/Parameter
}
void fun (int p, int q)          // function definition
                  → Formal Argument/Parameter
{
  body of user defined function
}
```


Q Difference between call by value and call by reference with example? / explain function Parameter Passing method with example?

Call by Value	Call by Reference/Address
value of actual argument is pass to the function.	Address of actual argument is pass to function.
Pointer is not used.	Pointer variable is used
If value of formal argument is changed, there is no change in value of actual argument	If value of formal argument changes, there is a change in value of actual argument.
Ex WAP to swap the value of two variable using call by value	WAP to swap the value of two variable using call by reference/address

```
#include <stdio.h>
#include <conio.h>
void swap (int, int);
void main()
{ int a,b;
  clrscr();
printf("\n Enter two numbers:");
scanf("%d%d", &a, &b);
swap (a,b);
getch();
}
void swap (int a, int b)
```

```
#include <stdio.h>
#include <conio.h>
void swap (int*, int*);
void main()
{ int a,b;
  clrscr();
printf("\n Enter two numbers");
scanf("%d%d", &a, &b);
swap (&a, &b);
getch();
}
void swap (int*a, int*b)
```

```
printf("Before swapping \n");
printf("a=%d b=%d",a,b);
a = a+b;
b = a-b;
a = a-b;
printf("After swapping \n");
printf("a=%d b=%d",a,b);
```

```
{
printf("Before swapping \n");
printf("a=%d b=%d",*a,*b);
*a = *a + *b;
*b = *a - *b;
*a = *a - *b;
printf("After swapping \n");
printf("a=%d b=%d",*a,*b);
}
```

Q: What are the main principle of recursion? Explain in detail? (2014-15, 2015-16)

Ans: A function is calling itself again and again to solve a particular problem is called recursion.

Principle of Recursion:

- A recursive program must call itself recursively.
- A recursive program must have a base condition
- A recursive program must change its state & move towards the base condition.

In recursion, function is calling itself again & again so we need a condition to stop the ifinite recursive calls, this condition is know as Base condition.

Q: WAP to Find factorial of a given number without using recursion

```
Ans: #include <stdio.h>
#include <conio.h>
int fact (int);
void main()
{ int n, f;
  clrscr();
printf ("Enter no \n");
scanf ("%d", &n);
  f = fact (n);
  printf ("factorial = %d", f);
getch();
}
int fact (int n)
{
int f=1, i;
for (i=1 ; i<=n ; i++)
{ f = f*i;
}
return (f);
}
```

output:

```
Enter no
5
factorial = 120
```

WAP to find factorial of a given number using recursion

```
Ans: #include <stdio.h>
#include <conio.h>
int fact (int);
void main()
{ int n, f;
  clrscr();
printf("Enter no\n");
scanf("%d", &n);
  f = fact (n);
printf("factorial = %d", f);
getch();
}
int fact (int n)
{ if (n==1)
  return (1);
  else
return (n * fact (n-1));
}
```

output:

```
Enter no
5
factorial = 120
```

Write a C program to generate fibonacci series using recursion

```
#include <stdio.h>
#include <conio.h>
int fib(int);
void main()
{
 int n,i,x;
 clrscr();
 printf("\n Enter no of Terms \n");
 scanf("%d",&n);
 for(i=1; i<=n; i=i+1)
 {
    x= fib(i)
    printf("%d \t",x);
 }
 getch();
}
int fib(int y)
{ if(y==1)
    return(0);
  else if(y==2)
    return(1);
  else
  return(fib(y-1) + fib(y-2));
}
```

output:

```
Enter no of Terms
5
0 1 1 2 3
```


Q: What is storage class? Describe automatic, register, static and external with neat syntax (2014-18)

Q: what do you mean by scope & life time of a variable (2015-16, 2018-19)

Q: Differentiate b/w static & register class in c language (2016-17)

Ans: storage class in c decides the part of storage to allocate memory for a variable.

Declaration of Variable:

Syntax:

storage class datatype variable name;

Storage class	Storage (keyword)	Default Initial Value	Scope	Life	Declaration Example
1) Automatic: auto	memory	garbage	local	the block where declared	auto int a,b;
2) Register: register	register	garbage	local	with in the block where declared	register int a,b;
3) Static: static	memory	zero	local	Value persists b/w different function call	static int a,b;
4) External: extern	memory	zero	global	Through out the program	extern int a,b;

UNIT-4

Q: What are subscript ? How they are specified ? (2018-19)

Ans: Array subscript is an integer type constant or variable name whose value range 0 to n-1, where n is the total number of element in array.

Syntax : Array name [subscript]

eg: int a [5];

Q: What is array ? In which situation array is advantageous over linked list ? (2018-19)

Ans: An array is the collection of similar datatype elements stored at contiguous memory location.

syntax : datatype Arrayname [array size];

eg : int Marks [5];

Advantage of array over linked list :-

• Array allow random access to elements.	• Link list only allow sequential acess.
• Array takes less storage for storing elements.	• Link list take more storage for storing elements.

what do you mean by sorting? write a program in C to sort 'n' positive integers using bubble sort (2019-20)

Or

Define sorting algorithm with example. (2020-21)

Ans: Sorting: It is the process of arranging data in order (in increasing or decreasing).

example: suppose we have an array

98, 7, 15, 56, 32

After sorting in ascending order

7, 15, 32, 56, 98

Types of Sorting Algorithm:

① Bubble sort.
② Selection sort.

// Program of Bubble sort.

```
#include<stdio.h>
#include<conio.h>
void main()
{int a[25], n, i, j, t;
printf("Enter no of elements \n");
scanf("%d", &n);
printf("Enter array elements \n");
for(i=0; i<n; i++)
```



```
{scanf("%d", &a[i]);
}
for(i=0; i<n; i++)
{for(j=0; j<=n-i-1; j++)
{if(a[j]>a[j+1])
{t=a[j];
a[j]=a[j+1];
a[j+1]=t;
}
}
}
printf("\n Sorted Array is: \n");
for(i=0; i<n; i++)
{
printf("%d", a[i]);
}
getch();
}
```

Q WAP to sort elements using selection sort? (2018-19)

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a[25], n, i, j, t;
clrscr();
printf("\n Enter no of elements \n");
scanf("%d", &n)
printf("\n Enter each elements \n");
for(i=0; i<n; i++)
{ scanf("%d", &a[i]);
}
for(i=0; i<n; i++)
{ for(j=0; j<n; j++)
  { if(a[i] > a[j])
    { t = a[i]
      a[i] = a[j];
      a[j] = t;
    }
}}
printf("\n sorted Array is \n");
  for(i=0; i<n; i++)
```



```
printf("%d", a[i]);
getch();
}
```

Q: Differentiate between linear search and binary search? (2018-19)

Basis	Linear Search	Binary Search
Time complexity	$O(n)$	$O(\log_2 n)$
Prerequisite for an array	Not required	Array must be in sorted order
Best case time	First element $O(1)$	Center element $O(1)$
Worst case for n number of elements	n comparisions are required	can conclude after only $\log_2 n$ comparisions
Can be implemented on	Array & linked list	Can't be directly implemented on linked list
Insert operation	New elements can be inserted at the end of the list	Processing is required to insert element at its proper place in sorted list
Usefulness	Easy to use	Tricky Algo
Lines of codes	Less	More

Q: Explain binary search technique for searching an item in a given array.

Ans:
```
#include <stdio.h>
#include <conio.h>
void main()
{ int i,e,n,f,l,mid,a[30];
  clrscr();
  printf("Enter no of elements\n");
  scanf("%d",&n);
  printf("Enter array Elements\n");
  for(i=0;i<n;i++)
  { scanf("%d",&a[i]);
  }
  printf("Enter number to search\n");
  scanf("%d",&e);
  f=0;
  l=n-1;
  while(f<=l)
  { mid=f+(l-f)/2;
    if(e==a[mid])
    { printf("%d is present at location %d",e,mid+1);
      break;
    }
```



```
    else if(e>a[mid])
    f=mid+1;
    else
    l=mid-1;
  }
  if(f>l)
  printf("%d is not present",e);
  getch();
}
```

Q: Explain Linear search technique for searching an item in a given array?

```
#include <stdio.h>
#include <conio.h>
void main()
{ int i,e,n,a[30];
  printf("Enter no of elements\n");
  scanf("%d",&n);
  printf("Enter array Elements\n");
  for(i=0;i<n;i++)
  { scanf("%d",&a[i]);
  }
  printf("Enter number to search\n");
```

```
scanf("%d", &e);
for (i=0; i<n; i++)
{ if (a[i]==e)
  { printf("%d is present at location %d", e, i+1);
    break;
  }
}
if (i==n)
printf("%d is not present", e);
getch();
}
```


Q: Write a program to find odd & even number from the array elements and its count.

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{ int n, i, even=0, odd=0, a[100];
printf("Enter no of the Elements \n");
scanf("%d", &n);
printf("Enter Elements \n");
for(i=0; i<n; i++)
scanf("%d", &a[i]);
for(i=0; i<n; i++)
{ if(a[i]%2==0)
  { printf("%d is even \n", a[i]);
    even++;
  }
  else
  { printf("%d is odd \n", a[i]);
    odd++;
  }
}
printf("No of odd number are %d \n", odd);
printf("No of Even number are %d \n", even);
getch();
}
```

) write a program to multiply two matrices of dimension 3x3 and store the result in another matrix (write comments also at appropriate places) (2018-19).

```
ns #include <stdio.h>  #include <conio.h>
   void main()
{ int a[3][3], b[3][3], c[3][3], i, j, k;
printf("Enter elements of first matrix\n");
for(i=0; i<3; i++) // Take first matrix from user
{ for(j=0; j<3; j++)
{ scanf("%d", &a[i][j])
}
}
printf("Enter elements of second matrix\n");
for(i=0; i<3; i++) // Take second matrix from user
{ for(j=0; j<3; j++)
  { scanf("%d", &b[i][j])
  }
}
for(i=0; i<3; i++)      //calculating product of matrix
{ for(j=0; j<3; j++)
  { c[i][j]=0
    for(k=0; k<3; k++)
```



```
{ c[i][j] = c[i][j] + a[i][k] * b[k][j];
  }
}
printf("output matrix:\n"); // Printing output matrix
for(i=0; i<3; i++)
{ for(j=0; j<3; j++)
  { printf("%d", c[i][j]);
    }
    printf("\n");
  }
  getch();
}
```

8) write a program to add two matrices of dimension 3x3 and store the result in another matrix (2017-18)

```
Ans: #include <stdio.h>  #include <conio.h>
     void main()
{ int a[3][3], b[3][3], sum[3][3], i, j;
  printf("Enter elements of first matrix\n");
  for(i=0; i<3; i++)
  { for(j=0; j<3; j++)
    { scanf("%d", &a[i][j]);
    }
  }
```

```
printf("Enter elements of second matrix\n");
for(i=0; i<3; i++)
{ for(j=0; j<3; j++)
  { scanf("%d", &b[i][j]);
  }
}
printf("output matrix is :\n");
for(i=0; i<3; i++)
{ for(j=0; j<3; j++)
  { sum[i][j] = a[i][j] + b[i][j];
  }
}
for(i=0; i<3; i++)
{ for(j=0; j<3; j++)
  { printf("%d \t", sum[i][j]);
  }
  printf("\n");
}
getch();
}
```


Q: Program to find transpose of matrix

```
#include<stdio.h>
#include<conio.h>
void main()
{ int a[3][3], m, n, i, j;
  clrscr();
  printf("\n Enter order of matrix\n");
  scanf("%d %d", &m, &n);
  printf("\n Now Enter Elements\n");
  for(i=0; i<m; i++)
  { for(j=0; j<n; j++)
    { scanf("%d", &a[i][j]);
    }
  }
  printf("\n Transpose of Matrix is\n");
  for(j=0; j<n; j++)
  { for(i=0; i<m; i++)
    { printf("%d \t", a[i][j]);
    }
    printf("\n");
  }
  getch();
}
```

Q Write a program in c to find the largest number of a 4*4 matrix (2017-18, 2016-17)

```
#include <stdio.h>
#include <conio.h>
void main()
{ int a[4][4], largest, i, j;
  clrscr();
  printf("Enter elements of 4*4 matrix");
  for (i=0; i<4; i++)
    { for (j=0; j<4; j++)
      { scanf("%d", &a[i][j]);
      }
    }
  largest = a[0][0];
  for (i=0; i<4; i++)
    { for (j=0; j<4; j++)
      { if (a[i][j] > largest)
          largest = a[i][j];
      }
    }
  printf("\n largest element is %d", largest);
  getch();
}
```


Q: What is string? (2015-16)

Ans: A string is a sequence of characters terminated with a null character '\0'.

ex: char c[] = "Hello World";

When the compiler encounters a sequence of characters enclosed in the double quotation marks, it appends a null character at the end by default.

String handling function

Q: Explain all predefined string function?

function	description	syntax
• strlen()	It is used to find the length of string	char s[10] = "hello"; int l; l = strlen(s);
• strcpy()	It is used to copy string s1 to s2.	char s1[10] = "hello", s2[10]; strcpy(s2, s1);
• strcat()	It is used to concatenate two string s1 in s2.	char s1[10] = "hello", s2[10]; strcat(s2, s1);
• strcmp()	It is used to compare two string	char s1[10] = "hai", s2[10]; int i; i = strcmp(s2, s1);
• strrev()	It is used to reverse the string	char s[10] = "hello"; strrev(s);

Q: Program to check string is pallindrome or not

Ans:
```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
char s1[10], s2[10];
clrscr();
printf("\n Enter String\n");
gets(s1);
strcpy(s2, s1);
strrev(s2);
if(strcmp(s2, s1)==0)
{
printf("\n Pallindrome\n");
}
else
{
printf("\n Not Pallindrome\n");
}
getch();
}
```


Q: Program to sort string in alphatatical order.

Ans:
```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
char a[10], t;
int n, i, j;
clrscr();
printf("\n Enter string\n");
gets(a);
n=strlen(a);
for(i=0; i<n; i++)
{
for(j=i+1; j<n; j++)
{
if(a[i]>a[j])
{
t=a[i];
a[i]=a[j];
a[j]=t;
}
}
}
printf("\n Sorted string is :");
puts(a);
getch();
}
```

am to reverse string using pointer

```
clude<stdio.h>
clude <conio.h>
clude<string.h>
 main()

 s[10], *ptr;
 i,l;
scr();
intf("\n Enter string \n");
s(s);
r = s;
= strlen(s);
 (i=0; i<l; i++)

 ptr++;
}
ptr--;
printf("\n Reverse string = ");
 for(i=l-1; i>=0; i--)
 {
  printf("%c", *ptr--);
 }
 getch();
}
```


Q Explain structure with suitable example (2015-16)
or
Difference between structure & array (2015-16, 2016-17)

Ans: Array	Structure
• Array is the collection of same data type element.	• It is the collection of variable of different data type.
• Elements are stored in contiguous memory location.	• Elements may not be stored in a contiguous memory location.
• Elements are accessed by their index numbers.	• Elements are accessed by their names.
• Array declaration and elements accessing operator is "[ ]" (square bracket)	• Elements are accessing operator is "." (Dot operator).
• Every element in array is of same size.	• Every element in a structure is of different data type.
• No keyword is used.	• struct keyword is used.
Syntax: data_type array_name[size];	Syntax: struct structname { data_type element 1; data_type element 2; ... } structure variable;

Q: WAP to check whether two dates are equal or not using structure

Ans:
```
#include <stdio.h>
#include <conio.h>
void main()
{
struct date
{
int d;
int m;
int y;
};
struct date d1,d2;
clrscr();
printf("\n Enter First Date\n");
scanf("%d %d %d",&d1.d,&d1.m,&d1.y);
printf("\n Enter Second Date\n");
scanf("%d %d %d",&d2.d,&d2.m,&d2.y);
if(d1.d==d2.d && d1.m==d2.m && d1.y==d2.y)
{
printf("\n Equal Dates\n");
}
else
{
printf("\n Unequal Dates\n");
```


Q: WAP to check whether Time 1 is equal to Time 2 or not using structure

Ans:
```
#include <stdio.h>
#include <conio.h>
void main()
{
struct Time
{
int h;
int m;
int s;
};
struct time t1,t2;
clrscr();
printf("\n Enter Time one\n");
scanf("%d:%d:%d",&t1.h,&t1.m,&t1.s);
printf("\n Enter Time Two\n")
scanf("%d:%d:%d",&t2.h,&t2.m,&t2.s);
if(t1.h==t2.h && t1.m==t2.m && t1.s==t2.s)
{
printf("\n Equal Times\n");
}
else
{
printf("\n Unequal Times\n");
}
getch();
}
```

Q Program to store record of 100 students & print record whose marks > 75

Ans:
```
#include <stdio.h>
#include <conio.h>
void main()
{
struct student
{
int rno;
char nm[10];
int mks;
};
struct student s[100];
int i;
clrscr();
printf("\n Enter detail of students \n");
for(i=0;i<100;i++)
{
scanf("%d %s %d", &s[i].rno, s[i].nm,
                  &s[i].mks);
}
printf("\n Student record mark > 75 \n");
for(i=0;i<100;i++)
{
if(s[i].mks>75)
{
printf("\n %d %s %d", s[i].rno, s[i].nm,
                  s[i].mks);
}
}
```


Q: Create a suitable structure in C language for keeping the records of the employees of an organization about their code, Name, designation, salary, department, city of posting. Also write a program in C to enter the records of 100 employee and display the name of those who earn more than 20,000 (2019-20)

Ans:
```
#include <stdio.h>
#include <conio.h>
struct emp
{ int eid;
  char en[20];
  char edes[10];
  float esal;
  char edep[10];
  char ecity[20];
} e[100];
void main()
{ printf("Enter details of 100 employees:");
  for(i=0;i<100;i++)
{ scanf("%d %s %s %f %s %s", &e[i].eid,
       e[i].en, e[i].edes, &e[i].esal, e[i].ed
       e[i].ecity);
}
```

```
printf("Name of employees whose salary is above 20k");
for(i=0; i<100; i++)
{ if(e[i].esal>20000)
  printf("\n%s", e[i].en);
}
getch();
}
```


Q Differentiate between structure and Union [2015-16, 18-19, 19-20, 20-21]

Ans: Structure	Union
• Keyword struct is use to define a structure.	• Keyword union is used to define a union.
• Every member within structure is assigned a unique memory location	• Memory location is shared by all data members.
• Changing the value of one data member will not affect other data member in structure.	• Changing the value of one data member will change the value of other data member of union.
• The total size depends on the sum of the size of every data member	• The total size depends on the size of the largest data member.
• We can access any member at a time.	• only one member can be access at a time

Syntax:

```
struct student {
    char name[20];
    int age;
    char address[50];
} std;
```

Syntax:

```
union student {
    char name[20];
    int age;
    char address[5
} std;
```

Q Write short note on • Union (2016-17, 2018-19)
• Enumerated data types

Ans: Union:

→ A union is a special data type available in C that enables us to store different data types in the same memory location.

→ Union can be defined with many members, but only one member can contain a value at a time.

→ Efficient way of using the same memory location for multiple purpose.

Syntax:-

```
union unionname
{ union_member1
  union_member2
} union_variable
```

Example:

```
union Data {
  int i;
  float f;
  char str[20];
} d;
```

Enumerated data type: It is user defined data type consists of integral constant and each integer constant is given a name.

• Keyword enum is use to define enumerated data type.

Syntax:-

```
enum type name { value1, value2, ... valuen };
```


example:

```
enum week { sun, mon, tue, wed, thu, fri, sat };
void main()
{ enum week today;
  today = wed;
  printf("%d day", today+1);
}
```

output: 4 days

Q: What is array of structure?

Ans: Declaring array variable of struct type like struct sample s[10]; using user defined datatype (struct sample)

Q: Find output

```
printf("%d", strcmp("QUIET", "QUILT"));
```

Ans:

output

= 7 (ASCII value of(E) - ASCII value of(L))

Q: Find output

```
printf("%d", strcmp("push", "pull"));
```

Ans:

output = 7 (ASCII(s) - ASCII(l))

= 115 - 108

= 7

UNIT-5

Q: What is pointer and how is it initialized? (2015-16,16-17, 2017-18)

Ans: Pointer is a special variable that stores the address of another variable. The pointer variable might belong to any of the data type (int, char, float).

Syntax: (Declaration of Pointer)

Datatype * Variable name ;

ex: int *p;
char *q;

where * denotes that p & q are pointer variable not a normal variable.

Initialization:

int a;
int *p = &a;

Q: What is double pointer? How will you declare double pointer in c?

Ans: When we define a pointer to pointer. The first pointer is used to store the address of the variable and the second pointer is used to store the address of the first pointer. That's why second pointer is called double pointer.

Declaration: • same as single pointer declaration
• Need to add one more * before the pointer

Syntax: datatype ** Variable name ;

example: int **p;

Q: What is void pointer ?

Ans: A pointer that does not have any datatype associated with it & can be store address of any type of variable.

like

```
void * ptr;
  int n;
  ptr = &n is valid
```

Q: state the features of pointer. WAP to sort given numbers using pointers? (2015-16, 2016-17, 2019-20)

Ans: Features of Pointer are :

- It reduce the length & complexity of program.
- It provide fast execution & better memory efficiency.
- It is use to perform dynamic memory allocation & deallocation.
- It helps to built complex data structure programs.

// WAP to sort numbers using pointer

```
#include <stdio.h>
#include <stdlib.h>
void main()
{ int *a, n, i, j, t;
  printf("Enter how many numbers");
  scanf("%d", &n);
```



```
  scanf("%d", &a[i]);
 }
 for(i=0; i<n; i++)
 { for(j=0; j< n-i-1; j++)
   { if(a[j] > a[j+1])
     { t = a[j];
       a[j] = a[j+1];
       a[j+1] = t;
     }
   }
 }
 printf("\n Sorted Array is : \n");
 for(i=0; i<n; i++)
 {
 printf("%d", a[i]);
 }
 getch();
```

WAP to sort elements using selection sort? (2018-19)

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a[25], n, i, j, t;
clrscr();
printf("\n Enter no of elements \n");
scanf("%d", &n)
printf("\n Enter each elements \n");
for(i=0; i<n; i++)
{
scanf("%d", &a[i]);
}
for(i=0; i<n; i++)
{
for(j=i+1; j<n; j++)
{
if(a[i] > a[j])
{
t = a[i]
a[i] = a[j];
a[j] = t;
}
}
printf("\n sorted Array is \n");
for(i=0; i<n; i++)
```


Syntax:

#define macrotemplate macro expansion

Example:

1) Simple Macro

```
#defined KEY 25
#defined PRINT printf("%d", i);
void main()
{ int i;
clrscr()
for(i=1; i<=KEY; i=i+1)
{ PRINT;
}
getch();
}
```

Note: The following code will replace macro template KEY & PRINT with its macro expansion 25 & printf("%d", i) respectively.

2) Argument Based Macro

```
/* program to calculate are of circle using macros*/
#include <stdio.h>
#include <conio.h>
#define AREA(r) 3.14*r*r
void main()
{ int r;
float ac;
clrscr();
```

```
printf("\n Enter Radius \n");
scanf("%d", &r);
ac = Area(r);
printf("area = %f", ac);
getch();
}
```

Note: In following code AREA(r) will be expand in form of 3.14 * r * r where r is the argument of macro.

File Inclusion:

It is used to include the content of any file in any other program using #include preprocessor or directive.

There are two ways to do it:

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

- This method will search the file stdio.h in standard / specific list of directories only.

```
#include "stdio.h"
```

- This method will search the file in standard as well as current directory.

example (stdio.h)

Conditional Compilation:

→ This method help programmes to write compatible code for two different machines.

→ The preprocessor directive used for this is

```
# if def
# else
# end if
```

example:

```
#define CODE
void main()
{ printf("a");
  # if def CODE
  printf("b");
  # else
  printf("c");
  # end if
  printf("d");
}
```

Output:

abd

Note: Since macro CODE is defined it will com
#if def block otherwise # else blo

...ious different modes in which these files can ...e used with proper syntax? (2015-16, 2017-18, 2019-20, 2020-21)

...: FILE *fp;
fp = fopen("filename", "mode");

...ere are 6 main types of file opening mode:

"r":- Open file for reading and file must exists

"w":-
- If file does not exists it created.
- If file already exist it's content is erased.

"a":-
- It adds all information at the end of the file leaving old data untouched.
- If file does not exist it is created.

"r+":- Open file for reading and writing and file must exist.

"w+":-
- Open file for reading and writing.
- If file does not exists it is created.
- If file already exist its content is erased.

"a+":-
- Open file for appending and reading.
- All data is written at the end of file leaving old data untouched.
- If file does not exists it is created.

Q Explain file handling? Explain few file handling library function with syntax?
or
Write the various input function used in file handling in C?

function	Description	Example
• fopen()	It is used to open the file.	fp = fopen("a.txt","r")
• fclose()	It is used to close the file.	fclose(fp)
• getc() & fgetc()	It read single character from file.	n = getc(fp) n = fgetc(fp)
• putc() & fputc()	It write single character to file.	putc(n, fp) fputc(n, fp)
• getw()	It read single integer from file.	n = getw(fp)
• putw()	It write single integer to file.	putw(n, fp)
• ftell()	It give current location of pointer.	ftell(fp)
• fseek()	It set pointer to desire location	fseek(fp, n, ref)
• rewind()	set pointer to begning of file	rewind(fp)
• fscanf()	It is used to read formatted data from a file.	fscanf(fp, "%d", &n)
• fprintf()	It is used to write formatted data into a file	fprintf(fp, "%d", n)

```
fclose(fe);
}
```

WAP in C to count the number of character in a file and copy the text of that file to another file. (2015-16, 2016-17, 2017-18, 2019-20, 2020-21)

Ans:
```
#include<process.h>
#include<stdio.h>
void main()
{ char ch;
  FILE *f1, *f2;
  int n=0;
  f1 = fopen("abc.txt","r");
  f2 = fopen("myz.txt","w");
  if((f1==NULL) || (f2==NULL)
  { printf("File can't open");
    exit(0);
  while((ch=fgetc(f))!=EOF)
  { n++;
    fputc(ch,f2);
  }
  printf("number of characters in file=%d",n);
  fclose(f1);
  fclose(f2);
}
```


Q: A file named data.txt contains a series of integer number. WAP to read number from file and then write all "odd" number to file odd.txt and all "even" number to file even.txt? (2019-20)

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{ FILE *fp, *fo, *fe;
  int n
  fp = fopen("data.txt","r");
  fo = fopen("odd.txt","w");
  fe = fopen("even.txt","w");
  if((fp==NULL) || (fo==NULL) || (fe==NULL)
  { printf("File can't open");
    exit(0);
  while((n=getw(fp))!=EOF)
  { if(n%2==0)
    putw(n,fe);
    else
    putw(n,fo);
  }
  fclose(fp);
  fclose(fo);
```

Q: #define PRODUCT(n) n*n
int main()
{
int i;
i = 64 / PRODUCT(4);
printf("%d", i);
return 0;
}
What will be the output? (2017-18)

Ans Preprocessor replace PRODUCT(4) with 4*4 so

i = 64 / PRODUCT(4).
i = 64 / 4*4;
i = 16 x 4
i = 64
so 64 will be print in output.

Q What is dynamic memory allocation and what is its role? How does it help in building complex programs? (2015-16, 2016-17, 2017-18, 2018-19, 2019-20, 2020-21)
Or
Differentiate (i) static & dynamic memory allocation
(ii) malloc() & calloc()

Ans: There are 2 types of memory allocation :-

Static Memory Allocation	Dynamic Memory Allocation
• static memory allocation is done at compile time.	• Dynamic memory allocation is done at run time.
• Memory allocation is fixed in nature.	• Memory allocation is flexible in nature.
• We can't reuse the unused memory	• The user can allocate more memory when required. and also the user can release the memory.
• Easy execution & faster	• Complex to implement & slower
• It is less efficient	• It is more efficient
example: Array	example: Linked List

DMA (Dynamic Memory Allocation) :-

→ The concept of DMA enables C programmer to allocate memory at run time

→ There are 4 library functions of stdlib.h or alloc.h header file.

1) malloc() 2) Calloc() 3) realloc() 4) free()

malloc() & calloc() → Both function are used to allocate memory at run time and return NULL if memory is not allocated.

malloc()	calloc()
• Allocate memory in single block	• Allocate memory in multiple block
• Initial value is garbage	• Initial value is zero
• Use single argument	• use two argument
• malloc is faster than calloc.	• calloc takes little longes than malloc
• syntax: int *a, n; a = (int *) malloc (n*2)	syntax: int* a, n; a = (int*) calloc (n, 2)

realloc() → We can reallocate the memory by realloc() function.

Syntax: int *a;
a = realloc (a, newsize)

free() → The memory occupied by malloc() or calloc() function must be release by free() function.

syntax: free (a)

Q: What is linked list ? Write the self reference structure of a node in linked list ? Write the adiantage of linked list. (2018-19)

Ans: • Linked list is also know as dynamic data structure where each structure of the list is called a node.

• The node consist of two fields, one store the value and other contain address of next node.

• The node is represented by
struct node
{ int item;
struct node * next;
};

• The above structure which contain a member field that point to same structure type is called self refuential structure.

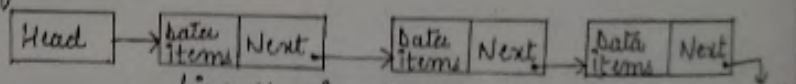


fig: Showing linked list

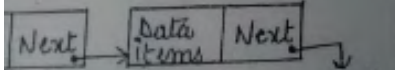
Adiantage	Disadiantage
• It can grow or shrink during execution, Hence avoid memory wastage	• It consume more time to access the arbitrary item
• It provide flexiblety.	• It consume more space compare to array.

he self reference structure
ite the advantage of

as dynamic data
ucture of the list

s, one store the value
next node.

ntain a member field
type is called self



list

isadvantage

nsume more time to
the arbitrary item.

nsume more space
re to array.

Explain command line argument in c with example?
Ans: It is a parameter supplied to a program when the program is invoked.
This parameter may represent a file name & input strings

Syntax:
```
main (int argc, char * argv [])
{
}
```

Here argc → argument count
argv → array of characters pointers

Example:
```
/* Program to find factorial using CLA */
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void main (int argc, char * argv[])
{ int n, f=1, i;
  clrscr();
  n = atoi (argv[1]);
  for (i=n; i>1; i--)
  { f = f * i;
  }
  printf ("Factorial = %d", f);
  getch();
}
```