Syllabus of 'C'

Page No
Date 1 feb

⇒ Overview of C
- ↳ History of C
- ↳ Advantage
- ↳ Disadvantage
- ↳ Application

(*) Tokens
- ↳ ~~structure~~ character set
- ↳ Constant
- ↳ variable
- ↳ Operater
- ↳ Data type
- ↳ Keywords
- ↳ Escape sequence
- ↳ Type specifier
- ↳ Comment
- ↳ Expression

(*) Structure of program
- ↳ syntax of program
- ↳ installing C compiler
- ↳ Compilation & execution

(*) I/O Operation
- ↳ printf()
- ↳ scanf()

(*) Control statement
- ↳ Sequential statement
- ↳ looping statement
- ↳ jumping statement

(*) ARRAY
- ↳ overview of Array
- ↳ Advantage & disadvantage
- ↳ Type of Array

Date

* String handling
  - one dimensional
  - Two dimensional
  - String library function

* function
  - Overview of function
  - Types of function
  - Architecture of function
  - Recursive function

* Pointer
  - over view of pointer
  - Operater used in pointer
  - pointer arithmetic
  - Pointer with array
  - void pointer / Generic pointer
  - Double pointer
  - Dynamic memory allocation
    - malloc()
    - calloc()
    - realloc()
    - free()
  - Pointer with string
  - Pointer with function

* Structure
  - overview of Structure
  - Structure with array
  - structure with ~~array~~ function
  - Structure with pointer
  - Nested Structure

Page No.
Date

* union
* Command line argument
* Preprocessor directive
* User- defind header file
* Library function
* file handling
  - ↳ overview
  - ↳ Sequential file
  - ↳ Data file
  - ↳ Binary file

* overview of graphics Programming

Overview:-

History of C:-

(i) C is a high level and low level Programming language so it is called mid or middle level language.

(ii) It was devloped by Dennis Ritchie at AT & T bell laboratory of USA in 1972.

(ii) AT & T stands for American Telegraph and Telecommunication.

(iv) It was derived from BCPL (Basic combind Programming Language) and B language

(v) It is set to be Mother language of all Programming language.

(vi) The main objective of C language is to devloped system s/w as operating system, Compiler, Linker, loader, Network driver etc.

(vii) Unix operating system was written within C programming

(viii) It is case sensetive programming language. it means lower case is not equavalent to upper case.

i.e abc ≠ ABC

(*) Advantage of 'C'

(i) It supports procedural programming / Modular programming / structured Programming Conventional programming.

It is a Programming technique in which a large program can be devided into Multiple parts called function or Module so we can say that this technique reduce the complexity of the program.

This concept is implemented with the help of function.

(ii) It is portable language because it can be executed on different operating system such as windows, linux, mac (macintash) etc

(iii) It is richset of library function such as mathematical function, String function, character function, Date and time function.

(iv) It is compiler based language so its execution is fast.

(v) It supports graphics programming. In this case we can design different shape as human face, car, home etc.

(vi) It also supports game programming due to graphics

(vii) It supports system programming. In this case we can design system s/w such as o/s, compiler, interpreter etc

(viii) It also supports application programming. In this case we can design application s/w such as school management system, Hospital management system etc

(ix)

Disadvantage :-

(i) It is plateform dependent / machine dependent.
It is major drawback of C.

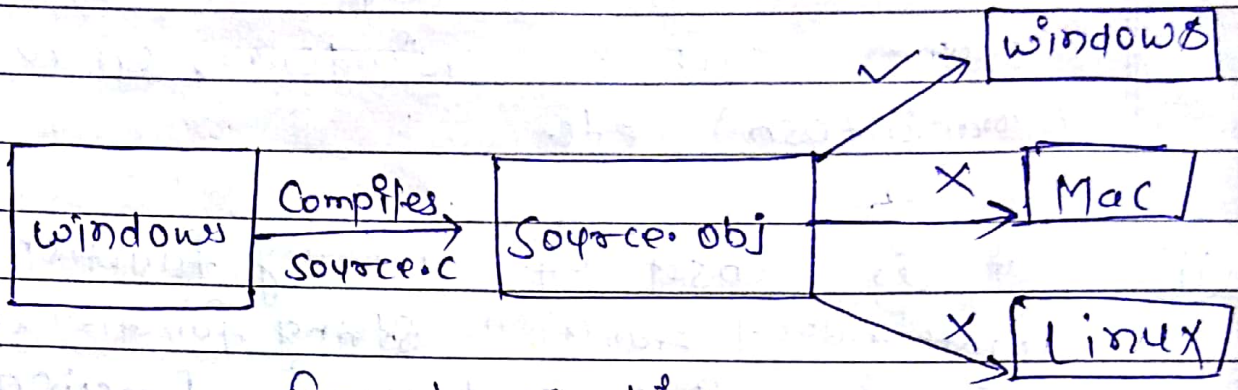


figure:- Machine dependent Plateform

(ii) It does not support object oriented Programming

(iii) It is not secure language

(iv) It does not support database directly.

(*) Application of C :-

(i) use to design system software.
(ii) use to design application software.
(iii) use to design game programming.
(iv) It is used in DBMS
(v) It is used in robotics.
(vi)

Page No. 1
Date 28/1/22

Tokens:-

Tokes are the smallest unit that are used to devlop a program.

There are some tokens in C.

(i) character set
(ii) Constant
(iii) variable
(iv) Operater
(v) Data type
(vi) Keywords
(vii) Identifire
(viii) Escape sequencie
(ix) Type specifier / formet specifier
(x) Comment
(xi) Expression

(i) character set

Three types

- Alphabet → Ascii
  - A-Z (65-90)
  - a-z (97-122)
- Digit
  - 0 to 9 (48-57)
- special char
  - $, ^, ~, ', ., ", -, - etc.

ASCII - American Standard code information Interchang

(ii) Constant:- Constant is a fixed value that can't be changed during execution of a program. It is also known as literal. There are three types of Constant.

Page No.
Date

Constant :-

(i) Numeric Constant
(ii) Character constant
(iii) String Constant

- Integer Constant: 10, 20, 500 etc.
- Real Constant / Float Constant: 10.5, 18.456 etc.

(ii) Character Constant

→ It represents only one char that is enclosed within single quote.
e.g.:- '$', '1', 'A' etc.

(iii) String Constant

→ String is a collection of zero or one or more chars that are enclosed within double quote. It must be terminated with the help of null char (\0).
e.g.:-

(a) "COC" → | C | O | C | \0 |
(b) "$" → | $ | \0 |
(c) "" → | \0 |

(iv) Variable:- It is a storage Memory location which is used to hold Constant value according to the specified data type. The value of Variable can be changed during execution of a program

e.g.

Variable:
- a = 10
- b = 10.5
- name = "Sanjeev"
- ch = 'B'

Constant

Page No
Date

Types of variable

(i) local variable

(ii) Global variable

(i) Local variable :- Those variable that are declared within function called local variable. It provides local accessibility. It means it can accessed within function so we can say that it can't be accessed outside function.

(ii) Global variable :→ Those variables that are declared outside function called global variable. It provides global accessibility. It means it can be accessed within function and outside function also.

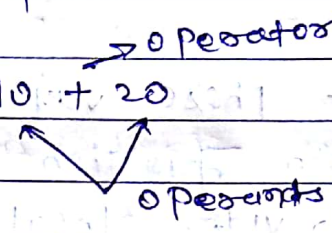
e.g :-

```
int a=10; // global variable
void main()
{
  int b=20; // local variable
  Printf("value of a =%d", a);
}
void abc()
{
  Printf("value of a =%d", a);
  Printf("value of b =%d", b); // error because local variable
```

Page No.
Date

(Y) **Operater :-**

→ Operater is a special symbol that performs operation on operands.



(a) Asithmetic operater
(b) Relational operater / Comparision operater
(c) logical operator
(d) Assignment operator
(e) Multiple operator
(f) Increment operator
(g) Compound assignment operator
(h) Decrement operator
(i) comma operator
(j) sizeof operator
(k) Bitwise operator
(l) unary operater
(m) Binary operator
(n) Ternary operator / conditional

(a) Asithmetic operator :→ This operator performs arithmetic operation as addition, substraction multiplication etc.

Operator		meaning
+	→	Addition
-	→	Substraction
*	→	multiplication
/	→	Division
%	→	Remainder / modulo division

10 * 5 : 50
10/4 = 2
10.0/4 = 2.5
10/4.0 = 2.5
10%.4 = 2
4%.10 = 4
-10%.4 = -2
-10%.-4 = -2
10%.-4 = 2
10.0%.4.0 = ? //error

(i) Note

(ii)

NOTE:

(i) If the value of first operand is less than second operand the reminder is the value of first operand.

(ii) The sign of reminder depends on the sign of first operand.

(iii) Real type operands can't be used with Modulo division.

(b) Relational operator :- This operator is used to compare between two values. It return Boolean value as zero or one.

Operator	Meaning
<	less than
>	Greater than
<=	less than or equal to
>=	Greater than or equal to
!=	Not equal to
==	Equal to

e.g:-

20>=20 =? True → 1

80!=30 =? false → 0

40<30 =? false → 0

© Logical operator:- This operator is used to combine more than one condition. It also return boolean value.

Operator	Meaning
&&	logical AND
|| or ||	logical OR
!	logical NOT

logical AND (&&) :- It means true if all combined conditions are true otherwise returns false.

e.g i> 80>=80 && 4!=6 && 5>3 => True(1)

ii> 40>=50 && 4==5 && 30>=30 → false(0)

× ↓

It's logical AND short circuit.

logical OR (||) :- It returns true if any combined condition is true otherwise returns false.

e.g:- (i) 80>=50 || 40!=40 || 30>40 → false (0)

× × ×

(ii) 40>=40 || 8!=5 || 4==6 => True(1)

✓ ↳ logical OR short circuit

Logical NOT (!) :→ It returns true if condition is false. It returns false if condition is true.

e.g:- i> !(20>=20) → false

ii> !(30>=20 && 4!=6) → false

iii> !(30>=50 || 5==5 || 6>=7)

→ false

(d) Assignment operator :- (=) → This operator is used to assign the value of in variable.

e.g a=10

b=10.6

(e) Multiple assignment :- This operator is used to assign the same value in multiple variables.

e.g:- a=20
b=20
c=20
d=20

or a=b=c=d=20

(f) Compound assignment operator:- Some operator that are used with operator called compound Assignment operator. It is also known as shortcut assignment operator

(i) a=a+10

↳ a+=10

(ii) a=a*20

↳ ~~a*20~~ a(*=)20

(iii) a=a/5

↳ a(/=)5

→ Compound assignment operator.

Date

a=10, b=30 — 1° +30 = 40

a+=30 ; ⇒ a= a+30

b*=5 ; ⇒ b=b*5

= 30*5 = 150

(g) Increment operater :→ This operater is used to increase the value of variable by 1.

sign of increment operater → (++)

There are two types of increment operater.

(i) Prefix increment :- In this case operater is used before operand

e.g :- ++a, ++c, ++d

(ii) postfix increment :- In this case operater is used after operand.

e.g → a++, b++, c++

⊛ Evaluation of prefix and postfix

Prefix

(i) a=10

++a ⇒ a=11

a=11

(ii) a=10

p= ++a ⇒ a=11, p=11

a=11, p=11

(iii) a=10, b=20

++a ⇒ a=11

++b ⇒ b=21

p= ++a ⇒ a=12, p=12

q= ++a = b=22, q=22

a=12, b=22, p=12, q=22

Postfix

(i) a=10

a++ = 10

a = 11

(ii) a=10

p=a++

=10,

p=10

a=?=11

p=10

(iii) a=10, b=20

a++ ⇒ a=10

b++ ⇒ b=20

p= a++ ⇒ a=11, p=11

q=b++ ⇒ b=21, q=21

q=12, b=22

p=11, q=21

Page No.
Date

a = 20, b = 30

a++ = 20

b++ = 30

++b = 31

a++ = 21 , ++b = 33

p = a++ = 21

q = b++ = 33

a = 23
b = 34
p = 22
q = 33

a = 40, b = 50

++a = 41

++a = 42

a++ = 42

b++ = 50

++b = 52

b++ = 52

p = ++b = 54

q = ++a = 44

a++ = 44

a = 45	Confirmed
b = 54	
p = 54	
q = 44	

Page No.
Date

(*) Decrement operator (--) :-

This operator is used to decrease the value of variable by one.

prefix decrement e.g :- --a, --b

postfix decrement eg :- a--, b--

e.g a=40 , b=50

--a = 39

b++ = 50

--b = 50

a-- = 39

p = ++a = 39

q = b-- = 50

++b = 50

b-- = 50

++a = 40

--a = 39

a = 39
b = 49
p = 39
q = 50

Confirmed

(I) Comma operator :→ This operator is used to separate multiple values within small bracket. In this case last value is evaluated.

e.g :→ (i) a = (10,20,30,40)

a = ? = 40

(ii) a = 10, b = 20

c = (++a, b--, ++b, a--, b--, ++a);

c = (11, 20, 20, 11, 20, 11);

a = 11, b = 19, c = 11 Ans

Page No.
Date

(J) sizeof operator :→ This operator returns the size of datatype and variable.

Syntax:-

sizeof (variable / datatype / constant)

e.g:- sizeof (int) ⇒ 2
sizeof (char) ⇒ 1
sizeof ('A') = 1
sizeof ("coc") = 4

c	o	c	\0

sizeof ('\n') = 1
sizeof (10) = 2
float pi ;
sizeof (pi) ⇒ 4
long int a; ⇒ 4
sizeof (a) ⇒

(K) Bitwise operator :→ This operator Performs operation in bit format as zero (0) or one (1). It is generally used in low level programming.

There are two types of bitwise operator.

(i) shift bitwise operator
(ii) logical bitwise operator

(i) shift bitwise operator :→ This operator is used to shift bit to left and right side. So it can be divided into two parts

(a) left shift bitwise operator (<<)
(b) Right shift bitwise operator (>>)

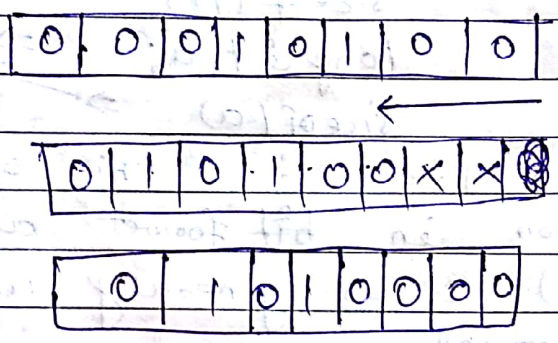
Page No.
Date

(a) left shift bitwise (<<) → it is used to shift bit to left side and filling with zero from right side.

e.g 20<<1
20<<2
20<<3
20<<4

Q 20<<2 = ?

sol:→ 20 ⇒ 10100
8 bit ⇒ 00010100



10100 00
64 32 16 8 4 2 1
64+16 = 80

short cut rule:→

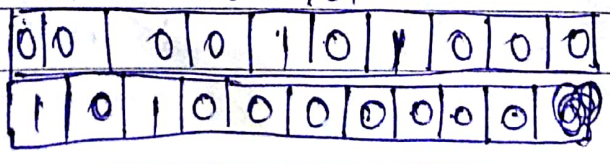
* 20<<2
$20 * 2^2 = 80$

* 40<<4
$40 * 2^4 = 640$

32 16 8 4 2 1
1 0 1 0 0 0

Q 40 <<4

40 = 101000
10 bit = 0000101000



1 0 1 0 0 0 0 0 0 0
512 256 128 64 32 16 8 4 2 1

512+128
= 640

(b) Right shift bitwise operator :→ This operator is used to shift bit to right side and filling with zero from left side

e.g:→ 20 >> 2

20 binary = 10100
8 bit = 00010100

0	0	0	1	0	1	0	0

0	0	0	0	0	1	0	1

1 0 1
4 2 1

= 5 Ans.

short cut rule
20>>2
20/2^2
= 20/4 = 5

① 50 << 3 = ?
② 90 << 2 = ?
③ 60 >> 3 = ?
④ a = 50, b = 60
c = (a--, b--, ++b, a--, --a)
a = ?
b = ?
c = ?

⑤ a = 10, b = 20

a--	q = b--	a = ?
b--	p--	b = ?
b--	q++	p = ?
--b		q = ?
p = q--		

Page No.
Date

(i) 50 << 3 = ?

Sol:→ 50 = 110010

8 bit = 00110010

0	0	0	1	1	0	0	1	0

←

1	1	0	0	1	0	0	0	0

1 1 0 0 1 0 0 0 0

256 128 64 32 16 8 4 2 1

256+128+16 = 400 Ans

50

32 16 8 4 2 1

1 1 0 0 1 0

(ii) 90 << 2 = ?

90 = 1011010

64 32 16 8 4 2 1

1 0 1 1 0 1 0

9 bit = 001011010

0	0	1	0	1	1	0	1	0

←

1	0	1	1	0	1	0	0	0

1 0 1 1 0 1 0 0 0

256 128 64 32 16 8 4 2 1

256+64+32+8 = 360 Ans

(iii) 60 >> 3 = ?

60 = 0111100

9 bit = 000111100

64 32 16 8 4 2 1

1 1 1 1 0 0

0	0	0	1	1	1	1	0	0

0	0	0	0	0	0	1	1	1

1 1 1

4+2+1

= 7

60/2

60/8

= 7

(4) a = 50, b = 60

c = (a--, b--, ++b, a--, --a)

a = ?

b = ?

c = ?

c = (50, 60, 61, 49, 47)

a = 47

b = 61

c = 47

(5.)

a = 10, b = 20

a--	= 10
b--	= 20
b--	= 19
--b	= 17
p = a--	= 9.
q = b--	= 17
p--	= 9
q++	= 17

a = ? = 8

b = ? = 16

p = ? = 9

q = ? = 17 Ans

Page No.
Date

(ii) Logical bitwise operator :→ This operator performs logical operation in 0 and 1 format. These are four types of logical bitwise operator.

(a) logical And bitwise (&)
(b) logical OR bitwise (|)
(c) logical Ex-OR bitwise (^) → caret
(d) logical NOT bitwise (~) → tilde

(a) Logical AND bitwise. :— it returns 1 if both bits are 1 otherwise returns 0 (zero).

Truth Table

a	b	a&b
0	0	0
0	1	0
1	0	0
1	1	1

e.g 40 & 50 = ?

40 → 101000
50 → 110010
100000 → 32

e.g 70 & 80 = ?

64 32 16 8 4 2 1
1 0 0 0 1 1 0
1 0 1 0 0 0 0

70 → 1000110
80 → 1010000
1000000 → 64

(b) logical OR bitwise operator :→ it returns 0 if both bits are (0) otherwise return 1.

Truth table :→

a	b	a|b
0	0	0
0	1	1
1	0	1
1	1	1

e.g 40 | 50 = ?

40 → 101000
50 → 110010
111010
32 16 8 4 2 1
58 Ans

(C) Logical EX-OR bitwise (^) → it returns 0 if both bits are same otherwise returns 1.

Truth table

a	b	a^b
0	0	0
0	1	1
1	0	1
1	1	0

e.g:→ 40^50 = ?

40 → 101000
50 → 110010
011010

16+8+2 = 26 Ans

(D) Logical NOT bitwise operator:→ (~) it returns 0 if bit is 1 and it returns 1 if bit is zero.

Truth Table

a	~a
0	1
1	0

e.g ~80 = -81
~60 = -61
~59 = -60

80 → 1010000
8bits → 01010000
~80 → 10101111
2^{1s} → 01010001
128 64 32 16 8 4 2 1
64+16+1 = -81

(L) UNARY Operator:→ this type of operator is used with only single operand.

e.g:→ increment operator
→ a++, ++b

Decrement operator
→ a--

Logical Not bitwise operator
→ ~40

MINUS and plus (unary + binary both)
+20, -10

Date

(M.) Binary operator :→ This type of operator must be used with two operands
e.g→ 10-5 , 10+5, 4==3 , 20<=80, 40&50

(N.) Ternary operator :→ sign→ (?:) (Conditional operator)

→ This operator is mainly used to check Condition. In this case if condition is true then true part will execute otherwise false part will execute. it is said to be ternary operator because it is used with three operands as condition, true part and false part.

syntax:- (Condition ? true part : false part)
true ↑ false ↑

e.g:→ (x%2==0 ? "even" : "odd");
(n>0 ? "Positive" : "Negative");

operator final

Page No.
Date 10 02 2018

(v) Data type :- it is used to specify type of data that is hold by variable. It is mainly used to declare variable in a Program. There are mainly two types of data type

(i) Pre-defind data type / basic data type / fundamental Data type / In-Built data type

(ii) User defind data type / derived data type / Composite Data type / abstract data type

(i) Pre-defind data type :→ Those data types data are already defind in 'C' language Called Pre-defind data type.
There are four basic data type.

(a) int :- It represents only integer value. Its size is 2 byte in turbo Compiler

(b) float :- It represents real type value. It size is four byte. It supports single precision (max 6 digit after decimal point)

(c) Double :→ It also represents Real type value. it size is 8 byte. it supports double precision (max 15 digit after decimal point).

(d) Char :- It Represents only character type value. Its size is one byte.

NOTE :- The size of data type depends a/c to Compiler.

e.g :→ int — 2 byte (Turbo compiler)
int — 4 byte (borland " / gcc Compiler)

Page No.
Date

(*) Data type modifier :→ This modifier is used with Data type and changes the meaning of data type. There are four data type modifier.

(i) Signed :- It represents both positive and negative value but it is defaut data type modifier

e.g:- int a;
a=10, a=-10;
signed int a;
a=10, a=-10;

(ii) unsigned :- It Represents only positive value.

e.g:- unsigned int a;
a = -100; X
a = 100;

(iii) Short :→ It decreases the size of data type.

e.g
int - 2 byte
short int - 1 byte

(iv) long :- It increase the size of data type.

e.g int → 2 byte
long int → 4 byte

List of data type

Data type	size (byte)	
int or signed int	2	integer
unsigned int	2	
short int or signed short int	1	
unsigned short int	1	
long int or signed long int	4	
unsigned long int	4	
float or signed float	4	Real type
unsigned float	4	
double or signed double	8	
unsigned double	8	
long double or signed long double	10	(Largest datatype in C.C++)
unsigned long double	10	
Char or signed char	1	character
unsigned char	1	

Q How to calculate the range of data type?

→ for signed data type

$$-2^{bit-1} \text{ to } 2^{bit-1} - 1$$

e.g:- int → 2 byte (16 bits)

$$\Rightarrow -2^{16-1} \text{ to } 2^{16-1} - 1$$

$$\Rightarrow -2^{15} \text{ to } 2^{15} - 1$$

$$= -32768 \text{ to } 32767$$

e.g:- int a;

a = 30000 , a = -31000;

a = 40000 X, a = -40000 X

for unsigned data type

$$0 \text{ to } 2^{bit} - 1$$

e.g:- unsigned int

$$0 \text{ to } 2^{16} - 1$$
$$= 0 \text{ to } 65536 - 1$$
$$= 0 \text{ to } 65535$$

i.e unsigned int a;

a = 40000;

a = 60000;

a = -10 ✗

(ii) user defind data type :→ These data types are created with the help of user. In other words, these data types that are derived from basic data type called user defind data type.

e.g:- Array, Structure, union, enumeration etc.

(iii)

(vi) Keywords :- / Reserved wood

Keywords are special reserved wood that consist of special meaning in programming. Keywords cant be identifire as variable name, function name, structure name etc.

There are 32 keywords in C Language.

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

(vii) (*) Escape sequence :- Those characters that cant be typed / Printed directly in a program called Non-printable character / escape character. So these characters are represented / Printed in a program with the help of escape sequence. All escape sequence must be started with backslash (\).

e.g.:-

Date

Escape sequence	Meaning
\n	New line
\t	Horizental tab
\v	Vertical tab
\b	Backspace
\r	Carriage return
\\	Backslash
\"	double quote
\'	single quote
\?	Question mark
\a	Alert bell/ Beep sound
\0	Null

e.g:→

(i) Printf("welcome \n to \n c");

o/p→ welcome
to
C

(ii) Printf("7\a 2\a 7\a 7\a ----");

e.g "coc"

c	o	c	\0

(iii) printf("welcome \b\b\b\b");

o/p→ wel
welcome

(iv) Printf("welcome to\r c");

op→ C

(viii) Type specifier / formet specifier :→ It is mainly used to Print the stored value in a variable and get different types of value from console. It is used in formeted function as printf, scanf, fprintf etc. All Type specifier must be started with %. sign.

Data type	Type specifier
int	%.d or %.i
unsigned int	%.u
long int	%.ld
short int	%.h
float	%.f, %.e, %.g
double	%.lf
long double	%.Lf
char	%.c
string	%.s
octal NO	%.o
Hexa Decimal	%.x
Memory address	%.u or %.p

(ix) Identifire :— Identifire is used to Identify variable name, function name, structure, Name, union name etc. There are some Rules for Creating Identifire.

for all programming language

(a) it is collection of alphabet, digit and only one special character as underscore.
(b) it is can't be started with digit
(c) Keywords can't be Identifire.
(d) space is not allowed in spec Identifire.

e.g:→ first Name ✗
s.i ✗
abc123 ✓
-first ✓
for ✗
if ✗
123xyz ✗
last-Name ✓
abc$@123 —✗
int —✗

(x) Comment :→ Comment is the most important token in programming language which is used for documentation purpose. Comment section can not be read by compiler. It creats user friendly program so new user can understand source code easily.

There are two types are Comment :-

(i) single line comment (//)
(ii) multi line Comment (/* ---- */)

```
void main()
{
clrscr(); // clear previous output on console
printf("welcome");
// printf("\n to");
/* printf("\n thanks");*/
getch(); // holds output screen
{
```

single Comment ← // printf("\n to");

multiline Comment ← /* printf("\n thanks");*/

Page No.
Date

(*) expression :→ It is collection of operand and operators but it is not equated. If it is equated than it is called equation.

e.g (i) a+b-c*d → expression

(ii) a+b = c-d → equation

(*) Structure of C program :→

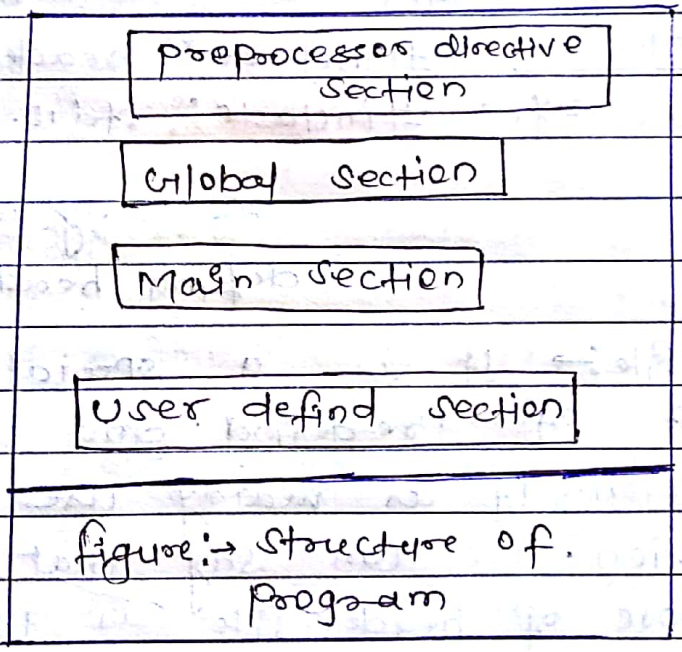


figure :→ Structure of program

Program :-

```
#include <stdio.h>
#include <conio.h>
```
→ Preprocessor directive section

```
void main()
{
clrscr();
printf("welcome to C");
getch();
}
```
→ Main function

13 feb 2018

Page No.
Date

explain:—

#include :→ It is a Preprocessor directive which is used to include header file in a program. It must be written as first statement in source code. There is no space between # and include.

1st syntax:— #include <header file name>

eg #include <stdio.h>

#include <math.h>

2nd syntax:— #include "header file"

eg:→ #include "stdio.h" ↓

Generally used in user-defind header file

* Header file:→ It is a special file that Consist of pre-defind and user-defind function. It is mainly used to organise function. we can say that the main purpose of header file is to access function in a program. The extension of header file is .h.

There are some headerfile in C.

stdio.h → standard INput output header file
conio.h → console " " "
math.h → mathematically function
ctype.h → Character section
process.h →
stdlib.h → standard library "
string.h →
dos.h → disk operating system "

Page No.
Date

time.h ⇒ Allocation " "
malloc.h ⇒ memory allocation "

(*) **Main function :→** It is an entry point in program which we write source code a/c to requirement. The execution of program is started with main function. There are two points regarding main function.

(i) It returns by default integer type value.
(ii) It is an user defind function because its body is defind with the help of user a/c to requirement.

In C language

Syntax:-

```
main()
{
    block of statements;
}
```

(2) In c++ language

(1)
```
Void main() // Non-return type
{
  block of statements;
}
```

(2)
```
int main() // return type
{
  block of statements;
    return (0);
}
```

Page No.
Date

* clrscr() → This function is used to clear the previous output on the Console. This function available in "<conio.h>" header file.

Syntax:→ clrscr();

* getch() → It is used to input one character at a time. So this function ~~show~~ holds output screen after executing of a Program. This function is available in <conio.h> header file.

Syntax:- getch();

* How to declare variable :→

→ Syntax:-

<data type> <variable name1> [= value], <variable name2> [= value]; -----;

e.g

```
int a=10, b, c, d=20;
float p, q;
char ch;
```

⇒ Initialization

1st method

During declaration time

e.g int a=10, b=20;

2nd method

After declaration

e.g

```
int a, b;
a = 10;
b = 20;
char ch;
ch = 'A';
```

Page No.
Date

(*) How to save 'C' program :—

C Program must be saved with the help of .c extension

1st Method → f2

2nd method → using menu

* How to compile 'C' program

→ 1st method → Alt + f9

2nd method → using menu → Compile

* How to execute 'C' program :—

1st → Ctrl + f9

2nd → using menu → run

(*) How to display output

Alt + f5

(*) INput / output function :—

printf() :→ It is an output function which is used to print message and Result on the Console. This function is available in stdio.h header file.

Syntax:→

int printf("<message><escape square> <Type specifier>", <list of variables);

i> printf("welcome to C");

o/p → welcome to C

ii> printf("welcome \n to \n C");

o/p → welcome
to
C

(iii) printf("\" welcome to C\"");

o/p → "welcome to C"

Date

```
int a=10, b=20;
```

o/p Value of a = 10
Value of b = 20

```
printf("Value of a = %d \n Value of b = %d",
                                    a, b);
```

or,

```
printf("Value of a = %d", a);
printf("\n Value of b = %d", b);
```

Q Write a program in c to find area and Perimeter of rectangle for given length and breadth.

→

```
#include <stdio.h>
#include <conio.h>
void main()
{
clrscr();
int l, b, area, Peri;
l=20; // Compile time input
b=10;
area = l*b
Peri = 2*(l+b)
Printf("Area of rectangle = %d \n
Perimeter of rectangle = %d", area,
peri);
getch();
}
```

Write a program in C to find addition and product for given three Number.

```
#include <stdio.h>
#include <conio.h>
  void main ()
{
 clrscr();
int  a, b, c, add, product;
  a=10; b=20; c=30;
  a=10;
  b=20;
  c=30;
add = a+b+c;
Product = a*b*c;
Printf("Addition of three Number = %d \n
    product of three Number = %d", add,
    product);
  getch();
}
```

Q How to read / input value during Runtime ?

Scanf :- It is an input function which is used to input value during Runtime. This function is available in "<stdio.h>" header file. INput operation must be performed on memory address.

Syntax :-

```
int Scanf (" Type specifier ", &<var1>,
                &<var2>, ____);
```

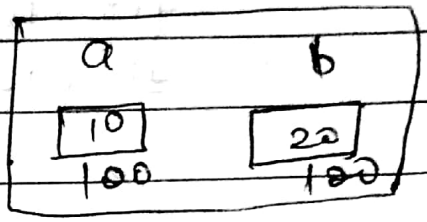
& → address of operator

It returns the memory address of any Variable.

Page No.
Date

a=10, b=20

e.g a, b;
printf("Enter two Number");
scanf("%d%d", &a, &b);



(*) Write a program in c to input two Number after that perform arithmetic operation.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
 int  a,b;
 printf("Enter two Number :—");
 scanf("%d%d", &a, &b);
printf("Addition = %d\n substraction = %d\n
        multiplication = %d\n", (a+b),(a-b),
        (a*b));
 getch();
}
```

Write a program in to input redius after that find area and perimeter of circle.

```
#include<stdio.h>
#include<conio.h>
  void main
{
  clrscr();
float r, area, peri;
Printf("Enter the redius of circle");
scanf("%f", &r);
  area = 3.14 * r * r;
  peri = 2 * 3.14 * r;
Printf("Area of circle = %f \n Perimeter of
    circle = %f", area, peri);
  getch();
}
```

Q Write a program in c to input marks of four subject after that find total marks and percentage marks.

```
#include<stdio.h>
#include<conio.h>
  void main()
{
  clrscr();
int m1, m2, m3, m4, totalm, per;
Printf("Enter marks of four subjects");
scanf("%d %d %d %d", &m1, &m2, &m3, &m4);
totalm = m1+m2+m3+m4;
per = m1+m2+m3+m4/4;
Printf("Total marks of four subjet = %d \n
  Percentage marks = %d", totalm,
  per);
getch();
}
```

Write a program in c to input purchase amt of product after that find net amt if discount is 150%.

```
#include <stdio.h>
#include <conio.h>
    void main ()
{
  clrscr();
  int puramt, netamt, diss ;
  printf("Enter Purphase amount of product");
  Scanf("%d", &puramt);
  Dis = (puramt * 1.5)/100;
  Netamt = puramt - Dis ;
  printf(" Net amount of product = %d \n
        purchase amount of product = %d \n
        Discount of product = %d",
        Netamt, puramt, Dis);
  getch();
}
```

Write a programming in c to input basic salary of employee after that find gross salary for given extra allowance as :-

HRA = 40%.

DA = 90%.

TA = 30%.

MA = 10%.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
float bs, gs, hra, da, ta, ma;
Printf("Enter basic salary");
scanf("%f", &bs);
  hra = (bs * 40)/100;
  da = (bs * 90)/100;
  ta = (bs * 30)/100;
  ma = (bs * 10)/100;
  gs = bs + hra + da + ta + ma;
printf("salary = %f", gs);
  getch();
}
```

* Write a program in c to input any number after that check number is even or odd using conditional operator.

→
```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
  int x;
Printf("Enter any number");
scanf("%d", &x);
(x%2==0 ? Printf("Even no"): Printf("odd no"))
  getch();
}
```

How to check:-

```
#include<stdio.h>
#include<conio.h>
  void main()
{
  clrscr();
  int a=10, b=20, c;
  c=(a++, ++b, b--, a--);
  printf("value of a=%d\n value of b=%d\n value of
        c=%d", a,b,c);
  Printf("30<<3 = %d\n", 30<<3);
  printf("size of(int) = %d\n", sizeof(int));
  printf("size of(\"sanjeev\") = %d\n", sizeof("sanjeev"));
  Printf("size of(long double) = %d\n", size of(long double));
  printf("~89 = %d\n", ~89);
    printf("70&40 = %d\n", 70&40);
    printf("30%-40 = %d", 30%-40);
    printf("50%-12 = %d", 50%-12);
    getch();
}
```

(*) Write a program in c to input any Number after the check number positive or Negative

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
    int n;
  Printf("Enter any Number");
  scanf("%d", &n);
  (n>0 ? Printf("positive") : Printf("Negative"));
    getch();
}
```

a, b a==b

Page No
Date

* Write a program in c to input age of student after check who is eligible for voting or not?

```
#include<stdio.h>
#include<conio.h>
  void main()
{
  clrscr();
    int a;
Printf("Enter age of student");
scanf("%d",&a);
( a>=18 ? Printf("eligible") : Printf("not
                                eligible"));
  getch();
}
```

(*) Write a program in c to input two Number after that check both are equal or not

```
#include<stdio.h>
#include<conio.h>
  void main()
{
  clrscr();
    int a,b;
Printf("Enter two Number");
scanf("%d%d",&a,&b);
( a==b ? printf("equal") : Printf("not equal"));
```

Page No.
Date

(*) Write a program in C to input two Numbers after that find greatest number using ternary operator

```
#include <stdio.h>
#include<conio.h>
    void main
{
clrscr();
        int a,b;
printf(" Enter two Number");
scanf(" %d%d ", &a, &b);
(a>b ? printf(" Greatest no=%d", a) :
 printf(" Greatest no=%d", b));
 getch();
}
```

or

```
int gt;
Printff
gt=(a>b? a:b);
printff(" Greatest no=%d ", gt);
```

(*) write a program in C to input any alphabet character after that check character is lower or upper

```
#include<stdio.h>
#include <conio.h>
void main()
{
clrscr();
int char ch;
printf(" Enter any alphabet char");
scanf(" %c ", &ch);
ch>='A' && ch<='z' ? printf(" upper") :
printf(" Lower"));
getch();
}
```

Nested Ternary

Page No
Date

(*) W A Program in C to input three Number after that find greatest Number.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int a,b,c,gt;
  Printf("Enter three Number");
  Scanf("%d%d%d",&a,&b,&c);
 gt=(a>b&&a>c?a:(b>c?b:c));
  printf("Greatest no=%d",gt);
   getch();
}
```

or,

```
gt = (a>b ? a:b);
gt = (gt>c ? gt:c);
```

(*) write a program in C to input two Number after that swap it.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int a,b,c;
Printf("Enter two Number");
scanf("%d%d",&a,&b);
   c=a;
   a=b;
   b=c;
Printf("Swap Number value of a=%d\n
 getch(); value of b=%d\n",a,b);
}
```

y%4==0 && y%100!=0
or
y%400==0

Page No.
Date

(c) write a programming in c to input three Number after swap it.

(b) write a pregam in c to input two Number after that swap it without extra variable.

(d) write a program in c to input any year after that check year is leap or not using ternary operator.

write a program in c to input any character after that print ASCII value

1. *

```
#include <stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
  int a,b,c,temp;
printf(" Enter three Number ");
scanf(" %d %d %d ", &a, &b, &c);
    temp = a;
    a = b;
    b = c;
    c = temp;
printf(" value of a = %d \n value of b = %d \n
    value of c = %d ", a,b,c);
getch();
}
```

d = a;
a = b = c;
b = c = d;

3.

```
#Include<stdio.h>
#Include<conio.h>
  void main()
{
  clrscr();
  float y;
  Printf("Enter any year");
  scanf("%f", &y);   // y%.400==0
  (y%.4==0 && y%.100!=0) Printf("leap year"):
        printf("Not leap year"));
  getch();
}
```

4.

```
#Include<stdio.h>
#Include<conio.h>
  void main()
{
  clrscr();
  char ch;
  Printf("Enter any character");
  scanf("%c", &ch);
  printf("The ASCII value of given character = %d", ch);
  getch();
}
```

(2)

```
  void main()
{
  clrscr();
  int a,b;
  Printf("Enter two Numbers");
  scanf("%d%d", &a, &b);
    a = a+b;
    b = a-b;
    a = a-b;
Printf("after swapping \n value of a = %d\n
```

Page No.
Date 20 02 2018

Control Statements:-

→ It is mainly used to control the flow of execution of statements in a program. There are three types of control statement.

(i) Decision making statement / conditional stmt / Sequential statement
(ii) Looping stmt / Iterative stmt
(iii) Jumping stmt / Ad-hoc stmt

(i) Decision making:- In this stmt we can take a decision for executing block of statements. In this case condition is checked.

There are five types of decision making statements

(a) Simple if stmt / single alternative if stmt.
(b) if-else stmt / Double alternative if stmt
(c) else if ladder stmt. / if-else if stmt.
(d) Nested if stmt / Compound if stmt
(e) Switch Case Statements.

(a) Simple if statement:- In this stmt condition is checked. if condition is true then true part will execute otherwise no actions will generate.

else is the optional part of if-stmt.

Syntax:-

```
if (condition)
{
    block of statements;
}
```

e.g i> if (age >= 18)

mandatory ←
```
{
  printf(" I am Sanjeeni");
  printf(" I am eligible for voting");
}
```

(ii)
```
if (n>0)
{
  printf(" positive number");
}
if(n<0)
  printf(" Negative no");
```
optional for this sign "{}".

Write a program in c to input any character after that check character is upper alphabate lower alphabet or digit using simple if

A to Z
ch=

```
#include<stdio.h>
#include<conio.h>
void main()
{
  clrscr();
  char ch;
  printf(" Enter any character");
  scanf(" %c", &ch);
  if( ch>='A' && ch<='Z')
    printf(" upper alphabet");
  if(ch>='a' && ch<='z')
    printf(" lower alphabet");
  if( ch>='0' && ch<='9')
    Printf(" digit");
  getch();
}
```

if-else statements:— In this statement condition is checked if condition is true then true part will execute otherwise false part will execute.

Syntax:

```
if( condition)
true [ {
        block of stmts;
     }
else
false [ {
        block of stmts;
     }
```

e.g:—

```
if (x%2==0)
printf("even");
else
{
printf("odd");
}
```

(*) write a programming in C to input length & breadth after that find area & perimeter of rectangle if length & breadth are positive otherwise message should flow out as negative length & breadth

→

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int l, b, area, peri;
printf("Enter length and breadth of
    rectangle");
scanf("%d%d", &l, &b");
```

Date 20 02 2018

```
if (l>0 && b>0)
{
area = l*b;
Peri = 2*(l+b);
}
Printf("area of rectangle = %d\n Perimeter of
        rectangle = %d", area, Peri);
else
   Printf("length or breadth is negative");
   getch();
}
```

(c) else if ladder:- In this stmt we can specify more than one condition. if condition is true then true part will execute after that terminate from if statement otherwise next condition is checked.

if is opt

Syntax:-

```
if (condition 1)
{
   block of stmts;
}
else if (condition 2)
{
   block of stmts;
}
else if (condition 3)
{
   block of stmt;
   ---------
   ---------
else if (condition n)
{
   block of stmts;
}
```

e.g:- int digit = 2;
if (digit == 0)
printf("zero");
else if (digit == 1)
Printf("ONE");
else if (digit == 9)
printf("NINE");
else
Printf("Invalid digit");

(*) WAP in C to input four Number after read find smallest Number.

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a, b, c, d, st;
Printf("Enter four Number");
Scanf("%d%d%d%d", &a, &b, &c, &d);
if( a<b && a<c && a<d)
printf("smallest value=%d", a); st=a;
else if ( b<c && c<d)
Printf("b is smallest"); st=b;
else if ( c<d)
st = c;
else
st=d;
Printf("smallest Number=%d", st);
getch();
}
```

write a programming in c to input marks of five subject after that find total marks, Percentage mark and remarks a/c to given condition.

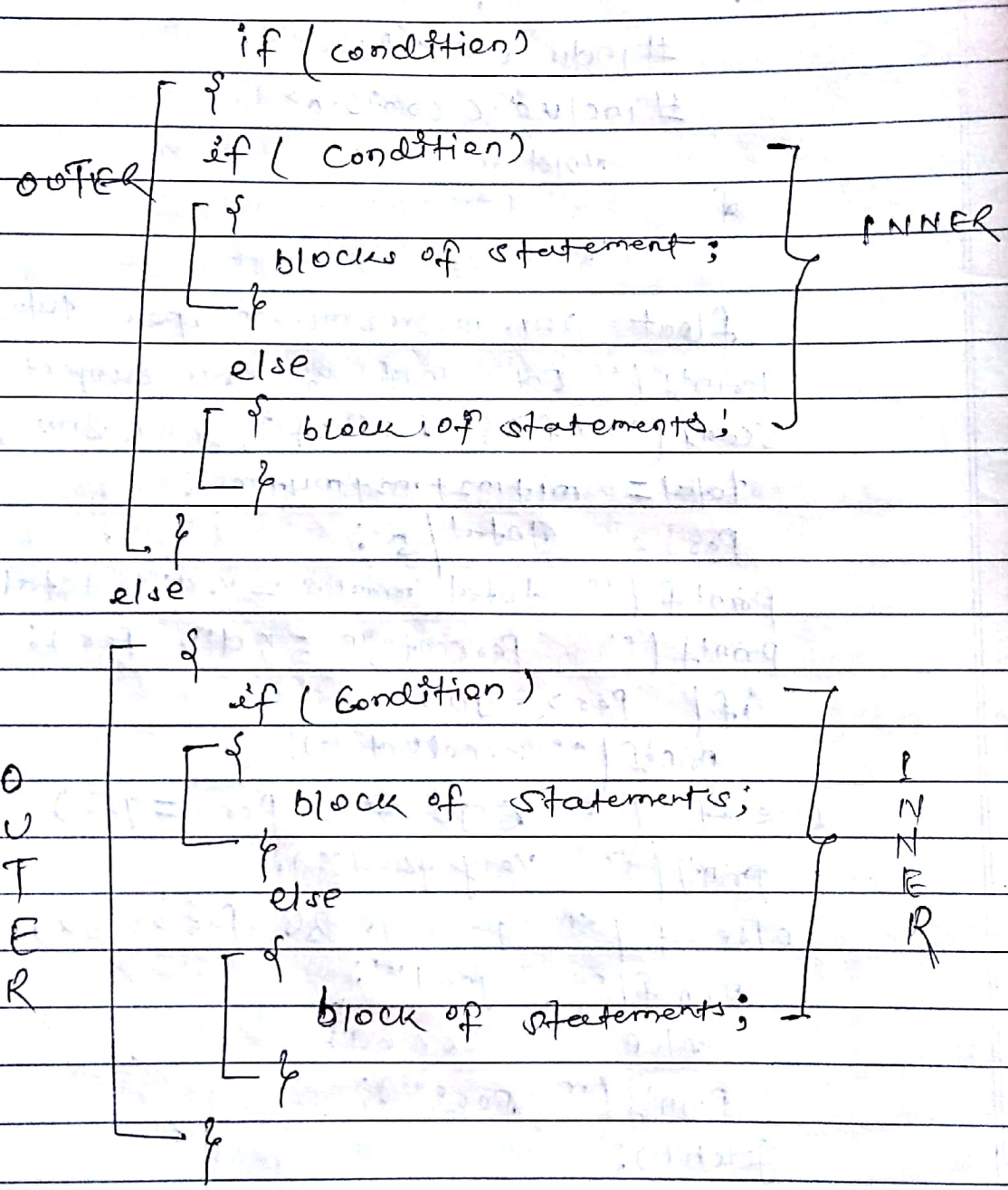
```
#include <stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
float m1, m2, m3, m4, m5, per, total;
printf("Enter marks of five subject");
scanf("%f%f%f%f%f", &m1, &m2, &m3, &m4, &m5);
total= m1+m2+m3+m4+m5;
Per = total/5;
printf("total marks = %d", total);
printf("Percentage = %d", Per);
if( Per>=90)
  printf("excellent");
else if ( Per>=75 )
  printf("Very good");
else if ( Per<75 && Per>=60)
  printf("good");
   else ( Per<60)
  printf("poor");
getch();
}
```

Nested if statement

Page No.
Date

When one if stmts is defind within another if statements called Nested if statement.

Syntax:-



e.g → // To check given no is +ve even, -ve even +ve odd or -ve odd.

```
int n = -10;
if( n % 2 == 0)
 {
   if (n > 0)
   printf("+ve even");
   else
   printf("-ve Even");
 }
else
 {
   if (n > 0)
   printf("+ve odd");
   else
   printf("-ve odd");
 }
```

Nested if

(*) WAP in C to input three Number after that find greatest Number.

(*) WAP in C to input four Number after that find greatest Number.

(*) WAP in C to input any year after that check year is Leap or Not using nested if.

(*) WAP in C to input basic salary of employee after that find gross salary, for given extra allowance on basic salary

if else if

<i> if bs >= 20000 then HRA = 40%, DA = 90%, TA = 40%, MA = 20%

<ii> if bs < 20000 and bs >= 10000 then HRA = 30%, DA = 80%, TA = 30%, MA = 10%

<iii> otherwise HRA = 20%, DA = 60%, TA = 20%, MA = 8%

Page No.
Date

(*)

```
#include <stdio.h>
#include< conio.h>
  void  main()
{
   clrscr();
  int  a, b, c, gt;
printf(" Enter three Number ");
scanf(" %d%d%d ", &a, &b, &c);
  if( a>b)
{
  if( a>c)
    gt = a;
else
    gt = c;
}
else
{ if( b>c)
    gt= b;
else
    gt= c;
}
printf(" greatest Number = %d", gt);
  getch();
}
```

(*)

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

```
if( a>b && a>c && a>d)
   gt = a;
else
  if(b>c && b>d)
   gt = b;
else
   if (c>d)
   gt = c;
else
    gt=d;

getch();
}
```

Logical AND
Not
Nested if

Page No.
Date

(8.)

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    float y;
printf("Enter any year");
scanf("%f", &y);
    if( y%100 !=0)
{
   if( y%4 ==0)
    printf("leap year");
  else
    printf("Not leap year");
}
else
    if( y%400 ==0)
    printf("leap year");
    else
    printf("Not leap year");
}
~~printf("~~  getch();
    }
```

(9.)

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

(2.)

```
void main()
{
clrscr();
int a,b,c,d,gt;
a=20;
if(a>b);
{
if(a>c)
{
  gt=a;
else
  gt=d;
}
else
{
if(c>d)
  gt=c;
else
  gt=d;
}
}
else
{
if(b>c)
{
if(b>d)
  gt=b;
else
  gt=d;
}
else
{
if(c>d)
  gt=c;
else gt=d;
```

(4)

```
#include<stdio.h>
#include<conio.h>
long int bs, gs, hra, da, ta, ma;
printf("Enter basic salary");
scanf("%ld", &bs);

if( bs>=20000)
{
 hra = (bs*40)/100;
  da = (bs*90)/100;
  ta = bs*0.4;
  ma = bs*0.2;
}
else if (bs>=10000)
{
  hra = (bs*30)/100;
  da = (bs*80)/100;
  ta = bs*.3;
  ma = bs*.1;
}
else
{
  hra = bs*.2;
  da = bs*.6;
  ta = bs*.2;
  ma = bs*.08;
}
gs = bs+hra+da+ta+ma;
printf("Basic salary = %ld\n", bs);
printf("Gross salary = %ld\n", bs);
Printf("Hra = %ld\n", hra);
Printf("da = %ld\n", da);
printf("ta = %ld\n", ta);
printf("ma = %ld\n", ma);
getch();
}
```

Page No.
Date

Switch case statement :-

(i) It is used to match only character and integer type value. (* not real type value)

(ii) It is generally used to create menu-based program.

(iii) It is an alternative method of if-else-if stmts. but in this case condition can't be checked.

(iv) Its execution is fast rather than else-if ladder stmts.

Syntax:-

```
switch (variable / constant / expression)
{
    case <value>:
        block of statements;
        break;
    case <value 2>:
        block of stmts;
        break;
        - - - - - - - -
        - - - - - - - -
    case <value n>:
        blocks of stmts;
        break;
    default:
        block of stmts;
}
```

Note:-

* case keyword is used to specify value.
* break is optional part that is used to terminate from switch.
* default is optional that executes if any case does not match.

Page No.
Date 24 02 2018

eg:-

```
int wodn = 2;
switch (wodn)
{
Case 1 :
printf(" monday ");
break;
Case 2:
printf(" Tuesday");
  break;
- - - - - - - - - -
- - - - - - - - - - - -
Case 7;
printf(" ~~Invalid~~ ~~week day number~~ Sunday");
break;
default:
Printf(" Invalid week day number");
}
```

* Write a programming in c to input any alphabet character after that check character is vowel or consonent.

```
#include <stdio.h>
#include< conio.h>
    void main()
{
  clrscr();
  char ch;
Printf(" Enter any alphabet char");
scanf(" %c", &ch);
switch (ch)
```

```
{
    case 'A':
    case 'a':
    case 'E':
    case 'e':
    case 'I':
    case 'i':
    case 'O':
    case 'o':
    case 'U':
    case 'u':
    Printf("vowel");
      break;
    default:
    Printf("consonent");
  }
  getch();
}
```

(*) write a programming in c to input two Numbers and arithmetic operator after that perform arithmetic operation.

→

```
#include<stdio.h>
#include<conio.h>
    void main()
  {
    clrscr();
  char opt;
  int  a,b;
Printf("Enter two Number");
scanf("%d%d", &a, &b);
fflush(stdin); // clear buffer
```

```
printf("Enter any arithmetic operator");
scanf("%c", &opt);
switch (opt)
{
 Case '+':
 printf("Addition=%d", (a+b));
  break;
 Case '-':
 printf ("Substraction = %d", (a-b));
  break;
 Case '*':
   printf ("multiplication = %d", (a*b));
   break;
 Case '/':
printf("Division = %d", (a/b)); break;
  Case '%':
printf("Remainder=%d", (a%b)); break;
default:
  print ("Invalid operator");
  }
  getch ();
}
```

Sanjeev Kumar

Page No.
Date 06 03 2018

Looping Statements :-

It is mainly used to repeat a block of statements multiple times according to the specified condition.

There are four loops in C programming :

(i) for loop :- → entry control loop
(ii) while loop : → entry control loop
(iii) DO-While loop → exit control loop
(iv) Dummy for loop → entry control loop

(i) for loop :-

→ We can ~~say that~~ specify three parts at a time.

(i) Initialization part
(ii) condition
(iii) operational part / increment / Decrement part

Syntax :-

```
for ( initialization part; condition; operational
      part)
{
    block of stmts;
}
```

(Body of loop)

e.g :-

```
for ( i = 1; i <= 100; i++)
{
    printf ("%d\n", i);
}
```

o/p → 1, 2, 3, ..., 100

or

```
i = 1;
for ( ; i <= 100; )
{
    printf ("%d\n", i);
    i++; // i = i+1;
```

Page No.
Date

Standard form of for loop

(i) for(; ;) → infinity for loop

```
{
  ---
  ---
}
```

(2)
```
for( i=1, J=1, K=1; i<=100; i++, J++, K++)
{
  ---
  ---
}
```

(*) write a program in C to print the series of even Natural Number upto 100.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
  int i;
Printf("The series of even natural no
      upto 100\n");
for( i=2; i<=100; i=i+2) // i+=2
  {
    Printf("%d\n", i);
  }
  getch();
}
```

or,

```
for( i=1; i<=100; i++)
{
  if( i%2==0)
  Printf("%d\n", i);
{
```

(*) write a program in C to print the series of odd Natural number from 50 to hundred.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int i;
Printf("The series of odd Natural
    Number up to 100\n");
for(i=50; i<=100 ;i++)
{
  if(i%2==1)
  printf("%d\n", i);
}
getch();
}
```

50
(%2==1)

* write a program in C to print the series of even Natural Number from 100 to 1

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int i;
Printf("The series of even Natural Number
    up to 1\n");
for(i=100; i>=1; i--)
{
  if(i%2==0)
  Printf("%d\n", i);
getch();
}
```

Date

(*) write a program in c to input any Number after that print table.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int n, i;
printf("Enter any Number =");
scanf("%d", &n);
for( i=1; i<=10; i++)
{
  printf("%d\n", (i*n));
}
  getch();
}
```

write a Programming in c to input any Number after that print all factor

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int n, i;
printf("Enter any Number");
scanf("%d", &n);
for(i=1; i<=n; i++)
{
    if( n%i==0)
  printf("%d\n", i);
}
getch();
}
```

* write a program in c to input any Number after that find sum of all factor.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int n, i, s=0;
printf("\n input any Number");
scanf("%d", &n);

for(i=1; i<=n; i++)
{
  if(n%i==0)
{
printf("%d\n", i);
}
  sum = sum+i;
}
}
printf("\n sum of all factor=%d", s);
  getch();
}
```

(*) write a program in c to input any Number after that print only even factor

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int n,i;
printf("\n Enter any Number");
```

Page No.
Date

```
scanf("%d", &n);
for(i=1; i<=n; i++)
{
    if(n%i==0)
    {
        printf("%d\n", i);
        if(i%2==0)
        Printf("only even
        Printf("only even factor=%d", even
        {
            "%d", i);
            factor);
        getch();
    }
```

→ write a program in c to input any Number after that find sum of even and odd factor separately

→ write a program in c to input any Number after that count even or odd factor

→ WAP in C to input any Number after that check No is prime or Not

→ WAP in C to input any Number after that check No is perfect or Not

→ WAP in C to input two Number after that check No is amicable or Not.

→ WAP in C to input any No after that find factorial value.

(1.)

```
#include <stdio.h>
#include <conio.h>
    void main()
{
    clrscr();
    int n, i, esum, osum;
printf("Enter any Number");
scanf("%d", &n)
    for(i=1; i<=n; i++)
{
if(n%i==0)
{ if(n%2==0)
esum = esum+i;
else
```

Scanned by CamScanner

```
        osem = osem + i;
    }
}
printf("\n sum of esem and osem = %d %d",
esemfactor, osemfactor);
getch();
}
```

2

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int n, i, even=0, odd=0;
printf("\n input any number");
scanf("%d", &n);
for(i=1; i<n; i++)
{
if(n%i==0)
{
if(i%2==0)
even++;
else
odd++;
}
printf("\n Total even and odd elements = %d %d\n",
even, odd);
getch();
}
```

3.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int n, i, flag=0;
Printf("Enter any Number");
Scanf("%d", &n);
for(i=2; i<=n/2; i++)
{
  if(n%i==0)
  {
    flag=1;
    break;
  }
}
if(flag==0)
  Printf("%d is a prime No", n);
else
  Printf("%d is a Not prime No", n);
  getches;
}
```

4.

```
#include <stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int n, i, sum=0;
Printf("Enter any Number");
for(i=1; i<n; i++)
{ if(n%i==0)
  { sum=sum+i;
```

```
if( sum==n)
{
Printf("%d is Perfect number");
else
Printf("%d is not Perfect Number");
getch();
}
```

another no:-

(3.)

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
  int n, i, count=0;
Printf("Enter any Number");
Scanf("%d", &n);
for(i=1; i<=n; i++)
 {
   if( n%i==0)
     count = count+1;
 }
if( count==2)
Printf("%d is prime no");
else
  Printf("%d is not prime no");
getch();
}
```

(5.)

```
#include<stdio.h>
#include<conio.h>
Void main()
{
clrscr();
int a, b, i, sum1=0, sum2=0;
Printf("Enter two Number");
Scanf("%d%d", &a, &b);
for(i=1; i<a; i++)
 if(a%i==0)
  { sum1 = sum1+i;
 if(b%i==0)
  sum2 = sum2+i;
for(i=1; i<b; i++)
 if(b%i==0)
  sum2 = sum2+i;
if(sum1==b && sum2==a)
 Printf("No is amical");
else
Printf("No is not amical");
```

(6.)

```
#include<stdio.h>
#include<conio.h>
 Void main()
{
 clrscr();
  int n, i, fact=1;
Printf("Enter any number");
Scanf("%d", &n);
 for(i=1; i<=n; i++)
{
 fact = fact * i;
}
getch(); Printf("factorial Number = %d", fact);
```

While loop:-

Page No.
Date 10 03 2018

While loop :-

→ In this loop we can specify only condition

Syntax:-

```
<initialigation part>
while(condition)
{
    block of statements;
    <increment/Decrement>;
}
```

e.g:

```
i=1;
while(i<=100)
{
printf("%d\n", i);
i++;
}
```

* W A P in c to input any Number after that find sum of all digit.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
int n, sum=0, d;
Printf("Enter any Number");
scanf("%d", &n);
while(n>0)
{
    d= n%10;
    sum= sum+d;
    n= n/10;
}
Printf("sum of all digit = %d", sum);
getch();
}
```

* W A P in C to input any Number after that count digit.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int n, count=0;
Printf("Enter any Number");
scanf("%d", &n);
    while(n>0)
{
    n=n/10;
    count=count+1;
}
Printf("Total count= %d", count);
getch();
}
```

W A program in C to input any Number after that find sum of even and odd digit

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int n, esum=0, osum=0, d;
Printf("Enter any Number");
scanf("%d", &n);
    while(n>0)
{
    d=n%10;
```

```
if(d%2==0)
  esum = esum+d;
else
  osum = osum+d;
  n = n/10;
}
printf("\n Sum of even digit & odd digit =
      %d %d", esum, osum);
getch();
}
```

* Write a programming in c to input any Numbers after that find areaage of even digit.

```
#include <stdio.h>
#include <conio.h>
    Void main()
{
    clrscr();
    int n, esum=0, avg, d, count=0;
printf("\n Enter any Numbers");
scanf("%d", &n);
  while(n>0)
{
  d = n%10;
  if(d%2==0)
  {esum = esum+d;
   count = count+1;
   n = n/10;
  }
  avg = esum/count ;
printf("\n areaage of even digit = %d", avg);
```

getch();
}

Page No.
Date

(*) WAP in C to input any Numbers after that find sum of square of even digit and Qube of odd digit ~~but~~.

```
#include <stdio.h>
#include<conio.h>
    void main
{
    clrscr();
    int n, esum=osum=0,esum=0, osum=0,
        d;
Printf("\n Enter any Number");
scanf("%d", &n);
while( n>0)
{
    d= n%10
    if( d%2==0)
{ esum= esum+d;
    d=d*d;
{esum= esum+d;
else
{
    d= d*d*d;
    osum= osum+d;
}
    n=n/10;
Printf("\n sum of square of even and Qube of
    odd digit", esum, osum);
getch();
}
```

(*) WAP in C to input any number after that check is armstrong or not.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int n, d, sum=0, num;
Printf("Enter any Number");
scanf("%d", &n); num=n;
  while(n>0)
{
  d=n%10;
  sum= sum+d*d*d; n=n/10;
}
  if(sum==num)
  printf("No is armstrong");
else
  printf("No is not armstrong");
}
```

* WAP in C to input any number after that find reverse of digit.

→

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int n, d, rev=0;
Printf("Enter any Number");
scanf("%d", &n);
```

```
while (n>0)
{
    d= n%10;
    rev= rev*10+d;
    n=n/10;
}
printf("reverse of Number = %d", rev);
getch();
}
```

* WAP to input any Number after that check NO is Palindrom or not

* WAP in c to input any Number that find Sum and avg of even and odd digit

* write a program in c to input any Number after that print each digit in words.

* WAP in C to input any Number after that check no is Composite or Not

* WAP in c to input two Numbers after that check both prime number.

1.

```
#include<stdio.h>
#include<conio.h>
    Void main()
{
    clrscr();
int n, d, sum=0, num;
Printf("Enter any Number");
scanf("%d", &n);
num=n;
while(n!=0)
{
d= n%10;
sum= sum*10+d;
n = n/10;
```

```
    if(sum==num)
{
  printf("Number is palindrom number");
else
  Printf("Number is not palindrom No");
}
  getch();
}
```

2.

```
#include<stdio.h>
#include<conio.h>
      void main()
{
      clrscr();          int oCount;
int n, d, sum=0, esum=0, osum=0, avg;
                                ,ecount;
Printf("Enter any Number");
scanf("%d", &n);
  while(n>0)
{
    d=n%10
  if(d%2==0)
{
esum= esum+d;
}else ecount++;
{osum= osum+d;
  }ocount++;
  }
    n=n/10;
  }

avg =   esum/count;
avg =   osum/count;
Printf("Sum and average of even & odd digit =
    %d%.d%.d", esum, osum, avg);
  getch();
}
```

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int n,m, i, c1=0, c2=0;
  printf("Enter two Number");
  scanf("%d%d", &n, &m);
    i=1;
  while(i<=n)
{
   if(n%i==0)
     c1++;
     i++;
  }
  i=1;
  while(i<=m)
{
   if(m%i==0)
     c2++;
     i++;
  }
if( c1==2 && c2==2 && (n-m==2 || n-m==-2))

  printf("Number is twin prime Number");
else
printf("Number is not twin prime
     Number");
  getch();
}
```

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int  n, i, count=0;
printf("\n Enter any Number");
scanf("%d", &n);
for(i=1; i<=n; i++)
{
  if(n%i==0)
    count++;
}
if( count!=2 && n!=1)
    printf("\n No is Composite");
else
    Printf("\n No is not composite");
```

DO-while

Page No.
Date 11 03 2018

In this loop only first time body of loop execute without checking a condition but next time body of loop will execute after checking a condition. So it is also know as exit control loop.
This type of loop is generally used to create menu based program / choice based program.

Syntax:-

```
<initialization part>
do
{
 block of stmts;
 <increment / decrement part>
}
 while (condition);
```

eg:-

```
    i = 1;
do
{
printf("%d\n", i);
    i++;
}
while ( i<=100);
```

Page No.
Date

(*) write a progoam in c to input any binary Number after that convert into decimal Number. this operation must be performed multiple times a/c to requirement.

→
```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int n, dec=0, d, i=1;
    char ch;
do
{
 Printf("Enter any binary Number");
 scanf("%d", &n);
  while( n>0)
 {
  d = n%10;
  dec = dec + d*i;
   i = i*2;
   n = n/10;
 }
 Printf("Decimal no=%d", dec);
  fflush (stdin);
 Printf("Do you want to continue this
       process ? [press y/N]");
 scanf("%c", &ch);
}
  while (ch=='y');
  getch();
}
```

Date

(*) WAP in C to input three Numbers after that find HCf multiple times a/c to requirement.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
  int a,b,c, Hcf;
  char ch;
do
{
Printf("Enter three Number");
scanf("%d%d%d", &a,&b,&c);
 hcf = (a<b && a<c ? a: (b<c ? b: c)
while (a%.hcf!=0 || b%.hcf!=0 || c%.hcf
                                !=0)
{
    hcf--;
}
Printf("H.Cf = %d", hcf);
   fflush(stdin);
  Printf("Do you want to Continue [
        press y/n]?");
 scanf("%c", &ch);
 while(ch=='y');
  getch();
}
}
```

Page No
Date

Dummy for loop →

when terminated for loop with semi-colon is called dummy for loop. this type of loop does not contain body.

Syntax:—

```
for(initialization part ; condition ;
                    operational part);
```

e.g ①
```
int n=5, fact, i;
for( i=1, fact=1; i<=n; fact=fact*i,
                    i++);
Printf("factorial value = %d", fact);
```

②
```
int n= 489, sum, d;
for(sum=0; n>0; d= n%10, sum=sum+d,
                    n= n/10);
Printf("sum of all digit = %d", sum);
```

Nested loop →

when one loop defind within another loop called Nested loop. in this case one loop is called outer loop and other loop is called inner loop.

WAP in C to Print the series of prime Number up to hundred

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int i, count=0, j ;
Printf("the series of prime number
        upto 100\n");
for( i=1; i<=100; i++) // outer loop
```

```
{
    for(j=1; j<=i; j++) // inner loop
    {
    if( i%j ==0)
        count++;
    }
    if( count ==2)
      printf("%d\n", i);
        count=0;
    }
    getch();
}
```

1. * WAP in C to input any Number after that check no is strong or Not
2. * WAP in C to ~~input~~ print the series of fibonacci series up to 20 terms
3. * WAP in C to print Lucas series up to 20 terms
4. * WAP in C to input three Number after that find L.C.M using do-while
5. WAP in C to input any Number after that count Prime digit
6. * WAP in C to print the series of perfect No up to 1000.

Page No.
Date Prog Ques

(2.)

```
#include <stdio.h>
#include<conio.h>
        void main()
{
int n, a, b, c;
n=0, a=1; c=1;
while( c<=n)
{
printf("%d", n);
b = n+a;
n = a;
a = b;
c = c+1;
}
getch();
}
```

n+a=b
n+a=b
n+a=b
n+a=b
0 1 1 2 ...

Printf("n numbers of fabonicii");

(1.)

```
#include <stdio.h>
#include <conio.h>
        void main()
{
clrscr();
int n, i, d, fact=1, sum=0, temp;
printf("Enter a Number");
scanf("%d", &n);
temp = n;
while(n>0)
{
d = n%10;
for(i=1; i<=d; i++)
fact = fact * i;
sum = sum + fact;
n = n/10; fact = 1
}
if( sum == temp)
printf("%d is a strong number", temp);
else printf("%d is not a strong number", temp);
getch();
}
```

Page No.
Date

(4.)

```
#include<stdio.h>
#include<conio.h>
        void main()
{
    clrscr();
  int a,b,c, Lcm;
   char ch;
 do
 {
Printf(" Enter three Number");
Scanf(" %d%d%d", &a, &b, &c);
Lcm = (a>b && a>c ? a : (b>c ? b : c));
while( Lcm%a==0 || Lcm%b==0 || Lcm%c==0)
{
    Lcm++;
}
Printf(" Lcm = %d ", Lcm);
 } fflush(stdin);
Printf(" Do you want to continue
      [Press y/N]?");
Scanf(" %c", &ch);
 while( ch=='y');
  getch();
}
```

(6.)

```
#include<stdio.h>
#include<conio.h>
        void main()
{
    clrscr();
  int n, i, sum=0;
Printf(" The Series of perfect NO upto
      1000 ");
for(i=1; i<=1000; i++)
   n=1;
while( n<=i/2)
```

Date

```
{
  if( i%n==0)
    sum=sum+n;
    n++;
  }
if (sum==i)
printf("%d", i);
  sum=0;
}
getch();
}
```

5.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
int n, i, ~~count~~ count=0;
printf("Enter the any number");
scanf("%d", &n);
~~printf("prime number~~
for( i=
```

1, 1, 1, 3, 5, 9, 17 - - - Series

Page No.
Date

(3)

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a,b,c,d,i;
a=b=c=1;
Printf("%d\t%d\t%d\t", a,b,c);
for(i=1; i<=17 ;i++)
{
d=a+b+c;
Printf("%d\t", d);
a=b;
b=c;
c=d;
}
getch();
}
```

Jumping statements :-

This statement is used to jump from one block of statement into another block of statement there are some jumping statement in C.

(i) Break
(ii) continue
(iii) go to
(iv) return

(i) Break statement :- It is a keyword which is used to terminate from looping statement forcefully w/c to requirement.

Syntax :- break;

e.g :-
```
for( i=1, i<=10; i++)
{
    if( i==5)
        break;
    Printf("%d\n", i);
}
```

(*) write a programming in c to input any Number and any particular digit after that check digit is available or Not

```
void main()
{
    clrscr();
    int n, digit, d, flag=0;
    Printf("Enter any Number");
    Scanf("%d", &n);
    Printf("Enter any searching digit")
    Scanf("%d", &digit);
    while(n>0)
```

```
while( n>0)
{
 d =n%10;
 if( sdigit ==d)
{
    break;
     flag =1;
  }

    n=n/10;
 }
 if (flag ==1)
printf("\n Digit is found");
else
printf("\n Digit is not found ");
  getch();
}
```

* Write a programming in C to input any Number after that check Prime digit is available or not

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
int n, pdigit, d, i,flag=0;count=0;
Printf("\n Enter any Number ");
scanf("%d", &n);
Printf(" Enter any Prime digit");
scanf("%d", &pdigit);
  while(n>0)
{
   d= n%10;
for(i=1;i<=d;i++)
```

```
{
  if( n%i==0)
{
    count++;
  }
  if (count==2)
  {  flag=1;
     break;
     flag=F;
  }
    n=n/10;
  }  count=0;
  if(flag==1)
printf(" pdegit is available");
else
printf(" pdegit is not available");
```

(ii) Continue :-

```
e.g   for( i=1; i<=10; i++)
      {
        if( i==5)
          Continue;
        Printf("%d\n", i);
      }
```

Page No.
Date

* Write a program in c to print the series of even natural Number continue keywords.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int i, ~~n~~;
printf("The series of even natural NO are :-");
for(i=1; i<=100; i++)
{
  if (i%2!=0)
    continue;
  printf("%d\n", i)
}
getch();
}
```

if i i%2≠0

(*) Write a program in c to print only composite ~~digit for given~~ Number up to 100.

```
#include<stdio.h>
#include<conio.h>
    void main()
{   int i, j, count=0;
for(i=1; i<=100; i++)count=0)
{ for(j=1; j<=i; j++)
  if( n%j == 0)
  { count++
if(n!=2 && n!=1)
  continue;
Printf("%d", i);
}
```

```
if( count ==2)
    continue;
if( i!=1)
printf("%d", i);
}
getch();
}
```

goto statements :- This statement is used to repeat a block of statements multiple times o/c to requirement. It is an alternative Method of looping statement but it is traditional method and used to impliment difficult task. Rather than looping statements.

Syntax:-

```
label Name:
if( condition)
{
block of stmts;
goto label Name;
}
```

```
int n=5, i, fact=1;
  i=1
abc;
if( i<=5)
{
fact = fact*i;
  i++;
goto abc;
}
```

Page No.
Date

Write a Programming in C to Input any No. after that find sum of all digit using goto statement.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
 int n,i,sum=0;
Printf("Enter any Number");
Scanf("%d", &n);
    xyz;
if(n>0)
  {
   d = n%10;
    sum = sum+d;
  }
  n = n/10;
   goto xyz;
 }
Printf("%d", sum);
 getch();
}
```

Pattern Design Program

Page No.
Date

1.

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

```
void main()
{
clrscr();
int i, j, line;
Printf("Enter any Number of lines");
scanf("%d", &line);
for(i=1; i<=line; i++)
{
for (j=1; j<=i; j++)
{
Printf("*");
}
Printf("\n");
}
getch();
}
```

2.

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

```
#include<stdio.h>
Void main()
{
clrscr();
int i, j, line;
Printer("Enter Number of lines");
scanf("%d", &line);
for(i=1; i<=line; i++)
{
for(j=line; j>=i; j--)
{
Printf("*");
}
Printf("\n");
}
getch();
}
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
void main()
{
  clrscr();
  int i, j, line;
Printf(" Enter Number of lines ");
scanf(" %d", &line);
 for(i=1; i<=line; i++)
  {
   for(j=1; j<=i; j++)
   {
    printf("%d", j);
   }
   printf("%d", i);
   Printf("\n");
}
  getch();
}
```

(4)

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

```
void main()
{
 clrscr();
  int i, j, line, k=1;
Printf(" Enter Number of line ");
scanf(" %d", &line);
 for( i=1; i<=line ;i++)
  {
  for ( j=1; j<=i; j++)
  {
    Printf("%d", K);
    K++;
  }
   Printf("\n");
}
```

Date

(5.)

```
#include<stdio.h>
#include<conio.h>
    Void main()
{
    clrscr();
int i, j, line, s;
Printf("Enter Number of line");
Scanf("%d", &line);
for(i=1; i<=line; i++)
{                          line-i
for(j=1; j<=i; j++)
{
  Printf("  ");
}
Printf for(j=1; j<=i; j++)
{
  printf("* ");
}
  Printf("\n");
}
```

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

H.W

(1)

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

(2)

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

(3)

```
        0
      1 0 1
    2 1 0 1 2
  3 2 1 0 1 2 3
4 3 2 1 0 1 2 3 4
```

(4)

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

5 4 3 2 1 0 1 2 3 4 5

8969816311

Page No
Date

Array :-

Array

Overview of array :-

(i) It is a special variable which is used to store multiple values of same type.

(ii) It is a linear data structure that works in linear function.

(iii) It is also called subscripted variable bcz It is based on index/subscript

(iv) The value of array can be accessed with the help of Index value.

(v) Index value is positive integer value

(vi) It's lower limit is zero and upper limit is (size-1)

(vii) This variable shares common name for all values.

(*) Advantage :-

(i) It store multiple values at a time.

(ii) It provides simple operation like Insertion /deletion/searching/sorting.

(iii) It shares common name.

Disadvantage :-

(i) It support static memory allocation. It is a technique in which memory is allocated during compile time. In this case memory is wasted due to fixed size of array. It is major drawback of array.

(ii) It stores same types of values. But it can be avoided with the help of structure.

(iii) It execution is slow rather than pointer

Page No.
Date

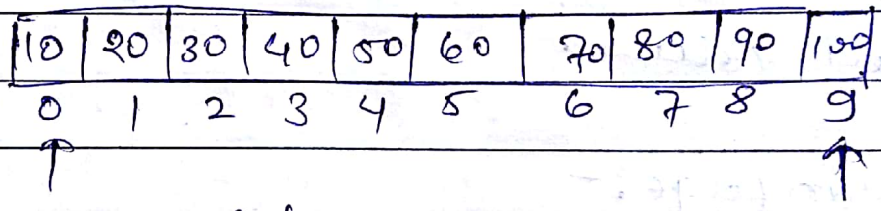
(*) Types of array:- There are three types of array.

(i) One Dimensional Array
(ii) Two Dimensional Array
(iii) Three Dimensional array } → multi Dimensional Array

(i) ONE Dimensional Array:-

→ This type of array is used to store Multiple values in a Singal row. This array consist of only one subscript that represents size of array.

e.g



(*) Declaration :-

Syntax:-

<data type> <variable1>[size], <variable2> [size] ----;

e.g int a[10], b[5];
float P[5];
char ch[20];

Initialization :—

1st Method :-

During declaration time.

optional

Syntax:-

<data type> <variable> [size] = {value1, value2, value3, ----};

e.g. int a[5] = {10,20,30,40,50};

int b[] = {4,5,6};

int char ch[10] = {'c', 'o', 'e'};

2nd Method :—

After declaration

Syntax:-

<variable name> [index] = value;

e.g int a[5];

a[0] = 10;

a[1] = 20;

a[2] = 30;

a[3] = 40;

a[4] = 50;

How to access the element of array?

syntax:-

<variable name> [index];

e.g int a[10] = {10,20,30,40,50};

a[1] = 20;

a[0] = 10;

a[3] = 40;

a[4] = 50;

base address = address of lower limit

Page No. _____
Date 20 03 2018

* write a programming in c to input 10 Numbers after that print it.

→

```
#include <stdio.h>
#include <conio.h>
    void main()
{
    clrscr();
    int a[10],i;
printf("\n Enter 10 Numbers");
for(i=0; i<10; i++)
 {
 scanf("%d", &a[i]);
 }
printf("\n All Numbers are \n");
 for(i=0; i<10; i++)
  {
  Printf("%d\n", a[i]);
  }
  getch();
}
```

④ write a program in c to input 10 Numbers after that print only even elements

```
#include <stdio.h>
#include <conio.h>
    Void main()
{
    clrscr();
    int a[10], i;
printf("\n Enter 10 Numbers");
 for(i=0; i<10; i++)
```

```
{
scanf("%d", &a[i]);
}
Printf("All even Numbers are\n");
for(i=0; i<10; i++)
{
if(a[i]%2==0)
Printf("%d", a[i]);
}
getch();
}
```

(*) Write a program in C to input 10 Numbers after that find sum of even and odd elements.

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a[10], i, osum, esum;
Printf("Enter 10 Number");
for(i=0; i<10; i++)
{
scanf("%d", &a[i]);
}
Printf("sum of even and odd elements are\n");
for(i=0; i<10; i++)
{
if(a[i]%2==0)
esum = esum+a[i];
else
osum = esum+a[i];
```

printf("\n Sum of even and odd elements =%d,%d\n", esum, osum);

~~printf("\n %d\n", a[i]);~~

}

getch();

}

(*) Write a program in C to input 10 Number after that print all element in reverse order.

```
#include<stdio.h>
#include<conio.h>
        void main()
{
    clrscr();
  int a[10], i;
printf("\n Enter 10 Number \n");
for(i=0; i<10; i++)
{
  scanf("%d", &a[i]);
  }
printf("\n All Numbers are in reverse \n");
 for(i=9; i>=0; i--)
{
  printf("\n %d", a[i]);
  }
  getch();
}
```

Page No.
Date

(*) write a Programming in C to input 10 Numbers after that find highest and second highest Number

→
```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int a[10], i, gt, sgt;
    gt= sgt = 0;
Printf("\n Enter 10 Number");
for(i=0; i<10; i++)
{
 scanf("%d", &a[i]);
}
 for(i=0; i<10; i++)
 {
   if ( a[i] > gt)
   {     gt = a[i];
  for( i=0; i<10; i++)
  {
     if ( a[i] > gt && a[i] != gt)
          sgt= a[i];
    }
Printf("\n Greatest NO = %d \n", gt);
printf("\n Second greatest no = %d ", gt);
   getch();
}
```

Date

(*) write a program in c to input 10 Number and any searching value after that check no is available or not.

```
#include<stdio.h>
#include<conio.h>
        void main()
{
    clrscr();
    int a[i], i, sdigit, flag=0;
Printf("Enter 10 Number");
for(i=0; i<10; i++)
{
  Scanf("%d", &a[i]);
}
Printf("Enter any searching element");
for(i=0; i<10; i++)
{
  if(a[i]==sdigit) flag=1;
  break;
  if(flag==1)
Printf("Number is available");
else
Printf("Number is not available");
getch();
}
```

Scanf("%d", &value);

Page No
Date 29

(*) write a program in c to input 10 Numbers and any particular element of the list after that Count frequency of that Number.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int a[10], i, p, count=0;
printf("Enter 10 Number");
 for(i=0; i<10; i++)
{
scanf("%d", &a[i]);
}
printf("Enter any particular element");
scanf("%d", &p);
for(i=0;  i<10; i++)
 {
   if(a[i] == p)
      count++;
 }
 printf("frequency no = %d", p);
  getch();
}
```

Page No.
Date

(*) Write a program in c to input 10 Number after that sort in accending order.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int a[10], i, j, temp;
printf("\n Enter 10 Number");
for(i=0; i<10; i++)
{
   scanf("%d", &a[i]);
}
for(i=0 ; i<10; i++)
{
 for (j=i+1; j<10 ; j++)
 {
    if (a[i]>a[j])
    {
      temp = a[i];
      a[i] = a[j];
       a[j] = temp;
    }
}
}
printf("\n All elements in asending order\n");
for(i=0; i<10; i++)
 {
    printf("\n%d\n", a[i]);
 }
  getch();
}
```

Page No ______

Date

write a program in c to input 10 Number after that print prime Number.

```
#include< stdio.h>
#include< conio.h>
void main()
{
    clrscr();
    int a[i], i, j,count=0;
    Printf(" Enter 10 Number ");
    for(i=0; i<10; i++)
    {
    Scanf(" %d", &a[i]);
    }
    printf(" Enter Number ");
    for(i = 0; i<10; i++)
    {
      ~~d = n%10;~~
      for(j=1 ; j< = a[i] ; j++)
      {
          if ( a[i] % j ==0)
              Count++;
      }
      if( count==2)
      printf(" %d\n", a[i]);
         Count =0;
    }
    getch();
}
```

Q. Write a Program in C to input 10 Numbers after that print.

Q. WAP in C to input 10 Numbers after that print reverse of digit for each Number

Q. WAP in C to input decimal number after that convert into binary Number

Q. WAP in C to input 10 Numbers after that delete any Particular Element from the list

Q. WAP in C to input 10 Numbers after that insert any element at any particular position.

Q.

```
void main()
{
    clrscr();
    int bin[10], dec, d, i, j
printf("\n Enter any decimal Number");
scanf("%d", &dec);
    i=0;
while( dec>0)
{
    d= dec%2;
    bin[i] = d;
    i++;
    dec = dec/2;
}
    Printf("\n Binary NO = ");
for(j = i-1; j>=0; j--)
{
    printf("%d", bin[j]);
}getch();
}
```

(5)

```
void main()
{
    clrscr();
    int list[20], n, i, position, element;
    Printf(" How many numbers to input : ");
    scanf(" %d", &n);
    printf(" Enter the value of list ");
    for( i=0 ; i<n; i++)
    {
    scanf(" %d", &list[i]);
    }
  Printf(" Enter any particular element for
          insertion ");
  scanf(" %d", &element);
  scanf(" Enter the position");
  printf(" Enter the position");
  scanf(" %d", &position);
  for( i=n ; i > position ; i--)
  {
    list[i] = list[i-1];
  }
    list[position] = element;
printf(" After insertion values are \n");
   for(i=0; i<=n ; i++)
  {
  printf(" %d\n", list[i]);
  }
   getch();
}
```

2-D

Page No.
Date

(*) Two Dimensional Array :- This type of array is used to store values in matrix formed such as row and column, this array consist of two subscripts in which first subscripts represents size of row and second subscript represents size of column.

e.g :-

0	10	20	30	→ R_1
1	40	50	60	→ R_2
2	70	80	90	→ R_3
	↓ C_1	↓ C_2	↓ C_3	

Declaration :-

syntax :-

<data type> <variable1 [row] [column],
<variable 2> [row] [column], ------ ;

e.g

int a[3][2], b[3][5];
float c[3][4];

⇒ Initialization (optional)

syntax :-

<data type> <variable name> [row][column] =
{ row value1 }, { row value2 }, { row value3 },
------ };

e.g :- int a[3][2] = { {4,5}, {6,8}, {9,10} };

↘

4	5
6	8
9	10

Date

int a[][] = {{4,5,6},{8,9,10}};

i.e

4	5	6
8	9	10

2×3

Accessing :-

Syntax:- <variable name>[row index][column index];

e.g

	0	1	2
0	4	5	6
1	8	9	10

a[1][2] = 10

Q write a program in c to input any matrix after that print it.

→

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a[3][2], i, j;
Printf("Enter the value of Matrix");
for(i=0; i<3; i++) //Row
{
for(j=0; j<2; j++) //Column
{
scanf("%d", &a[i][j]);
}
printf("Matrix is \n");
for(i=0; i<3; i++)
{for(j=0; j<2; j++)
```

```
{
printf("%d\t", a[i][j]);
}
printf("\n");
}
getch();
}
```

Q. WAP in C to input any number of matrix element and print sum and average of matrix element:

→

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a[3][2], i, j, sum=0, avg;
printf("enter the value of matrix");
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
scanf("%d", &a[i][j]);
}
}
printf("the matrix is\n");
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
sum=sum+a[i][j];
}
}
avg= sum/i*j // sum/6;
```

```
Printf("matrix is \n");
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
Printf("%d\t", a[i][j]);
}
  Printf("\n");
Printf("sum of all elements = %d\n", sum);
printf("Average = %d", avg);
getch();
}
```

(1) write a program in C to input any matrix after that print Transpose of matrix.

(2) write a program in c to input any matrix after that check matrix symmetric or not

(3) WAP in c to input two matrices after that check both matrix are equal or not

(4) WAP in c to input two matrices after that find addition

(5) WAP in c to input two matrices after that find Multiplication

(6) WAP in C to input any square matrix after that find sum of left, Right, Top, and bottom elements of matrix.

(7) WAP in C to input any square matrix after that find sum of both perpendicular diagonal elements.

(8) WAP in C to input any square matrix after that find sum of lower and upper triangle

H.W

Page No.
Date

(a) (*) write a program in c to input any matrix after that find smallest and greatest no element.

(b) (*) WAP in c to input any matrix and search any particular element after that check no is available or not

(c) (*) WAP in c to input any matrix after that print only prime element

(a)

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
int a[3][2], i, st, gt;
  gt=st=a[0][0];
Printf("Enter the value of matrix");
for(i=0; i<3 ; i++)
{
for(j=0; j<2; j++)
{
 scanf("%d", &a[i][j]);
}
}
{
  if( a[i][j] > gt)
     gt= a[i][j];
}
for( i=0; i<3; i++)
{
  for(j=0 ; j<2; j++)
  {
if( a[i][j] != gt && a[i][j] < st)
```

```
            st = a[i][j];
        }
    }
    printf("smallest and greatest element is = %d %d",
           "", st, gt);
    getch();
}
```

(b)

```
#include <stdio.h>
#include <conio.h>
    void main()
{
    clrscr();
int a[3][2], i, selement, flag=0;
printf("Enter any value of matrix");
for(i=0; i<3; i++)
{
 for(j=0; j<2; j++)
{
scanf("%d", &a[i][j]);
printf("Enter any searching element");
scanf("%d", &selement);
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
  if( a[i][j] == selement)
  {
    flag = 1;
    break;
  }
}
  if (flag == 1)
    printf("element is available");
  else
    printf("element is not available");
}
getch();
```

Page No.
Date

(c)

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
int a[3][2], i, j, count=0;
Printf(" Enter the value of matrix");
for(i=0; i<3; i++)
{
 for(j=0; j<2; j++);
 {
 Scanf("%d", &a[i][j]);
 }
 }
Printf(" Enter element ");
for(i=0; i<3 ; i++)
 {
 for( j=1; j<=a[i][j] ; j++)
 {
 if( a[i][j] % j == 0)
     count++;
 }
 if (count == 2)
printf("%d\n", a[i][j]);
 count=0;
}
 getch();
}
```

Matrix mathematical based Program:-

Page No
Date

1. Transpose matrix:-

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int a[2][2], trans[2][2], i, j;
Printf("Enter the value of matrix");
 for(i=0; i<2; i++)
 {
 for(j=0; j<2; j++)
 {
 Scanf("%d", &a[i][j]);
 }
 }
 for(i=0; i<2; i++)
 {
  for(j=0; j<2; j++)
  {
   trans[j][i] = a[i][j];
  }
 }
Printf("Transpose of entered matrix\n");
 for(i=0; i<2; i++)
  {
  for(j=0; j<2; j++)
 {
  Printf("%d\t", trans[i][j]);
 }
  Printf("\n");
 }
 getch();
}
```

4.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
 int a[3][2], b[3][2], c[3][2], i,j;
printf("\n Enter the value of first
      matrix");
for(i=0; i<3; i++)
 {
 for(j=0; j<2; j++)
 {
  scanf("%d", &a[i][j]);
 }
 }
 Printf("\n Enter the value of second matrix");
 for(i=0; i<3; i++)
  {
  for(j=0; j<2; j++)
  {
   scanf("%d", &b[i][j]);
  }
  }
  for(i=0; i<3; i++)
  {
   for(j=0; j<2; j++)
 {
  c[i][j] = a[i][j]+ b[i][j];
 }
 }
Printf("\n Sum of given matrix is \n");
  for(i=0; i<3; i++);
{
for(j=0; j<2; j++)
```

```
    {
      printf("%d", c[i][j]);
    }
      printf("\n");
    {
      getch();
    }
```

Transpose matrix — 2nd method

```
#include <stdio.h>
#include <conio.h>
        void main()
{
        clrscr();
  int a[3][2], i, j;
  printf("Enter value of matrix");
  for( i=0; i<3; i++)
  {
   for( j=0; j<2; j++)
  {
   printf( scanf("%d", &a[i][j]);
  }
}
printf(" Matrix is \n");
for( i=0; i<3; i++)
  {
   for ( j=0; j<2; j++)
{
   printf("%d\t", a[i][j]);
}
   printf("\n");
}
```

a[i][j]=[

```
printf("Transpose of matrix\n");
for(i=0; i<2; i++)
{
for(j=0; j<3; j++)
{
printf("%d\t", a[j][i]);
}
printf("\n");
}
getch();
}
```

* symmetric matrix

```
int flag=0

for(i=0; i<3; i++)
{
for(j=0; j<3; j++)
{
if(a[i][j]!=a[j][i]
{
flag=1;
break;
}
if(flag==1)
{ break;

if(flag==0)
printf("symmatrix matrix");
else
printf("Not symmetric matrix");
}
getch();
}
```

equal or not

(3.)

```
int a[i][j], b[i][j], flag=0;

for (i=0; i<3; i++)
{
for (j=0; j<2; j++)
 {
   if( a[i][j] != b[i][j])
     {
       flag=1;
     } break;
      if (flag==1)
          break;
     }
   if (flag==0)
   printf(" equal matrix");
else
   printf(" Not equal matrix");
}
 getch();
}
```

(2) Symmetric matrix:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr(); int a[3][2],i,j,flag=0;
Printf("Enter the value of matrix");
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
scanf("%d", &a[i][j]);
}
}
Printf("matrix is \n");
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
Printf("%d\t", a[i][j]);
}
} Printf("\n");
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
if(a[i][j]!=a[i][j]
{
flag=1;
break;
}
if(flag==0)
Printf("Symmetric matrix");
else
Printf("Not symmetric matrix");
getch();
}
```

equal or not?

(3)

```
void main()
{
    int a[3][2], b[3][2], i, j, flag=0;
    Printf("Enter value of first matrix");
    for(i=0; i<3; i++)
    {
        for(j=0; j<2; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    for(i=0; i<3; i++)
    {
        for(j=0; j<2; j++)
        {
            printf("%d\t", a[i][j], b[i][j]);
        }
    }
    for(i=0; i<3; i++)
    {
        for(j=0; j<2; j++)
        {
            if(a[i][j] != b[i][j])
            {
                flag=1;
                break;
            }
        }
        if(flag==1)
        Printf("equal matrix");
    else
        Printf("Not equal matrix");
    }
    getch();
}
```

⊛ Three dimensional Array

Page No.
Date 29 | 03 | 18

→ Three dimensional Array is used to store multiple matrices of same order. This array consist of three subscripts in which first subscripts represents number of matrices & last two sub-scripts represents order of matrices.

e.g:-

	0	1
0	4	5
1	8	9
2	10	12

0

	0	1
0	15	20
1	25	30
2	35	40

1

	0	1
0	50	55
1	60	65
2	70	75

2

	0	1
0	80	85
1	90	95
2	100	105

3

Declaration:-

<data type> <variable> [No. of matrix] [row] [column]

e.g:- int a[4][3][2], b[4][2][2];

Initialization:-

Syntax:-

<data type> <variable> [No. of matrix] [row] [column] =

{
{ {row1}, {row2}, ------},
{ {row1}, {row2}, -----},
{ {row1}, {row2}, ------},

};

e.g:- int a[4][3][2] = {
{ {4,5}, {8,9}, {10,12} },
{ {15,20}, {25,30}, {35,40} },

};

/* */ Multiple

Page No.
Date

Accessing :-

Syntax :- <variable> [matrix index] [row index] [column index]

e.g
a[2][1][1] = 65
a[1][0][1] = 20
a[2][2][0] = 70

Q. Write a program in c to input 5 matrix after that print matrix.

```
#include<stdio.h>
#include<conio.h>
    void main ()
{
    clrscr();
  int a[5][3][2], i,j,k;
Printf("Enter five matrix\n");
  for(i=0; i<5; i++)  //for no. of matrix
{
  Printf("Enter value of matrix %d", i+1);
  for(j=0; j<3 ; j++)
{
for(k=0; k<2; k++) // No. of column
{
  Scanf("%d", &a[i][j][k]);
}
}
}
  Printf("All matrices are\n");
  for(i=0; i<5; i++)
{
  Printf("matrix %d\n", i+1);
```

```
for( j=0; j<3; j++)
{
    for( k=0; k<2; k++)
    {
        printf("%d \t", a[i][j][k]);
        printf("\n");
    }
}
getch();
}
```

(*) WAP in C to input five matrix after that find sum of elements for each matrix.

→

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
int a[5][3][2], i, j, k, sum=0;
printf(" Enter five matrix\n");
for( i=0; i<5; i++)
{
printf(" Enter value of matrix %d", i+1);
for( j=0 ; j<3; j++)
{
for( k=0; k<2 ; k++)
{
scanf(" %d", &a[i][j][k]);
}
}
}
```

```
printf("All matrices are\n");
for(i=0; i<5; i++)
{
printf("matrix %d\n", i+1);
for(j=0; j<3; j++)
{
for(k=0; k<2; k++)
{
printf("%d\t", a[i][j][k]);
sum = sum + a[i][j][k];
}
printf("Sum of all elements = %d",
sum);
sum=0;
}
getch();
}
```

String handling

Page No.
Date 29 03 2018

(*) **String:-** It is a collocation of zero, one or more characters that are enclosed within double quote (" "). We know that string must be terminated with NULL character.

"COC" | C | O | C | \0 |

"Sanjeev" | S | a | n | j | e | e | V |

"S" | S | \0 |

" " | \0 |

⇒ How to hold string value ?

→ character type array variable is used to store string value.

e.g:- char name[30];

Name = "COC" ; ~~Error~~ // it is not ✗ possible ~~during~~ after declaration time.

char name[30] = "COC" ;

char name[30] = {'C', 'O', 'C'};

Page No.
Date

Q. Write a program in c to input any string after that print it.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
char name[20] = {'c', 'o', 'e'};
char str[20];
Printf("Enter any string");
Scanf("%.s", str)
Printf("my College name = %s \n", name);
printf("string = %s", str);
    getche();
}
```

Q. Write a program in c to input any string after that print each character.

→

```
#include<stdio.h>
#include<conio.h>
{
clrscr();
char str[20];
int i;
Printf("Enter any string");
Scanf("%.s", str);
Printf("All chars are \n");
for(i=0; str[i] != '\0'; i++)
{
Printf("%.c \n", str[i]);
}
getche();
}
```

Write a program in C to input your name after that find length of your name.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  char str[20];
   int i, count=0;
Printf("\n Enter any string"); printf("\n Enter the length of name")
for(i=0; str[i]!='\0'; i++)
  {
    count=count+1;
  }
Printf("\n length of name=%d", count);
 getch();
}
```

Write a program in C to input your name after that count vowel and consonent character.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  char str[20];ch;
   int i, V=0, C=0;
Printf("\n Enter any string");
for(i=0; str[i]!='\0'; i++)
 {
    ch= str[i];
if(ch=='A'||ch=='a'||ch=='E'||ch=='e'||
  ch=='I'||ch=='i'||ch=='O'||ch=='o'||
  ch=='U'||ch=='u')
```

```
        v++;
    else
        c++;
}
printf("\n Total vowel char = %d\n Total
Consonet char = %d", v, c);
getch();
}
```

(*) Write a program in c to input any string after that count lower alphabet, upper alphabet, digit and special character

→
```
#include<stdio.h>
#include<conio.h>
        void main()
{
    clrscr();
    char str[20], ch;
    int i, u=d=d=s=0;
printf("\n Enter any string");
scanf("%s", str);
for(i=0; str[i]!='\0'; i++)
{
    ch=str[i];
    if( ch>=65 && ch<=90)
        u++;
else if( ch>=97 && ch<=122)
        u++;
else if( ch>=0 && ch<=9)
        d++;
else
        s++;
}
```

```
printf("Total upper character = %d \n total lower
    character = %d\n total digit = %d\n total
    special character = %d\n", u, l, d, s);
getch();
}
```

(3*) write a program in c to input your name and any particular character after that check character is available or not.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  char str[20], ch;
    int i, flag=0;
printf("Enter any string");
scanf("%s", str);
printf("Enter any particular character");
scanf("%c", &ch);
for(i=0; str[i] != '\0'; i++)
{
    if(ch==str[i])
    {
      flag=1;
      break;
    }
}
    if(flag==1)
printf("character is found");
else
printf("character is not found");
getch();
}
```

Page No.
Date

(*) W A P in c to input your name after that print in reverse order.

```
#include <stdio.h>
#include<conio.h>
void main()
{
clrscr();
char str[20];
int i, count=0;
Printf("Enter any string");
Scanf("%s", str[i]);
printf("Enter length of your name");
for( i=0; str[i]!='\0'; i++)
{
count++;
}printf("reverse of name=");
for( i=count-1; i>=0; i--)
{
printf("%c", str[i])
}
getch();
}
```

Q. Write a program in C to input any string after that check string is palindrom or not.

→

```
#include <stdio.h>
#include <conio.h>
void main()
{
clrscr();
char str[20];
int i, j, count=0, flag=0;
Printf("Enter any string");
Scanf("%s", str[i]!='\0'; i++)
{
count++;
}
for(i=0, j=count-1; i<j; i++, j--)
{ if(str[i]!=str[j])
{
flag=1;
break;
}
}
if(flag==0)
printf("palindrom string");
else
printf("not palindrom string");
getch();
}
```

⑧ Two Dimensional String 2D String

⑧ String library function ⑧

→ These functions are used to manupulate string operation. These function are available in <string.h> header file.

There are some string library function in c.

ⓐ strlen()
ⓑ strcpy()
ⓒ strcat()
ⓓ strcmp()
ⓔ strlwr()
ⓕ strupr()
ⓖ strrev()

ⓐ ~~Strlen();~~

ⓐ strlen():- this function Returns the length of string.

Syntax:- int strlen("string value");

eg:-

```
int rs;
rs = strlen("welcome");
Printf("%d", rs);
```

ⓑ strcpy():- this function is used to copy one string into another string.

```
char * strcpy (str1, str2);
```

eg

```
Char str[20];
strcpy (str, "Adarsh");
Printf("%s", str);
```

↳ Adarsh

Page No.
Date

(c) strcat() :- This function is used to concatenate one string with another string.

```
char strcpy (str1, str2);
```

e.g:-

```
char str1[20] = "wel", str2[20] = "come";
strcat (str1, str2);
printf("%s", str1);
```

→ welcome

(d) strcmp() :- This function is used to compare string value. It returns integer value according to ASCII value.

```
int strcmp (str1, str2);
```

Note:-

(i) str1 > str2 => >0
(ii) str1 < str2 => <0
(iii) str1 == str2 => 0

e.g:- (a) strcmp ("Ram", "Ram"); => 0

ASCII values

(b) strcmp ("ABC", "abc"); => -32

65 - 97 = -32

(c) strcmp ("abc", "ABC") => 32

97 65

(d) strcmp ("Raman", "Ravan"); => -ve

m < v

Page No.
Date

(e) strlwr() :→ This function is used to convert upper string into lower string.

Syntax:-

```
Char * strlwr (string value)
```

eg:→

```
printf("%s", strlwr("COC"));
```

op→ coc

```
char str[20] = "WELCOME";
strlwr(str);
printf("%s", str);
```

↳ welcome

(f) strupr() :→ This function is used to convert lower string into upper string.

Syntax:-

```
char * strupr (string value)
```

eg:→

```
printf("%s", strupr("sanjeev"));
```

opt :→ SANJEEV

```
char str[20] = "sanjeev";
strupr(str);
printf("%s", str);
```

↳ SANJEEV.

(g) strrev() :→ This functions Returns string value in reverse order.

```
char * strrev (string value)
```

eg :→ printf("%s", strrev("come"));

→ emoc

```
CHAR str[20] : "WELCOME";
strrev(str);
printf("%s", str);
```

↳ EMOCLEW

Two Dimensional String

character type array variable

→ This type of variable is used to hold more than one string value. In this case first subscript represents no of string and second subscript represents length of string.

⇒ Declaration

char <variable>[No. of strong][length of string];

e.g:- char name[10][30];

⇒ Initialization

eg

char name[10][30] = {"Sanjeev", "krishana", "prabham", "aderash"};

⊛ WAP in to input Name of 10 students. and afterthan print it.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
char name[10][30];
    int i;
Printf("Enter name of 10 students");
 for(i=0; i<10; i++)
 {
  scanf("%s", name[i])
 }

 printf("All names are\n");
  for(i=0; i<10; i++)
 {
 Printf("%s\n", name[i]);
```

```
    getch();
}
```

→ How to access any particular element of any string?

e.g name[0][3] = ? (v)
name[2][4] = ? (y)

8. Write a Program in c to input Name of five students after that count Number of characters for each Name.

```
#include<string.h>
#include<stdio.h>
#include<conio.h>
    void main()
{
clrscr();
char name[5][30];
    int i;
printf("Enter name of five students\n");
for(i=0; i<5; i++)
{
 scanf("%s", name[i]);
}
printf("Name\t length\n");
for(i=0; i<5; i++)
{
printf("%s\t %d", name[i], strlen(
                        name[i]));
{
 getch();
}
```

Without string :-

```
void main()
{
  int i, j, count=0;
Printf("Enter name of five students\n");
for(i=0; i<5; i++)
{
  scanf("%s", name[i]);
}
printf("Name\t length\n");
for(i=0; i<5; i++)
{
  for(j=0; name[i][j]!='\0'; j++)
  {
    count++;
  }
printf("%s\t%d\n", name[i], count);
    count=0;
}
getch();
}
```

(*) :- write a programming in C to input Name of five students after that count lower and upper characters for each Name.

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
char name[5][30];
int i, j, u, l,
Printf("Enter name of five students\n");
for(i=0; i<5; i++)
{ scanf("%s", name[i]);
```

```
printf("Name \t upper \t lower \n");
for(i=0 ; i<5 ; i++)
{
for( j=0; name[i][j] != '\0' ; j++)
{
        Name[i][j]                Name[i][j]
if( s >= 'A' && s <= 'Z')
        u++;
else
        l++;
}
printf(" %.s \t %d \t %d \n", name[i], u, l);
        u=l=0;
}
getch();
}
```

(*) write a program in c to input Name of 10 students after that sort in accending order.

(*) write a program in c to input Name of 10 students after that print reverse of all Names.

(*) WAP in c to input Name of 10 std. after that print greatest Name (*) write a program in c to input Name of 10 students and any searching Name after that check Name is available or not

(*)

Page No.
Date

```
#include <string.h>
#include< conio.h>
#include< stdio.h>
  void main()
{
 char name[10][30], temp[30];
    int i,j;
Printf("Enter name of 10 student");
for( i=0; i<10; i++)
 {
   scanf("%s", name[i]);
 }
 for( i=0; i<10; i++)
   {
    for( j = i+1; j<10; j++)
  {
    if( strcmp (name[i], name [j]) > 0)
   {
     strcpy (temp, name[i]);
     strcpy (name[i], name[j]);
     strcpy ( name[j], temp);
   }
  }
 }
 Printf("All names in accending order\n");
 for( i=0; i<10; i++)
{
  Printf("%s\n", name[i]);
}
  getch();
}
```

function

Page No.
Date

* A function is a block of statement which is used to design for performing any particular task a/c to requirement.
* We know that c support procedural programming to reduce complexity of program. This concept can be implemented with the help of function.
* The main purpose of function is to reduce the complexity of program. In this case a large program can be devided into more than one parts.
* It provide readability and make user-friendly program.
(a) A function can parametrised
(b) A function can be return type.

(c) Types of function :—
There are two types of function
(i) pre-defind function
(ii) user-defind function / Customised function

(i) pre-defind function:— This type of function must be predefind in any language.
eg:- strcmp, strcpy, pow(), getch(), printf() etc.

(ii) User-defind function:— This type of function is created with the help of user. Its major advantage is that user can Customised source code a/c to requirement

Page No.
Date 05 04 18

function Architecture :-

→ It represents pattern of function as return type, Non-return type, Parameterised etc.

There are four architecture of function

(i) A function with no return type and No parameters

(ii) A function with return type but No Parameters

(iii) A function with no return type but with parameters

(iv) A function with return type and parameters.

(i) No Return and No Parameters.

Syntax:-

```
void <function name> ()
{
    block of stmts;
}
```

eg:-

```
void findadd ()
{
    int a=10, b=20, add;
    add = a+b;
    Printf("addition = %d", add);
}
```

→ How to call ?

Syntax:-

```
<function name> ();
```

eg:→ findadd ();

(ii) Return but NO Parameter

→

Syntax:-

```
<data type><function name> ()
{
    block of stmts;
    return (variable);
}
```

e.g:-

```
int findadd()
{
    int a=10, b=20, add;
    add = a+b;
    return (add);
}
```

⇒ How to call ?

Syntax:-

1st method

```
<variable> = <function name> ();
int xx;
xx = findadd();
printf("Addition = %d", xx);
```

2nd

```
printf("Addition = %d", findadd());
```

(iii) No return type but Parametre

→ Syntax:-

```
void <function name> (<data type> <parametre 1>,
                      <data type> <parametre 2>, ---)
{
   block of stmts;
}
```

e.g

```
void findadd (int a, int b)
{
  int add;
  add = a+b;
  Printf("Addition = %d", add);
}
```

⇒ How to call ?

```
<function Name> (Value of parametre);
```

eg:

```
findadd (20, 30)
```

(iv) Return type and Parametre :-

```
<data type> <function name> (<data type> <
      Parametre 1>, <data type> <parametre 2>---)
{
  block of stmts;
  return (variable);
}
```

e.g:-

```
int findadd (int a, int b)
{
  int add;
  add = a+b;
  return (add); //or return add;
```

→ How to call ?

<variable> = <function name> (value of parameter);

eg:- rs = findadd (20, 30);

or,

printf ("addition = %d", findadd (20,30));

(*) function Prototype :- It represents pattern of function which is created by user. It must be required if user defind function is written after main function. Otherwise it is not required.

(*) write a program in c to input three Numbers after that find greatest number using all types of pattern of function.

→

```
#include<stdio.h>
#include<conio.h>
  void findgreatest();// function Prototype
void main()
{
  clrscr();
  findgreatest();
}
void findgreatest()
{
  int a,b,c,gt;
  Printf("Enter three Number");
  Scanf("%d%d%d", &a, &b, &c);
gt = (a>b && a>c ? a:(b>c ? b:c));
Printf("greatest no=%d", gt);
```

2nd pattern

```
#include<stdio.h>
#include<conio.h>
int findgreatest();
void main()
{
    clrscr();
    Printf("Greatest no=%d", findgreatest());
}
int findgreatest()
{
    int a,b,c,gt;
    Printf("Enter three Number");
    scanf("%d%d%d", &a, &b, &c);
    gt=(a>b && a>c ? a: (b>c ? b:c));
    return (gt);
}
```

good

```
#include<stdio.h>
#include<conio.h>
void findgreatest(int, int, int);
void main()
{
    clrscr();
    int a,b,c;
    Printf("Enter three Number");
    scanf("%d%d%d", &a, &b, &c);
    findgreatest(a,b,c);
    getch();
}
```

actual parameter

formal

```
void int greatest findgreatest(int a, int b, int c)
{
    int gt;
    gt=(a>b && a>c ? a: (b>c ? b:c));
    Printf("Greatest no=%d", gt);
}
```

```
#include<stdio.h>
#include<conio.h>
int findgreatest(int, int, int);
void main()
{
clrscr();
int a,b,c,rs;
printf("Enter three Number");
scanf("%d %d %d", &a, &b, &c);
rs = findgreatest(a,b,c);
printf("Greatest no=%d", gt);
getch();
}
int findgreatest(int a, int b, int c)
{
int gt;
gt=(a>b && a>c? a:(b>c?b:c));
return(gt);
}
```

Page No.
Date

(*) Write a program in C to input any No after that find factorial value using function.

→

```
#include<stdio.h>
#include<conio.h>
int ~~void~~ findfact();
    void main()
{
    clrscr();
  printf("e factorial Number=%d", findfact());
}

  int findfact()
{
  int n, i, fact=1;
printf("Enter any Number");
scanf("%d", &n);
  for(i=1; i<=n; i++)
{
  fact=fact*i;
}
  return(fact);
}
```

Page No.
Date

(*) write a program in c to input any Number after that find sum of all digit using function.

```
#include <stdio.h>
#include <conio.h>
  void findsum();
  void main()
{
  clrscr();
  findsum();
}
  void findsum()
{
    int n, i, d, sum=0;
  Printf("Enter any Number");
  scanf("%d", &n);
    while(n>0)
{
    d= n%10;
  sum= sum+d;
    n= n/10;
}
  printf("Sum of all digit=%d", sum);
}
```

Page No.
Date

(*) write a program in c to input amount, rate and time as parametre with function after that find simple interest.

Simple interest

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

void findsinterest(int, float, float)
void main()
{
clrscr();
findsinterest(200, 10, 3);
getch();
}
void findsinterest(int a, float r, float t)
{
int si;
si = (p*r*t)/100;
printf("simple interest = %d", si);
}
```

(*) write a program in c to input any Number as parametre after that check no is prime or not.

→

```
#include<stdio.h>
#include<conio.h>
void findprime(int a)
{
int i, count=0;
```

Date

```
printf("Enter any Number");
scanf("%d", &n);

for(i=1; i<=n; i++)
{
    if (n%i == 0)
    count++;
}
if (count == 2)
printf("No is prime");
else
printf("No is not prime);
}

void main ()
{
    clrscr();
    findprime (11);
    getch();
}
```

or,

Page No.
Date

Q. write a program in c to input length and breath as parameter after that find area of Rectangle.

```
#include<stdio.h>
#include<conio.h>
void findarea(int l, int b)
{
    int ar;
    ar = l*b;
Printf("area of rectangle = %d", ar);
{
    void main()
    {
    clrscr();
    int l, b;
Printf("enter length and breath");
Scanf("%d %d", &l, &b);
    findarea(l, b);
getch();
    }
```

Q. write a program in c to input four Number as parameter with return type function after that find smallest.

```
#include<stdio.h>
# include<conio.h>
int findsmallest(int a, int b, int c,
                 int d)
{
    int st;
if( a<b && a<c && a<d)
    st = a;
```

```
else if( b<c && b<d)
    st= b;
else if( c<d)
    st= c;
else
    st= d;
return(st);
}
main()
{
    clrscr();
    int a, b, c, d, rs;
    printf("Enter four Numbers");
    scanf("%d %d %d %d", &a, &b, &c, &d);
    rs=findsmallest(a, b, c, d);
    printf("smallest no=%d", rs);
    getch();
}
```

write a program in c to input any Number as parameter with return type function after that count all factors.

```
#include<stdio.h>
#include<conio.h>
int countfactor(int n)
{
    int i, count=0;
    while(n>0)
    {
    if(n%i==0)
    count++;
    }
    return(count);
}
```

```
void main()
{
clrscr();
~~Countfactor(12);~~
printf("all factor = %d", Countfactor(12));
}
```

Q. WAP in C to input three Numbers as parameter with return type function After that find H.C.F & L.C.M.

Ans: →

```
#include<stdio.h>
#include<conio.h>
~~void main()~~
int findHcf(int a, int b, int c)
{
int Hcf;
Hcf=(a<b && a<c ? a: (b<c? b: c));
while( a%hcf!=0 || b%hcf!=0 || c%hcf!=0)
{
hcf--
}
return(hcf);
}
void main()
{
clrscr();
int a, b, c, rs;
printf("Enter three Number");
scanf("%d %d %d", &a, &b, &c);
rs = findhcf(a, b, c);
printf("Hcf = %d", hcf);
getch();
}
```

```
#include <stdio.h>
#include<conio.h>
int findlcm(int p, int q, int r)
{
    int lcm;
    lcm = (p>q && p>r ? p : (q>r ? q : r));
    while(lcm%p==0 || lcm%q==0 || lcm%r==0)
    {
        lcm++;
    }
    return(lcm);
}
void main()
{
    clrscr();
    int p,q,r,rs;
    printf("Enter three Number");
    scanf("%d%d%d", &p, &q, &r);
    rs = findlcm(p,q,r);
    printf("lcm = %d", lcm);
    getch();
}
```

Date

(9.) WAP in C to input two Numbers as parameter after that check Number is amicable or not.

```
#include<stdio.h>
#include<conio.h>
void checkami ( int a, int b)
{
   int i, sum1=0, sum2=0;
   for( i=1; i<a; i++)
   {
     if( a%i==0)
     sum1 = sum1+i;
   }
   for( i=1; i<b; i++)
   {
     if( b%i==0)
     sum2 = sum2+i;
   }
   if( sum1 = b && sum2 = a)
    Printf(" NO is amical ");
   else
    Printf(" NO is not amicable ");
}
   void main()
   {
     int a,b;
     Printf(" Enter two Number ");
     Scanf(" %d%d", &a, &b);
     Checkami (a,b);
     getch();
   }
```

Date

4. WAP in C to input any number and any particular digit as parameter after that check ~~Number~~ is digit is available or Not.

```
#include <stdio.h>
#include<conio.h>
void checkpdigit ( int n) {
int pdigit, d, flag=0;
printf("\n Enter any particular digit ");
scanf("%d", &pdigit);
while ( n>0)
{
d = n%10;
if( pdigit==d)
{
break;
flag=1;
}
n=n/10;
}
if( flag==1)
printf("\n Digit is found ");
else
printf("\n Digit is notfound ");
}
void main()
{
clrscr();
int n, ch;
printf("\n Enter any Number ");
scanf("%d", &n);
ch = checkpdigit(n);
getch();
}
```

Page No.
Date

(2) WAP in C to input any binary number as parameter after that convert into decimal.

```
#include <stdio.h>
#include <conio.h>
void convertdec(int n)
{
int d, dec=0, i=1;
while(n>0)
{
d= n%10;
dec= dec+d*i;
i= i*2;
n= n/10;
}
void main()
{
clrscr();
int n, s;
printf("Enter any binary digit");
scanf("%d", &n);
convertdec(n);
printf("decimal =10 = %d", dec);
getch();
}
```

Page No.
Date

Parameter :- parameter is a type of special variable that is supplied during declaration and calling of function. There are two types of parameter.

(i) formal parameter
(ii) Actual parameter

(i) formal Parameter :→ Those parameter that are passed during declaration of function called formal Parameter it is treated as local variable.

(ii) Actual Parameter :- Those Parameter that are supplied during calling a function called Actual parameter. It is also known as argument.

e.g:-
```
void findadd (int a, int b)
{
   ===
}

findadd (10,20);
```

formal parameter

Actual parameter

Date

WAP in C to input two Number after that find arithmetic operation using function

```
#include<stdio.h>
#include<conio.h>
    void input();
    void findadd();
    void findsub();
    void findmulti();
    void finddiv();
int a,b; // global variable
    void main()
{
    clrscr();
    input();
    findadd();
    findsub();
    findmulti();
    finddiv();
    getch();
}
    void input()
{
printf("Enter two Number");
scanf("%d%d", &a, &b);
}
    void findadd()
{
    printf("Addition = %d\n", (a+b));
}
    void findsub()
{
printf("Substraction = %d\n", (a-b));
}
```

```
void findmulti()
{
printf("multiplication = %d\n", (a*b));
}
void finddiv()
{
printf("division = %d\n", (a/b));
```

(*) WAP in C to input length and breadth after that find area and peri of rectangle.

→

```
#include<stdio.h>
#include<conio.h>
void input();
void findarea();
void findperi();
int l, b;
void main()
{
clrscr();
input();
findarea();
findperi();
getch();
}
void input()
{
printf("Enter length and breadth");
scanf("%d%d", &l, &b);
}
```

```
void findarea ()
{
printf(" area of rectangle = %d\n", (l*b));
}

void findperi ()
{
printf(" perimeter of rectangle = %d\n", 2(l+b));
}
```

WAP in c to input any matrix after that print it using function.

```
#include<stdio.h>
#include<conio.h>
void input();
void display();
int i, j, a[3][2]
void main()
{
clrscr();
input();
display();
getch();
}
void input()
printf(" Enter the value of matrix ");
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
scanf(" %d %d", &a[i][j]);
}
```

```
void display()
{
for(i=0; i<3; i++)
{
for(j=0; j<2; j++)
{
printf("%d\t", a[i][j]);
}
printf("\n");
}
}
```

* Passing array variable as parameter in function.

* WAP in c to input multiple Number a/c to size and any Searching element after that check value is available or Not using function.

→

```
#include<stdio.h>
#include<conio.h>
void searching(int[], int, int);
void main()
{
int a[50], size, svalue, i;
printf("enter the size of list/array");
scanf("%d", &size);
printf("enter %d numbers", size);
for(i=0; i<size; i++)
{
scanf("%d", &a[i]);
```

```
printf("Enter any searching element ");
scanf("%d", &svalue);
searching (a, svalue, size);
getch();
}
void searching (int n[], int svalue, int size)
{
int i, flag=0;
for(i=0; i<size; i++)
{
if (n[i]==svalue)
{
flag=1;
break;
}
}
if (flag==1)
printf("searching is found");
else
printf("searching is not found");
}
```

Page No.
Date

(*) WAP in C to input 10 Number as Parameter after that find greatest no using return type function.

```
#include<stdio.h>
#include<conio.h>
void findgreatest (int [10]);
void main()
{
int a[10], i, gt;
Printf("Enter 10 Number ");
for(i=0; i<10; i++)
{
scanf("%d", &a[i]);
}
Printf("greatest no = %d", findgreatest(n));
getch();
}
int findgreatest (a[i], i)
{
int gt = 0;
for(i=0; i<10; i++)
{
if(a[i] > gt)
gt = a[i];
}
return (gt);
}
```

Date

(*) WAP in c to your name after that convert into lower and upper case using Parameterise function.

→
```
#include<stdio.h>
#include<conio.h>
 void convertupper(char name[])
{
 char rs[20];
 int i, j;
  j=0;
for(i=0; name[i]!='\0'; i++)
{
 rs[j] = name[i]-32;
    j++;
 }
rs[j] = NULL;
 printf("your string in upper case = %s", rs);
}
void convertlower(char name[])
 {
  char rs[20];
  int i, j;
  j=0;
for(i=0; i<strlen(name); i++)
{
  rs[j] = name[i]+32;
    j++;
 }
 rs[j] = NULL;
printf("your string in lower case = %s", rs);
}
```

Page No.
Date

```
void main()
{
clrscr();
Convert upper("abcd");
Convert lower("ABCD");
getch();
}
```

Q How to create Header file?

→ We know that header file is a organise function we can save user-defind function with the help of .H extension in the following location.

c:\turboc4\tc\include

Page No
Date 14/04/2018

ⓐ Pointer ⓐ

→ It is a special variable which is used to hold memory address of variable.

→ It supports dynamic memory allocation.

→ In this case memory is allocated during runtime so memory is not wasted.

It is major advantage of pointer.

→ Its execution is fast because it works on memory address directly. It is also advantage

→ It is not secure because we can change the value of variable using memory address so c and c++ are not secure language.

We know that Java is highly secure bcz it does not support pointer. It is drawback of pointer regarding security

→ It provides complex operation Rather than array. It is also drawback.

(*) Operator used in pointer :- There are two operator.

(i) Indirection operator :- [*]

(ii) Address of [&]

(i) Indirection operator :→ This operator is used to declare pointer variable. other book is that it points value at memory address.

(ii) Address of :- This operator returns memory address of variable.

Page No
Date 14/04/2018

✱ Pointer ✱

→ It is a special veriable which is used to hold memory address of variable.

→ It supports dynamic memory allocation.

→ In this case memory is allocated during runtime so memory is not wasted. It is major advantage of pointer.

→ His execution is fast because it works on memory address directly. It is also advantage

→ It is not secure because we can change the value of variable using memory address so C and C++ are not secure language. We know that Java is highly secure bcz it does not support pointer. It is drawback of pointer regarding security

→ It provides complex operation Rather than array. It is also drawback.

✱ Operator used in pointer :- There are two operator.

(i) Indirection operator :- [*]

(ii) Address of [&]

(i) Indirection operator :→ This operator is used to declare pointer variable. other book is that it points value at memory address.

(ii) address of :- This operator returns memory address of variable.

Date

(*) How to declare pointer variable ?

→ Syntax:-

<data type> *<variable1>, *<variable2>, ... ;

e.g:-

int *a, *b ;
float *p, *q ;
double *c, *d ;

} These are Pointer variable.

(*) How to return memory address of any variable ?

→ Syntax:-

&<variable name>

e.g:- &a :- Address of a
&b :- Address of b

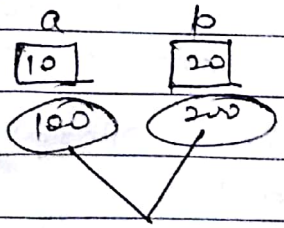
(*) How to hold memory address in Pointer variable ?

→ Syntax:-

<Pointer Variable name> = &<variable>;

e.g:- int a, b;
int *p, *q;

int a, b;



memory address

a = 10;
b = 20;

Page No
Date

(5) How to get value from memory address.?

→ Syntax:-

imp

*<pointer variable>

⇒ *p :- get value

p :- get address

(6) Simple program:-

```
#include<stdio.h>
#include<conio.h>
    void main()
{
 clrscr();
 int   a=10, b=20;
 int *p, *q ; // pointer variable
  p= &a ; // p holds address of a
  q= &b; // q    "      "    b
Printf(" the value of a = %d\n", a);
Printf("    value of a = %d\n", *p);
Printf(" memory address of a = %u\n", &a);
Printf(" memory addres of a = %u\n", p);

Printf(" the value of b = %d\n", b);
Printf(" value of b = %d\n", *q);
Printf(" memory address of b = %u\n", &b);
Printf(" memory address of b = %u\n", q);

 getch();
}
```

Page No.
Date

④ Write a program in C to input two Numbers after that find addition and ~~Substraction~~ using Pointer;

→
```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
    int a, b, *p, *q, add, sub;
  Printf("Enter two Number");
  scanf("%d %d", &a, &b);
    p = &a;
    q = &b;
    add = *p + *q;
    sub = *p - *q;
Printf("Addition = %d & Substraction = %d",
            add, sub);
  getch();
}
```

2nd Method

```
    void main()
{
    clrscr();
  int *a, *b, *add, *sub;
printf("Enter two Numbers");
scanf("%d %d", a, b);
  *add = *a + *b;
  *sub = *a - *b;
printf("Addition = %d \n Substraction = %d",
*add, *sub);
  getch();
}
```

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int *a, *b, *c, *gt;
printf("Enter three Number");
scanf("%d %d %d", a, b, c);
  for(i=0; i<3
  {
*gt = (*a>*b && *a>*c? *a:(*b>*c? *b:
                                *c);

printf("Greatest = %d", *gt);
  getch();
}
```

Pointer arithmetic

Page No. _____
Date 21 | 04 | 2018

→ In pointer arithmetic operation Memory address can be increased or decreased a/c to the size of data type.

e.g:-

```
int a, *p;
p = &a ; // p = 100
```

p+1 => 100 + 1*2 => 102

p+2 => 100 + 2*2 => 104

p+3 => 100 + 3*2 => 106

p-5 => 100 - 5*2 => 090

Pointer with array:-

Base address:→ The memory address of first element of array is called base address.

e.g:-

```
int a[10];
```

100	102	104	106	108	110	112	114	116	118
10	20	30	40	50	60	70	80	90	100
0	1	2	3	4	5	6	7	8	9

Note:- Only array variable name without subscript holds base address.

e.g:-

```
int *p;
```

Case I:- `p = a ; // p = 100`

↳ represents base address of array

Case II:- `p = &a[0] ; // p = 100`

Size of Pointer = 2 bytes

Case I: int *a; => sizeof(a)

Memory address	Value
a+0 => 100	*(a+0) => 10
a+1 => 102	*(a+1) = 20
a+2 => 104	*(a+2) = 30
a+3 => 106	*(a+3) = 40

Case II

Memory address	Value
p+0 = 100	*(p+0) = 10
p+1 = 102	*(p+1) = 20
p+2 = 104	*(p+2) = 30
p+3 = 106	*(p+3) = 40

Write a program in c to input 10 Number after that print it using pointer.

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a[10], i;
Printf(" Enter 10 Number ");
for( i=0; i<10; i++)
{
scanf(" %d", a+i);
}
Printf(" All elements are\n");
for( i=0; i<10; i++)
{
Printf(" %d\n", *(a+i));
}
getch();
}
```

Size of Pointer = 2 bytes

Case I: int *a; => sizeof(a)

Memory address	Value
a+0 => 100	*(a+0) => 10
a+1 => 102	*(a+1) = 20
a+2 => 104	*(a+2) = 30
a+3 => 106	*(a+3) = 40

Case II

Memory address	Value
p+0 = 100	*(p+0) = 10
p+1 = 102	*(p+1) = 20
p+2 = 104	*(p+2) = 30
p+3 = 106	*(p+3) = 40

Write a program in c to input 10 Number after that print it using pointer.

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a[10], i;
Printf(" Enter 10 Number ");
for( i=0; i<10; i++)
{
scanf(" %d", a+i);
}
Printf(" All elements are\n");
for( i=0; i<10; i++)
{
Printf(" %d\n", *(a+i));
}
getch();
}
```

Date

(8) Write a program in c to input 10 number after that find greatest number using pointer.

```
#include<stdio.h>
#include<conio.h>
    void main ()
    int a[10], i, gt=0;
printf("Enter 10 number");
for(i=0;  i<10; i++)
{
 scanf("%d", a+i);
}
{
    if(*(a+i)> gt )
     gt= *(a+i) ;
}
for(i=0;  i<10; i++)

printf("Greatest no= %d ", gt);
 getch();
}
```

(9) Write a program in c to input 10 Numbers after that Print in reverse order.

```
#include<stdio.h>
#include<conio.h>
    Void main()
    int a[10], i ;
Printf("Enter 10 Number");
for(i=0;  i<10; i++)
{
scanf("%d",  a+i);
}
```

Page No.
Date

```
printf("\n All Number in reverse order\n");
for(i=9; i>=0; i--)
{
printf("%d\n", *(a+i));
}
getch();
}
```

(*) Two Dimensional Array ~~with~~ using Pointer :-

```
int a[3][2] = { 10,20 }, { 30,40 }, { 50,60 };
or
int a[3][2] = {10,20,30,40,50,60};
```

Note :- memory allocation of two dimensional array is performed row wise.

i.e

Row1		Row2		Row3	
10	20	30	40	50	60
100	102	104	106	108	110

↑ Base address

(*)

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  int a[3][2] = {10,20,30,40,50,60};

   int i,j;
Printf("\n All elements are\n");
for(i=0 ; i<6; i++)
  {
   Printf("\n %d\n", *(a+i));
  }
   getch();
}
```

Page No.
Date

How to calculate memory address of element in two Dimensional ~~Array~~?

Syntax:-

=> Base address + (row index * no of columns + column index) * size of data type

e.g:- int a[3][2] = { {10, 20}, {30, 40}, {50, 60} };

memory address of a[2][0]

=> 100 + (2*2+0) *2

=> 100+8

=> 108

WAP in c to input any matrix after that print it using pointer.

```
#include< stdio.h>
#include< conio.h>
   void main()
{
clrscr();
int row, column, i,j;
int a[20][20];
Printf("\n Enter the size of row & column");
Scanf("%d %d", &row, &column);
printf("\n Enter the value of matrix");
for(i=0; i<row; i++)
{
  for(j=0; j<column; j++)
  {
  Scanf("%d", a+(i*column+j))
  }
}
```

```
Printf("matrix is \n");
for(i=0; i<row; i++)
{
for(j=0;j<column; j++)
{
Printf("%d\t", flat(i*column+j));
}

Printf("\n");
}
getch();
}
```

prop ⑧ Dynamic memory allocation :—

→ It is a technique in which memory is allocated during runtime.
In this case memory is not wasted.
It is performed with the help of the following method.

(i)	malloc ()	stdlib.h or alloc.h
(ii)	calloc ()	
(iii)	realloc ()	
(iv)	free ()	

⑧ malloc () :→ It is memory allocation method which is used to allocate memory for pointer variable in single block. This function is available stdlib.h

This function returns generic type Pointer. (i.e void *p)

Page No.
Date

Syntax:-

```
Void * malloc (size);
```

e.g:-

```
int *p;
p=(int *) malloc (20);
```

or

```
p= (int *) malloc (sizeof (int) *10);
```

(int *) — Casting; sizeof (int) — 2 byte

(*) write a program in c to input 10 Numbers after that print all Number using malloc function

→

```
#include< stdlib.h>
#include< stdio.h>
#include< conio.h>
    void main()
{
  clrscr();
  int *p,i;
  p = (int *) malloc (sizeof(int) *10);
  Printf(" Enter 10 Numbers ");
  for( i=0;  i<10; i++)
  {
    Scanf(" %d", p+i);
  }
  Printf(" All elements are\n");
  for(i=0;  i<10; i++)
  {
    printf(" %d", *(p+i));
  }
  getch();
}
```

Second method

Page No.
Date

```
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
  int *p, i, *s;
  p = (int*) malloc (sizeof(int)*10);
      s = p;
printf("Enter 10 Number ");
  for(i=0; i<10; i++)
  {
  scanf("%d", p);
    p++;
  }
  printf("All elements are\n");
  for(i=0; i<10; i++)
  {
    printf("%d", *(s+i));
  }
    getch();
}
```

```
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int *p,i,*s,flag=0;
p=(int*)malloc(sizeof(int)*10);

printf("\n Enter 10 Numbers\n");
for(i=0;i<10;i++)
{
scanf("%d",p+i);
}

printf("\n Enter any searching value");
scanf("%d",s);
for(i=0;i<10;i++)
{
if(*(p+i)==*s)
{
flag=1;
break;
}
if(flag==1)

printf("\n searching is found");
else
printf("\n Not found");
getch();
}
```

Date

* Calloc function :- This function is used to allocate memory for pointer variable during runtime. It is similar to malloc function but it allocates memory for multiple blocks.

Syn :- void * calloc (size, No. of blocks);

e.g.:→ int *p;

p= (int *) calloc (sizeof (int), 10);

* Write a program in C to input 10 Numbers after that ~~find~~ ~~or~~ count positive & negative value using calloc function.

→

```
#include< stdlib.h>
#include< stdio.h>
#include< conio.h>
  void  main()
{
  clrscr();
  int *p,  pcount=0, Ncount=0, i,n;
Printf(" How many number to input ?");
scanf(" %d", &n);
 p= (int * ) calloc (sizeof (int), *n);
 Printf(" Enter value for list");
 for( i=0;  i<n ; i++)
 {
  scanf(" %d", (p+i))
 }
 for( i=0;  i<n; i++)
 {
   if( *(p+i) > 0)
    pcount++;
```

```
else
    ncount ++;
}
printf("All values are\n");
for(i=0; i<n; i++)
{
    printf("%d\n", *(p+i));
}
printf("Total positive numbers = %d\n",
                                pcount);
printf("Total negative number = %d\n",
        //free(p); deallocate  ncount );
getch();          memory
}
```

* free() function :— This function is use to deallocate memory during runtime.

syntax:—

free (Pointer variable);

exp:→ free(p);

(*) Realloc ():→ This function is used to reallocate an allocate memory during Runtime. We can say that we can increase or decrease memory with the help of Realloc function.

syntax:— ~~void *~~

void *realloc (void *ptr, new size)

eg:— int *p;

p = (int*) malloc (size of (int) *10);

p = (int*) realloc (p, 40);

Page No.
Date

Implementation of realloc function

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

void main();
{
clrscr();
int *p, i, *np;
p= (int *) malloc (sizeof(int) * 3);

  *p=10;
  *(p+1)=20;
  *(p+2)=30;
np = (int *) realloc (p, sizeof(int) *5);
printf("All elements are\n");
for(i=0; i<5; i++)
{
printf("%d\n", *(np+i));
}
getch();
}
```

V.V.I

Some different type of pointer

(i) double pointer / pointer to pointer
(ii) void pointer / Gemic pointer
(iii) Null pointer
(iv) dangling pointer
(v) wild pointer

(i) Double pointer :- This type of pointer variable is used to hold memory address of pointer variable.

eg:-

```
int **a, **b;
```

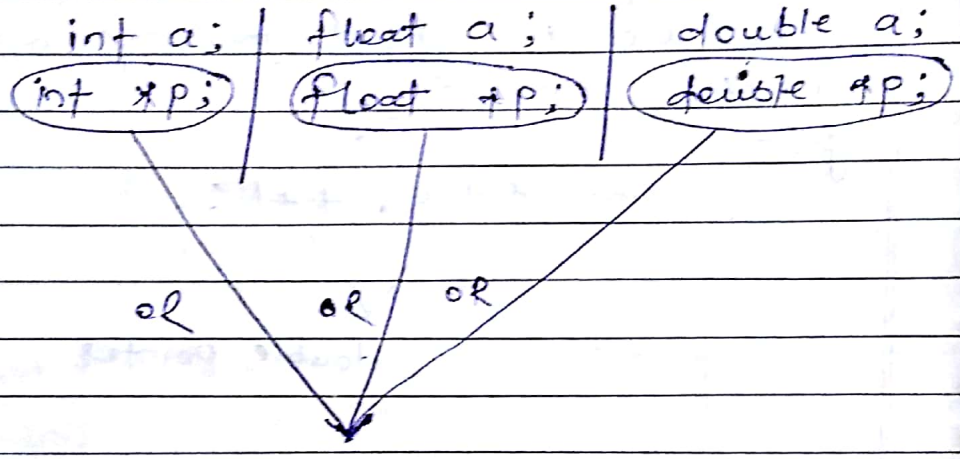
double pointer

```
void main()
{
    clrscr();
    int a, *p, **q;
    a=10;
    p=&a;
    q=&p;
  Printf("The value of a=%d\n", a);
  Printf("The value of a=%d\n", *p);
  Printf("The value of a=%d\n", **q);
    getch();
}
```

(ii) Void pointer / Generic pointer :→

→ It is used to hold memory address of any type variable. It must be declared with the help of void keyword so it is called void pointer.

e.g :—




```
void main()
{
    clrscr();
    int a=10;
    float b=10.5;
    char ch='A';
    void *p; // Generic / void pointer
    p=(int *)&a;
    Printf("The value of a=%d\n",*p);
    p=(float *)&b;
    Printf("The value of b=%f\n",*p);
    p=(char *)&ch;
    Printf("The value of ch=%c",*p);
    getch();
}
```

(iii) Null pointer :- When pointer variable is initialized with NULL value called NULL pointer. The value of NULL pointer is zero.

```
void main()
{
  clrscr();
  int *a = NULL;
  printf("The value of null
          pointer = %d\n", *a); //0
  getch();
}
```

(iv) Dangling Pointer :- The process of deallocating memory from pointer variable is called Dangling pointer.

```
void main()
{
  clrscr();
  int *p;
  p = (int *) malloc (40);

  free (p); // Dangling pointer
  getch();
}
```

VVI

Passing Parameter in function :—
There are two method for passing Parameter in function.

(i) Call by value
(ii) Call by reference

(i) Call by value :— In this method we supply value of parameter so in this case the value of actual parameter can't be changed

(ii) Call by reference :— In this method we supply memory address of variable so in this case the value of actual parameter is changed.

Eg:

Call by value:-

```
#include <stdio.h>
#include<conio.h>
void swap (int a, int b)
{
    int c;
    c=a;
    a=b;
    b=c;
printf("After swapping \n value of a=%d
    \n value of b=%d\n", a, b);
}
void main()
{
clrscr();
int a, b; // Actual parameter
printf("Enter two Numbers");
scanf("%d %d", &a, &b);
Printf("Before Calling value of actual
parameter\n value of a =%d\n value of b=%d\n",
a,b);
swap (a, b); // Calling a function
Printf("After Calling value of actual
    parameter\n value of a=%d\n
    value of b=%d\n", a, b);
getch();
}
```

O/P

```
Enter two numbers
10
20
Before Calling value of actual parameter
    value of a=10
    value of b=20
After swapping value of a=20
    value of b=10
```

After calling value of actual parameter

value of a = 10

value of b = 20

(*) Call by reference

```
#include <stdio.h>
#include <conio.h>
void swap (int *a, int *b)
{
    int c;
    c = *a;
    *a = *b;
    *b = *c;
printf("\n After swapping \n value of a = %d\n value of b = %d\n",
    *a, *b);
}
void main()
{
    clrscr();
    int a, b; // Actual parameter
    printf("\n Enter two Numbers");
    scanf("%d %d", &a, &b);
printf("\n Before calling value of
    Actual Parameter \n value of a = %d\n
    value of b = %d\n", a, b);
swap(&a, &b); // Calling a function
printf("\n After calling value of
    actual parameter \n value of a = %d\n value
    of b = %d\n", a, b);
getch();
}
```

o/p :- Enter two Numbers

10
20

Before Calling value of actual Parameter

value of a = 10

Value of b = 20

After swapping

value of a = 20

value of b = 10

After Calling value of Actual Parameter

value of a = 20

value of b = 10

//

Pointer with function

Page No.
Date

(*) WAP in c to input three Numbers as Pointer variable as parameter with pointer type after that find greatest and smallest no using return type function

→

```
#include<stdio.h>
#include<conio.h>
int findgreatest(int *a, int *b, int *c)
int findsmallest(int *a, int *b, int *c)
{
int st, gt;
gt=(
```

```
int findgreatest(int *, int *, int *);
int findgreatest(int *, int *, int *);
void main()
{
int a,b,c;
Printf("Enter three Numbers");
Scanf("%d%d%d", &a, &b, &c);
Printf("Greatest no=%d", findgreatest(&a, &b, &c));
Printf("Smallest no=%d", findgreatest(&a, &b, &c));
getch();
}
int findgreatest(int *a, int *b, int *c)
{
int *gt;
*gt=(*a>*b && *a>*c?*a:(*b>*c?*b:*c));
return(*gt);
}
int findgreatest(int *a, int *b, int *c)
```

Page No.
Date

(*) WAP in c to ~~input two~~ create a function as named check amicable which accepts two parameters as pointer type after that check both NO. is amicable or not.

```
#include<stdio.h>
#include<conio.h>
void checkamicable(int *a, int *b)
{
    int i, sum1=0, sum2=0;
    for(i=1; i<*a; i++)
    {
        if(*a%i==0)
        sum1=sum1+i;
    }
    for(i=1; i<*b; i++)
    {
        if(*b%i==0)
        sum2=sum2+i;
    }
    if(sum1==*b && sum2==*a)
    printf("NO is amicable");
    else
    printf("NO is not amicable");
}
    Void main()
{
    clrscr();
    int a, b;
printf("Enter two NO");
scanf("%d%d", &a, &b);
    checkamicable(&a, &b);
    getch();
}
```

Pointer with String

Page No.
Date

Q i.e

```
char str[20] = "welcome";
```

or

```
char *str = "welcome";
```

How to access the particular char from string using pointer?

e.g:-

```
char *str = "welcome";
```

~~*str~~

w	e	l	c	o	m	e	\0
100	101	102	103	104	105	106	107

(str+0) ⇒ 100	*(str+0) ⇒ w
(str+1) ⇒ 101	*(str+1) ⇒ e
(str+2) ⇒ 102	*(str+2) ⇒ l

→ WAP in C to input any string after that print each character using pointer

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
  char *str;
    int i;
  printf("Enter any string");
  scanf("%s", str);
  printf("All characters are\n");
  for(i=0; *(str+i)!='\0'; i++)
  {
    printf("%c\n", *(str+i));
  }
  getch();
}
```

* Structure *

→ It is collection of different types of element as integer, float, character etc.

→ It is mainly used to represent record of any entity. Record can be Roll no, Name, Course, fee of student.

→ It provides user defind data types.

* How to create structure :-

* Struct :- This keyword is used to create structure.

syntax :- struct < structure name >
{
<data type 1> <variable 1>, <variable 2>, ---
---- - ----
<data type 2> <variable 1>, <variable 2>,
- - - - ;
<data type N> <variable 1>, <variable 2>, ;
} [structure variables];
↓
optional part

e.g :-
```
struct student
{
  int roll no;
  float fee;
  char name [20];
} S1, S2;
```
↓
Structure variable

NOTE → Structure must be terminated with semi colon.

Page No.
Date

Structure variable

→ Structure variable is used to access the member of structure.

→ it can be used with two operators which are given

(i) Dot operator [.]

(ii) Arrow operator [→]

(i) Dot operator :→ [.] it is used with Non-pointer structure variable.

(ii) Arrow operator [→] :- it is used with pointer structure variable

How to create structure variable ?

→ There are two methods for creating structure variable.

→ During declaration time

→ After declaration time

Syntax:-

```
struct <structure name> <structure variable 1>,
        <structure variable 2>, ------;
```

e.g:- `struct student s1, s2, s3;`

`struct student` → user-defind data type

`s1, s2, s3` → structure variable

address of (&) operator is not used in string

Page No.
Date

* How to access member of structure?

Case 1 :- for Non-Pointer structure variable.

Syntax:-

e.g:- <structure variable>. <member Name>

```
Struct student s1
s1. roll no
s1. name
```

Case 2 :- for Pointer structure variable

Syntax:-

<structure variable> → member name

e.g:-

```
Struct student *s;
s → roll no
s → name
```

Q WAP in C to create a structure as named student to input roll no, name, course, and fee. After that print it.

struct.cpp

→

```
#include <stdio.h>
#include< conio.h>
 Struct student
{
 int roll no;
char name[20], course[10];
float fee;
};
 void main()
{
 clrscr();
Struct student s;
```

Date

```
Printf("Enter Roll no, name, Course and fee");
scanf("%d %s %s %f", &s.roll no, s.name, s.Course,
      &s.fee);
Printf("Roll no = %d \n Name = %s \n Course = %s \n
      fee = %f", s.Roll no, s.name, s.Course,
       s.fee);
       getch();
}
```

Q WAP in c to create a structure as named employee to input eid, name, post and basic salary after that print all record using pointer (structure variable).

→

```
#include<stdio.h>
#include<conio.h>
 Struct employee
{
  int eid;
 char name[20], post[20];
  float basic salary;
};
  void main()
{
  clrscr();
  Struct employee *e;
Printf("Enter employee id, name, post basic salary");
scanf("%d %s %s %f", &e->eid, e->name,
        e->post, &e->salary);
Printf("employee id = %d \n Name = %s \n post = %s \n
    salary = %f", e->eid, e->name, e->post,
    e->salary);
  getch();
}
```

Page No.
Date

* WAP in c to input three Numbers after that find greatest using structure.

* WAP in c to ~~input~~ create a ~~structure~~ structure as named student to input roll.no. name, course and fee of more than one student.

Q. Write a program in c to create a structure as named rectangle. Input length and breadth after that find area and perimeter of rectangle.

```
#include <stdio.h>
#include <conio.h>
    struct rectangle
{
  int length, breadth, area, peri;
};
    void main()
{
  clrscr();
    struct rectangle r;
printf("Enter length and breadth");
scanf("%d%d", &r.length, &r.breadth);
    r.area = r.l * r.b;
    r.peri = 2*(r.l + r.b);
printf("Area of rectangle = %d\n perimeter of
rectangle = %d", r.area, r.peri);
  getch();
}
```

Array type structure variable

→ This type of variable is used to hold multiple records at a time

```
struct student s[10]
```

↓

It holds records of 10 students.

s[0] → 1st student
s[1] → 2nd student
- - - - - -
- - - - - -
s[9] → 10th student

Array type structure variable

→ Used to hold more records

i.e struct < structure name > < structure variable > [size];

e.g:-

```
struct Employee
{
    int eid;
    char name[20];
} e[10];
```

→ After Declaration :-

```
struct Employee eid[10];
```

```
#include <stdio.h>
#include <conio.h>
struct student
{
    int roll no;
    char name[20], course[10];
    float fee;
};
void main()
{
    clrscr();
    int n, i;
    struct student s[50];
    Printf("How many records ?");
    Scanf("%d", &n);
    for(i=0; i<n; i++)
{
    Printf("Enter roll no, name, Course, and
fee of the student = %d\n", (i+1));
Scanf("%d %s %s %f", &s[i].roll no, s[i].name,
    s[i].course, &s[i].fee);
```

```
printf("All records of student \n");
    for( i=0; i<n; i++)
{
 Printf("Records of student %d \n", (i+1));
 printf("Roll no = %d \t Name = %s \t Course = %s \t
 fee = %f \n", S[i].roll no, S[i].name, S[i].course,
  S[i]. fee);
}
  getch();
}
```

Q. WAP in C to create a structure as named student. Input roll no, name, course and fee of many students a/c to requirements. after that print All records.

Page No.
Date

Nested Structure

2nd Method

e.g:-
```
struct department
{
    int did;
    char dname[20], dlocation[20];
};
struct employee
{
    int eid;
    char ename[20], post[20];
    struct department d; // inner structure
};
```

⇒ How to access member of inner structure?

Syntax:-

<Outer structure variable>.<inner structure var>.member name;

e.g:-
```
struct Employee e;
e.d.did = 10;
```

⇒ How to access member of outer structure?

Syntax:-

<outer structure var>.member name;

eg:- e.eid = 20;

Q. WAP in C to create outer structure as named student to input roll.no, name, course & fee. Again create inner structure as named DOB to input date of birth of student. After that print all records.

→
```
#include<stdio.h>
#include<conio.h>
struct student
```

```
{
    int roll no ;
    char name [20], course [20];
      float fee;
     struct DOB
    {
      int dd, mm, yy;
    } d;   // inner structure
    } s;   // outer structure

      void main()
    {
      clrscr();
    Printf("Enter roll no, name, course and
      fee");
    scanf("%d %s %s %f", &s.roll no, s.name,
          s.course, &s.fee);
    Printf("Enter Date of birth \n");
    Printf("Enter day of DOB");
    scanf("%d", &s.d.dd);
    Printf("Enter month of DOB");
    scanf("%d", &s.d.mm);
    Printf("Enter year of DOB");
    scanf("%d", &s.d.yy);
    Printf("All records of student\n");
    Printf("Roll no = %d\t name = %s\t
          Course = %s\t fee = %f\n Date
    of birth %d - %d - %d", s.roll no, s.name,
    s.course, s.fee, s.d.dd, s.d.mm, s.d.yy);
      getch();
}
```

Page No
Date

(*) How to initialize structure variable ?

Syntax:-

```
struct student
{
    int roll no ;
    char name[20], course[10] ;
    float fee ;
} s = {10, "xyz", "BCA", 20000.00};
```

```
struct student s = {10, "xyz", "BCA",
                    20,000.00};
```

(*) How to access ?

→

```
printf(" Name = %s", s.name);
printf(" fee = %f", s.fee);
```

Q. How to access member of structure using dot operator with pointer structure variable ?

→ Syntax:- (*structure variable>.<member name>);

eg:-

```
struct student
{
    int roll no;
    char name[20], course[10];
    float fee;
};
struct student *s;
```

using dot operator

→ s → roll no = 10;

or, or

→ (*s).roll no = 10;

Page No.
Date

6

Structure with function

WAP in C to create a structure as named employee to input employee id, name, post & basic salary, after that calculate gross salary using function.

```
#include <stdio.h>
#include <conio.h>
struct employee
{
    int eid;
char ename[20], post[20];
 float bs;
};
struct employee e; // global structure var
float hra, da, ta, ma, gs;
 void input()
{
printf("\n Enter employee id, name, post & basic salary");
scanf("\n %d %s %s %f", &e.eid, ename, epost, &e.bs);
void findgross()
{
  if( e.bs >= 20000)
  {
    hra = (e.bs * 40) / 100;
    da = (e.bs * 90) / 100;
    ta = (e.bs * 30) / 100;
    ma = (e.bs * 10) / 100;
  }
else
```

```
{
    hra = (e.bs * 20) / 100;
    da = (e.bs * 50) / 100;
    ta = (e.bs * 20) / 100;
    ma = (e.bs * 8) / 100;
}
gs = e.bs + hra + da + ta + ma;
}
Void getsalaryslip()
{
printf("\n Details of Salary\n");
printf("\n Employee id = %d\n Name = %.s\n
    post = %.s\n, Basic Salary = %.f", e.eid,
    e.name, e.post, e.bs);
printf("\n HRA = %.f\n", hra);
printf("\n da = %.d\n ta = %.d\n ma = %.d",
                da, ta, ma);
}
Void main()
{
    clrscr();
    input();
    findgross();
getsalaryslip();
 getch();
}
```

Page No.
Date

Passing parameter as structure variable

W.A.P in C to input create a structure as named greatest to input three numbers after that find greatest numbers using function.

→

```
#include< stdio.h>
#include< conio.h>
 struct greatest
{
  int a,b,c;
};
 void findgreatest (struct greatest g)
{
printf("\n Enter three numbers");
scanf("%d%d%d", &g.a, &g.b, &g.c);
if( g.a> g.b && g.a>g.c)
  printf("\n Greatest no=%d ", g.a);
else if ( g.b > g.c)
  printf("\n Greatest no=%d", g.b);
else
  Printf("\n Greatest no=%d", g.c);
}
   void main()
  {
    clrscr();
   struct greatest g1;
   findgreatest (g1) ; // Calling function
    getch();
}
```

UNION

Page No.
Date 17 05 18

→ It is similar to structure but its only one member can be activated at a time.

→ it also provides user defined data type.

→ its main objective is to save memory with higher data type of member

→ it is generally used in game programming

→ it is created with the help of union keyword.

* Implementation of union :-

```
#include <stdio.h>
#include< conio.h>
#include< string.h>
union student
{
   int roll no;
  char name[20], course[10];
   float fee;
};
void main()
{
  clrscr();
union student S;
S.roll no = 100;
Printf("Roll no. = %d")
Strcpy(S.name, "xyz");
Printf("Name = %.s\n", S.name);
Strcpy("Course = %.s", S.course);
   S.fee = 20000;
```

```
printf("fee = %f", s.fee);
printf("The size of union = %d", sizeof(s));
                                                    1120
    getch();
}

Void main()
{
    clrscr();
    union student s;
    s.roll no = 100;
    strcpy(s.name, "xyz");
    strcpy(s.course, "BSc(IT)");
        s.fee = 20,000;
Printf("Roll no = %d\n Name = %s\n Course =
    %s\n fee = %f\n", s.roll no, s.name,
    s.course, s.fee);
    getch();
}
```

Storage class

→ It is used to specify scope, storage and lifetime of variable in a program, its main purpose is to increase the execution of program and save memory.

There are four storage class in c :-

→ Automatic / variable or auto
→ Static variable or static
→ register variable or register
→ external variable or extern

Page No.
Date

① automatic variable

→ it is declared with the help of 'auto' keyword.

→ it works as local variable.

→ it is default storage class

→ its location is RAM.

[All variable are available in RAM. exp→ Register variable]

Syntax:-

auto <data type><variable> [= value];

eg:- auto int a,b;
or
int a,b;

```
void main()
{
  clrscr();
  int a=10;
  {
    auto int b=20;
    {
      printf("a= %d\n", a); //a=10;
      int c=30;
      printf("c=%d\n", c);
    }
    printf("b = %d\n", b);
    printf("a = %b\n", a);
  }
    printf("a = %d\n", a); // printf("b=%d\n", b);
  getch();
}
```

Con't access

Static Variable :-

→ This type of variable is declared with the help of "static" keyword.

→ it works as local and global variable

→ it allocates memory only one times so, it is used to save memory space.

→ its value is preserved until end of program.

→ its memory location in RAM.

Page No.
Date 9 06 2018

⊛ character type function :—

→ These functions are used to evaluate character. Some character functions are available in C

ctype.h " header file

ⓐ isdigit ()
ⓑ isalnum ()
ⓒ islower ()
ⓓ isspace ()
ⓔ isupper ()
ⓕ tolower ()
ⓖ toupper ()
ⓗ isalpha ()

ⓐ isdigit() :→

Syntax:—

```
int isdigit (char value)
```

e.g:—

```
char ch = 'A';
if( isdigit (ch))
printf(" Digit ");
else
printf(" Not digit");
```

ⓑ isalnum ()

Syntax:—

```
int isalnum (char value);
```

e.g:—

```
if( isalnum ('5'));
Printf(" Alphanumeric ");
else
Printf(" Not alphanumeric ");
```

ⓒ isspace ()

e.g:—

```
if( isspace (' '))
printf(" space");
else
printf(" Not space");
```

(f) tolower() :- To convert upper char into lower character.

syntax:-

character to lower (char value)

eg:-

Printf("%c", tolower('A'));

output:- a

* WAP in c to input your name in lower case after that convert into upper case using "toupper" character function.

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
    Void main()
{
clrscr();
char ~~ch[20]~~ str[10];
int i;
printf("Enter any name");
for(i=0; str[i]!='\0'; i++)
{
Printf("%c", toupper(str[i]));
}
```

Write a program in c to input any string after that count vowel characters

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
char str[20], ch;
int i, v=0;
Printf("Enter any string"); scanf("%s", str);
for(i=0; str[i]!='\0'; i++)
{
ch = str[i];
if( ch=='A' || ch=='a' || ch=='E' || ch=='e' ||
ch=='I' || ch=='i' || ch=='O' || ch=='o' ||
ch=='U' || ch=='u')
v++;
}
Printf("Total No of vowel = %d", v);
getch();
}
```

~~ch=toupper~~
ch = toupper(str[i])

Mathematical function

(i) pow ()
(ii) sqrt ()
(iii) fmod ()
(iv) sin ()
(v) cos ()
(vi) tan ()
(vii) log ()
(viii) exp ()

math.h

(i) pow ()

eg:-

```
Printf("%f", pow(2,5));
```

o/p → $2^5 = 32$

(ii) sqrt ()

eg → `Printf("%f", sqrt(625));`

↓

25

(iii) fmod ()

eg:- `Printf("%f", fmod(10.5, 2.3));`

o/p → 1.3

(iv) Sin ()

e.g:- `Printf("%f", sin(90));`

→ `Printf("%f", sin(3.14/180*90));`

(vii) `Printf("%f", log(100));`

→ 4.65

(viii) `Printf("%f", exp(0));`

→ 1

`Printf("%f", log10(100)`

→ 2

→ C & C++ doesn't support any database feature like oracle, mysql, sql server.

Date 10/06/2018

File handling

(*) File handling :- It is technique in which file is used to store information permanently. After that we can handle different types of operation as read, write, append, search, update etc. the process of handling these operation in a file called the concept of file handling.

(*) Types of file :-
There are two types of file.

(i) Sequential file - We perform read and write in sequence order.

(ii) Random file / Binary file.

(i) Sequential file

Text file
myinformation.txt

```
I'm Jayesh I belong
to west champaran.jlm
studying IT from
College of commerce.
```

Data file
we can store record of any entity as student, book etc.
student.dat

001	abc	IT
002	xyz	B.Tech
003	MN0	MCA
004	PQR	MBA

(ii) Random file / Binary file :- We perform Read & Write operation randomly.

(a) Accessing information randomly

(b) The information of binary file isn't visible to users directly.

(c) The information of binary file is stored in binary format such as 0 or 1.

(d) The execution of binary file is fast. rather than sequential file.

(e) It is more secure rather than sequential file

Page No.
Date

* Operation on file :-
  (i) Read :- To Retrive information from a file
  (ii) write :- To save record in a file
  (iii) append :- To add info at the end of file.

* file mode :- It represents that which operation can be perform in a file. There are following file in C :-

	file mode	meaning
Sequential file	r	Read
	w	write
	a	append
	r+	read + write
	w+	write + Read
	a+	append + Read
Binary file	rb	writing in a binary file
	wb	Read
	ab	Append
	rb+	—
	wb+	—
	ab+	—

* STep for Creating a file in C.
  (i) We declare file pointer variable

```
FILE  =  *fp
↓        ↓
```

Pre-defined Data type that is available in <stdio.h> header file.

file pointer variable

(ii) open a file or an existing.

fopen () :- This function is used to open a file.

<file pointer> = fopen("file name", "file mode");

eg:- FILE *fp;

Case 1. fp = fopen ("abc.txt", "w");

Case 2.

fp = fopen ("xyz.txt", "r")

↳ if is specified file name is not available then file pointer returns NULL.

file opened in write mode

(iii) We perform operation as Read, write, append a/c to requirement. These operations can be perform with the help of the following methods.

(a)	fgetc ()
(b)	fputc ()
(c)	getc ()
(d)	putc ()
(e)	fprintf ()
(f)	fscanf ()
(g)	fread ()
(h)	fwrite ()

Case 3.

fp = fopen ("xyz.txt", "a");

Note:- (i) "w" mode overwrite existing data with new data.

(ii) "a" mode appends/adds info at end of file it does not overwrite previous data.

(iv) We closed an opened file

fclose () :- used to close a file.

Syntax:-

fclose (file pointer)

eg fclose (fp);

(i) fgetc ():- This function is used to retrive one character from a file.

Syntax:-

char fgetc (file pointer)

e.g:- char ch;
ch = fgetc (fp);

(ii) fputc :- This function is used to write a one character in a file.

syntax:-

~~char~~ fputc (character, file pointer)

eg:- char ch = 'A';
fputc (ch, fp);

(*) WAP in C to write information about your self in a file

```
Void main()
{
  clrscr();
  FILE *fp;
  char ch;
  fp = fopen ("myself.txt
```

(1) WAP in c to read information in file, after that count lower alphabet, upper alphabet and digit & special character.

```
void main()
{
clrscr();
FILE *fp;
char ch;
fp = fopen("myself.txt", "r");
while((ch = fgetc(fp)) != Eof)
{
Printf("%c", ch); // putch(ch);
}
fclose(fp);
getch();
}
```

(2) WAP in c to count no. of character, words and sentences in a file

(3) WAP in c to count only vowel character in a file

(4) WAP in c to convert the contents of file in upper case.

(5) WAP in c to copy the contents of one file into another file

(6) WAP in c to input different of characterset but write only Alphabet character in a file

(i) putc():— It is an output function which is used to write information in a file character wise. It is available in <stdio.h> header file.

Syntax:-

putc (<char var>, <file pointer>)

e.g.:-

```
ch = 'A';
putc (ch, fp);
```

(ii) getc() :— This function is used to read information from file characterwise.

Syntax:-

int getc (file pointer)

e.g.:-

```
char ch;
ch = getc (fp);
```

(iii) WAP in c to write information about yourself in ~~your~~ a file.

→

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
FILE *fp;
char ch;
fp = fopen("self.txt", "w");
printf("Enter information about yourself");
while((ch = getchar()) != '#')
{
putc(ch, fp);
~~getch();~~
}
fclose(fp);
getch();
}
```

Page No.
Date

④ WAP in c to read information from a file.

```
#include<stdio.h>
#include<conio.h>
    void main()
{
  clrscr();
  FILE *fp;
   char ch;
 fp = fopen("self.txt", "r");
    if(fp==NULL)
 printf("file is not available");
while(ch=getc(fp))!=EOF)
 {
   printf("%c", ch); // putch(ch);
 }
    fclose(fp);
    getch();
}
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
FILE *fp;
int l,u,d,s;
l=u=d=s=0;
char ch;
fp=fopen("myfile.txt","r");
if(fp==NULL)
printf("file is not available");
else
{
while((ch=getc(fp))!=EOF)
{
if(ch>=65 && ch<=90)
u++;
else if(ch>=97 && ch<=122)
l++;
else if(ch>=48 && ch<=57)
d++;
else
s++;
}
printf("Total upper alphabet=%d\n lower alphabet=%d\n
Total digit=%d\n Total special character=%d\n");
fclose(fp);
getch();
}
```

```
#include<stdio.h>
#include<conio.h>
    void main()
{
 clrscr();
 FILE *fp;
 char ch;
fp = fopen("myfile1.txt", "w");
while((ch = getchar() != EOF)
{
if( ch>=65 && ch<=90) || (ch>=97 && ch<=122)
    putc(ch,fp);
}
    fclose(fp);
    getch();
}
```

Q WAP in C to input any paragraph of text after that write text as upper case in a file.

→

```
#include<stdio.h>
#include<conio.h>
    void main()
{
 clrscr();
 FILE *fp;
 char ch;
fp = fopen("myfile2.txt", "w");
printf("Enter information");
while((ch= getchar() != EOF)
{
 if( ch>=97 && ch<=122)
     ch= ch-32;
 putc(ch,fp);
```

```
    fclose(fp);
    getch();
}
```

Q WAP in c to copy the contents of file one file into another file.

→

```
#include<stdio.h>
#include<conio.h>
    void main()
{
    clrscr();
  FILE *fp1, *fp2;
    char ch;
fp1 = fopen("source.txt", "r");
fp2 = fopen("targed.txt", "w");
 if(fp1 == NULL)
printf("Source file is not available");
else
{
    while((ch = getc(fp1) != EOF)
    {
      putc(ch, fp2);
    }
  printf("file is copied");
  fclose(fp1);
  fclose(fp2);
   getch();
}
```

Page No.
Date

(*) Data file

Q How to read/write formated data in file ?

→ (a) fprintf()
(b) fscanf()

(a) fprintf() :- This function is used to read, write formated data in a file. This function is available in <stdio.h> header file.

Syntax:-

fprintf (<file pointer>, <message type specifier>, <variable>);

e.g (1) FILE *fp;
char name[20] = "XYZ";
fp = fopen("student.dat", "w");
fprintf(fp, "Name = %s", name);

another eg

int rollno = 10;
char name[20] = "XYZ";
char Course[20] = "BCA";

fprintf (fp, "%d\t%s\t%s\t", rollno, name, Course);

Page No.
Date 15/09/2018

(b) fscanf :— It is an input function which is used to read formatted data from file. It is available in <stdio.h>

Syntax:- fscanf(file pointer, "Type specifier", "list of variable");

e.g:-

```
int rollno;
char name[20];
fscanf(fp, "%d%s", &rollno, name);
Printf("rollno = %d\n name = %s", rollno, name);
```

Q. WAP in C to input rollno, name, course and fee after that write these records in a file.