+ve → 0
-ve → 1

Date / /
Page No. 1

* Escape Sequence :-

1. \n → New line/next line.
2. \t → Tab (Horizontal)
3. \b → Back Space
4. \r → Carriage return

⇒ 1 Nibble = 4 bits
1 Byte = 8 bits
1 KB = 1024 B
1 MB → 1024

	Keyword Datatype	Size	
(1.) Integer	int (%d)	2B (16 bits)	-2^{15} to $2^{15}-1$ (-32760 to 32767)
(2.) Floating Point	float (%f)	4B (32 bits) 4 bit	-38 38 (3.4) to (3.4)
(3.) Character	char (%c)	1B	-2^{7} to $+2^{7}-1$

Date / /
Page No.
Shivalal

		ASCII Value
26	a-z	97-122
26	A-Z	65-90
10	0-9	48-57

Cyclic property. long long int →

* Operators:-

Date 29/08/18
Page No.
Shivalal

→ Operand:- Data values on which the operator perform a operation task.

→ Operator:- Some special symbols operands on data values

Types of Operators:-

1. Arithmetic Operators (+, -, *, /, %)
2. Relational Operator (<, <=, >, >=)
3. Equality " (==, !=)
4. Logical " (&&, ||, !)
5. Conditional " (Ternary Operator) ? :
6. Assignment " (=, +=, -=, *=, /= etc)
7. Bitwise " (&, |, ^, ~, <<, >>)
8. Unary " (Unary minus, Increment & decrement Operator)
9. Sizeof "
10. Comma Operator.

Priority High / [*, /, %] → Priority High; [+, -] → Low

① Arithmetic Operator (+, -, *, /, %) :-

S.No.	Operation	Operator	Example	Result
1.	Addition	+	5+3	8
2.	Subtraction	-	8-4	4
3.	Multiplication	*	3 * 5	15
4.	Division	/	13/4	3
5.	Modulus	%	12 % 7	5

Integer & char data values

5 + 3 * 2/6 = 6

Left to Right
↳ Associativity (Order of Evaluation)

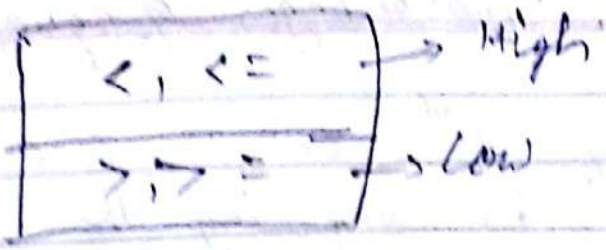
5.5 % 3 = Error.

② Relational Operators :-

S.No.	Operation	Operator	Example	Result
1.	Less than	<	4 < 7	1
2.	Less than or equal	<=	7 <= 8	1
3.	Greater than	>	7 > 11	0
4.	Greater than or equal to.	>=	7 >= 11	0

Scanned by CamScanner

Precedence.



(3) Equality Operator:-

(4) Logical AND (&&)

Truth Table:

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

2. Logical OR (||)

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

3). NOT (!)

A	!A
1	0
0	1

* Disadvantages of equality :-

(i) If 1st one is zero then second is anything the result is always zero.

```
#include<stdio.h>
void main()
{
int a=5, b=7, c=10, d;
d=(a=a+5) && (c=a+b+c);
printf("%d %d %d %d", a, b, c, d)
}
```

* If the first condition is false then not check second condition because it get false condition.

(3) Conditional Operator (Ternary Operator) :- ? :

Syntax :-

```
exp1 ? exp2 : exp3
```

Ex :-

```
int a=20, b=10, c
c = a>b ? a : b
    exp1  exp2 exp3
printf("%d", c)
```

(4) Bitwise Operators :-

1) Bitwise AND (&)

```
char a=5, b=7, c
c = a & b
printf("%d", c);
```

Result :- 5

5 :- 0101
7 :- 0111

0101

3) Bitwise EXOR (^)

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

Same terms = 0
Different terms = 1

Ex:- char a=8, b=13, c;
c = a^b

Ex:-
```
int a=5, b;
b = a<<2;
printf("%d", b);
```

Output: 20

(-)ve No. की 2 का compliment

Date 31/08/18
Page No

Shiva

(3) Bitwise :-

(*) Bitwise NOT (~) :-

Exc:- (i) int a = 5, b;
b = ~a;
printf ("%d", b);

Output :- b = -6

	64	32	16	8	4	2	1
0	0	0	0	0	1	0	1
1	1	1	1	1	0	1	0
1	0	0	0	0	1	0	1 (1 Compliment)
							+1
1	0	0	0	0	1	1	0

(2 Compliment)

Exc:- (ii) int a = -5, b;
b = ~a
printf ("%d", b);

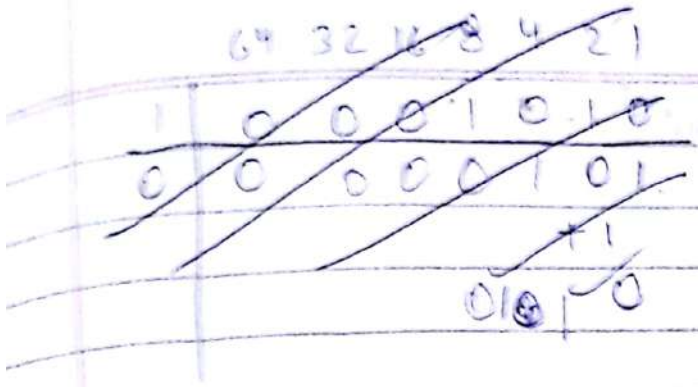
Output :- b = 4

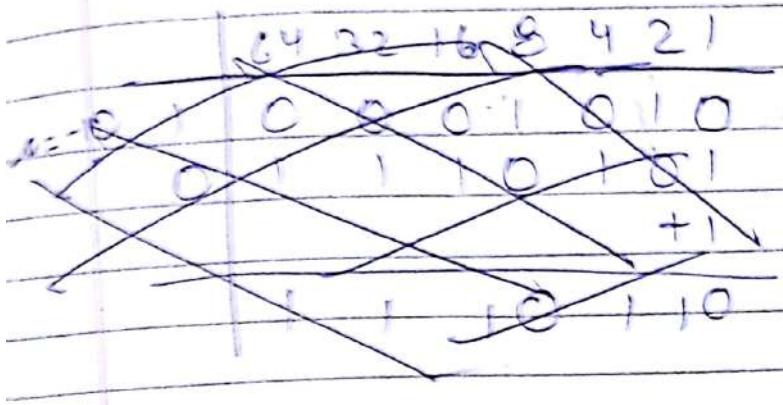
	64	32	16	8	4	2	1
1	1	1	1	1	0	1	1
0	0	0	0	0	1	0	0

Q.

int a = -10, b;
b = a << 2
printf ("%d", d)
}

[Run Code] -





⇒ Unary function :-

(1) Post Increament :- (p++)

```
int a=5, b;
b = a++;
printf("%d %d", a, b);
```

Output :- a = 6, b = 5

(2) Pre Increament :- (++p) :-

```
int a=5, b;
b = ++a
printf("%d %d", a, b)
```

Output :- a=6, b=6

(L to R) Post — Priority: p++ , p-- → ज्यादा

(R to L) Pre: ++p , --p → कम

③ Pre decrement (--p) :-

```
int a=5, b;
b = --a;
printf("%d %d", a, b);
```

Output :- a = 4 , b = 4

④ Post decrement (p--) :-

```
int a=5, b;
b = a--;
printf("%d%d", a, b)
```

Output :- a = 4 , b = 5

Q.

```
int a=5, c;
c = a++ + a++;
printf("%d%d", a, c)
```

For this case: c = 5 + 6 = 11 , a = 6++ = 7

* (depending upon compiler)

Q. Exception:

```
int a = 5, c;
c = ++a + ++a + ++a;
printf("%d%d", a, c);
```

(8, 7→8, 6 written under the three ++a terms)

All are same then before ++ increase value

a = 8, c = 22

keyword & also a Operator

Date / /
Page No.
Shivalal

(6) sizeof :-

evaluate size of datatype (Bytes)

Ex: (i) void main()
{
printf("%d %d %d", sizeof(char), sizeof(int), sizeof(float));
}

Output :- 1, 2, 4

(ii). void main()
{
printf("%d %d %d", sizeof('A'), sizeof(5), sizeof(5.));
}

Output :- 1, 2, 8

(7) Comma Assignment :-

Code :-

```
#include <stdio.h>
void main()
{
int a
a=(5,7)
}
```

Output :- 7

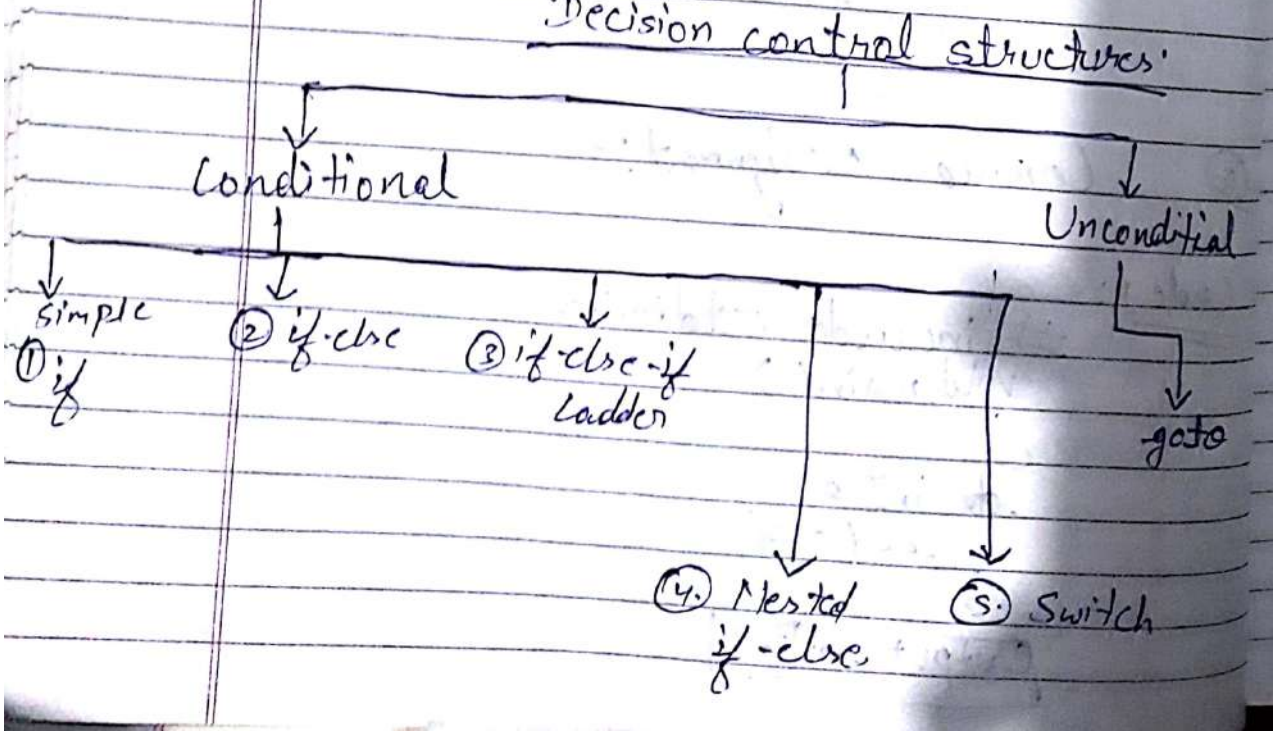
(7) Assignment Operator

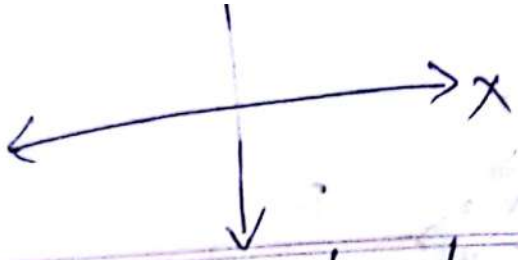
Ex:-
```
#include<stdio.h>
void main()
{
int a=b=5;
printf("%d",a);
```

Output a=5
b=5

→ a+=5 ⇒ a=a+5
→ a-=5 ⇒ a=a-5
→ a=a*5 ⇒ a*=5
→ a/=5 ⇒ a=a/5

* Decision Control structures :-





Date / Page No

Q. Input the co-ordinate of a pt. in x-y plane and check it which quadrant lies.

Sol^n

```
#include<stdio.h>
void main()
{
int x,y;
printf("Enter x and y value");
scanf("%d%d",&x,&y);
if(x>0 && y>0)
  printf("I st Quadrant");
else
  if(x<0 && y>0)
    printf("II nd Quadrant");
  else
    if(x<0 && y<0)
    printf("III rd Quadrant");
  else
    if(x>0 && y<0)
    printf("IV th Quadrant");
}
```

0 to 254 → are not used in Single.

* FLOAT.

IEEE Representation of floating point No.:-

Ex: $(42.73)_{10}$

64 32 16 8 4 2 1	
1 0 1 0 1 0	$0.73 \times 2 = 1.46 \rightarrow 1$
	$0.46 \times 2 = 0.92 \rightarrow 0$
	$0.92 \times 2 = 1.8 \rightarrow 1$
	$0.84 \times 2 = 1.68 \rightarrow 1$
	$0.68 \times 2 = 1.36 \rightarrow 1$

$\rightarrow (101010.10111)_2$

$\pm M \times B^{\pm E}$ → exponent

M → Mantissa

B → Scaling factor (Base)

$1.0101010111 \times 2^{+5}$ → E

M, B

- ve → 1

+ ve → 0

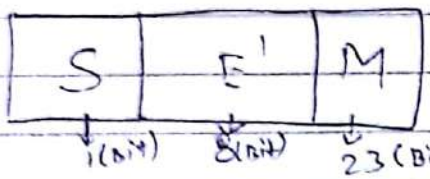
$E' = E + 127$

$E' = E + 127$

$E' = 132$

E' (Binary) = 10000100

1) Single Precision (32 bit)



2) Double Precision (64 bit)

$-126 < E < +127$

$1 < E < 254$

0	10000100	01010101111 (+3 no' 0's)
S	E'	M

short cut method

$(101010.10111)_2$ (54 3 2 1 0 -1 -2 -3 -4 -5)

$= 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times10^2 + 1\times10^1 + 0\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4} + 1\times2^{-5}$

$= 42.73$

Q. [redacted] -85.69

Represent in single presition form

$= [redacted] (85)_{10} \rightarrow (1010101)_2$

$0.69 \times 2 \rightarrow 1.38 \rightarrow 1$

$0.38 \times 2 \rightarrow 0.76 \rightarrow 0$

$0.76 \times 2 \rightarrow [redacted] \rightarrow 1$

$0.52 \times 2 \rightarrow 1.04 \rightarrow 1$

$\Rightarrow (-1010101.1011)_2$

$\Rightarrow \boxed{\pm M \times R^{\pm E}}$ → Normalized form

$(-1.01010111011 * 2^{+6})$

E = 6

E' = 6 + 127

E' = 133

$(133)_{10} = (10000101)_2$

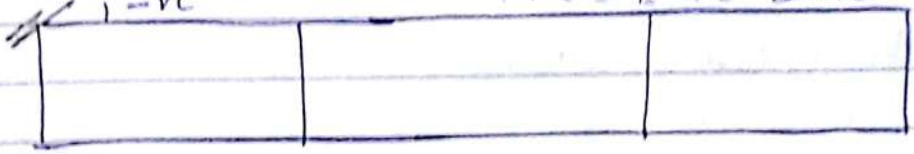
1	10000101	01010111011 13 no 0's

2. Double precision (64 bit) :-

$-1022 <= E <= +1023$

$+1 <= E' <= 2046$

0 +ve
1 -ve



$E' = E + 1023$

Ex: -85.69.

1	00010000101	01010111011 - - - 420's

Date / /
Page No.
Shivalal

Q. Write a C program to input electricity unit charges and calculate total electricity bill according to the given condition:

For first 50 units Rs. 0.50/unit.
For second 100 units Rs. 0.75/unit
" " " 100 " Rs. 1.20/unit
For unit above 250 units Rs. 1.5/unit
An additional surcharge of 20% is added to the bill.

Sol":

```
#include<stdio.h>
void main()
{
int t;
float c;
printf("Enter units");
scanf("%d", &t);
if(t<=50)
    c=t*0.5;
else
if(t>50 && t<=150)
  c=50*0.5+(t-50)*0.75;
else
 if(t>150 && t<=250)
    c=50*0.5+100*0.75+(t-150)*1.2;
else
 c=50*0.5+100*0.75+150*1.2
          +(t-250*1.5);
    c=c+c*20/100
printf("Total Bill=%f",c);
}
```

```
if (a >= '0' && a <= '9')
    printf (" Digit");
else
    printf ("Symbol");
}
```

Number System

* Octal (8) (0-7)

Hexadecimal ← int num1 = 0x a2b

int num = 070 → Octal

Hexadecimal (16) (0-15)

10 → a
11 → b
12 → c
13 → d
14 → e
15 → f

[%x, %X] → Size

Q.

```
int main()
{
int num1 = 0x a2;
printf("%d\t%o\t%x", num, num, num);
}
```

Output:

$(a2)_{16} \rightarrow (\ )_{10}$

$a * 16^1 + 2 \times 16^0 = 160 + 2$

$= (162)_{10}$

Output: 162 . 242 a2

Shortcut :-
Only for :-

$(457)_8 \longrightarrow (\quad)_2$

$2^3 \qquad 2^1$

$(100\ 101\ 111)_2$

4 2 1
1 1 1
1 0 1
1 0 0

$(45a)_{16} \longrightarrow (\quad)_2$

$2^4 \qquad 2^3$

$(0100\ 0101\ 1010)_2$

8 4 2 1
1 0 1 0
0 1 0 1
0 1 0 0

binary to

* Pair three to convert in Octal
* Pair of 4 to count binary to Octal

// atan() → $\tan^{-1}x$
asec() → $\sec^{-1}x$
Acos() → $\cos^{-1}x$

Date / /
Page No.
Shivalal

Q1: In a company there are different levels of employee, for example level1, level2, level3, level4 depending on the labels employees get convience and annual perks based on the following table :-

Employee level	Convience (in Rs.)	Perks (in Rs.)
level 1	1500	1000
level 2	1200	800
level 3	1000	400
level 4	500	200

You have to input level of employee, Basic Pay, HRA (in %), DA (in %) and then calculate his gross salary according to his level. Now based on the gross salary employee have to pay the income tax.

Salary	Tax (in %)
Upto 25000	5%
Upto 40000	7%
More 40000	10%

Calculate gross salary & tax that need to be paid.

Q2: Calculate the square of those no. only whose least significant digit 7.

Q.3: Calculate bill of a job workdone as follows?

Rate of typing	Rs. 3/page
Printing of first copy	Rs. 5/page
later every copy	Rs. 3/page

User should enter the no. of pages and printout given he/she want?

Q.4: Calculate the bill of total hours invested on a cyber cafe for surfing done based on the following condition:-

1. 1 Hour → Rs. 20
2. ½ Hour → Rs. 15
3. Unlimited Hour in day → Rs. 90.

Owner should enter the no. of hours spent by the customer. Calculate bill.

Soln:-

```
#include<stdio.h>
void main()
{
int level;
float bs, HRA, DA, gr;
printf("Enter level & basic pay");
scanf("%d %d", &level, &bs);
```

```
printf("Enter HRA & DA (in %)");
scanf("%f %f", &HRA, &DA);
gr = bs + (HRA * bs)/100 + (DA * bs)/100;
switch (level)
{
case 1: gr = gr + 1500 + 1000;
        break;
case 2: gr = gr + 1200 + 800;
        break;
case 3: gr = gr + 1000 + 400;
        break;
case 4: gr = gr + 500 + 200;
        break;
printf("The gross salary = %f", gr);
float it;
if (gr <= 25000);
        it = (gr * 5)/100;
else if (gr <= 40000);
        it = (gr * 7)/100;
else if (gr > 40000);
        it = (gr * 10)/100;
printf("Income Tax = %f", it);
}
}
```

Sol^n 2: (CPP)

```
#include <stdio.h>
void main()
{
int a, d, r;
printf("Enter no. = ");
scanf("%d", &a);
d = a % 10;
if (d == 7)
    r = a * a;
printf("Square = %d", r);
else
printf("This no. is not taken because "
        "that not have least count is 7");
}
```

Q-1

Sol^n 3:

```
#include <stdio.h>
void main()
{
int a, h, p, k;
printf("Enter no. of pages = ");   & Printout
scanf("%d %d", &a, &p);
h = a * 3;
printf("Typing cost = %d", h);
if (p == 1)
{
    k = a * 5;
printf("Print cost = %d", k);
}
else if (p > 0)
```

Date / /
Page No.
Shivalal

```
{
k = 1(a * 5) + (p-1) (a * 3);
printf(" Print cost = %d" k);
    }
}
```

m-II

```
#include<stdio.h>
    void main ()
{
int a, p, h, k;
    printf (" Enter no. of pages & Printout");
scanf (" %d %d", &a, &p);
k = 1(a * 5) + (p-1) (a*3);
printf (" Print cost = %d", h);
    h = a * 3;
printf (" Typing cost = %d", h);
    }
```

Sol"4":

```
#include<stdio.h>
    void main()
{
  int a;
  float k
printf(" Enter hours = ")
  scanf(" %d", &a);
if (a==1)
      printf (" Bill = 20");
else if (a==0.5)
      printf (" Bill = 15");
```

```
else if (a>0)
    k=(int) a*20+(a-(int a))*15
printf("Bill=%.f", k);
else if (a>=5)
        printf("Bill=90");
}
*/ {
HC=(int) H*20;
M=H-(int) H;
    if (H<0.5)
        HC=HC+15
else
        MC=MC+20    */
```

1. Start
2. set c=10
3. display "Name"
4. c=c-1
5. Repeat steps (3) & (4) will c>1
6. Stop

int c=5;
do
{ display("Name");
c=c-1;
} while(c>0);

Qn 1: Write a program to print the sum of first n natural no. (with all 3 loop structure).

Q.2: Write a programs to find the factorial of a no.

Q.3: Wre a program x^y without using power function.

Qn 4: Write a program to find the reverse of a no.

Q.5:- To check no. is pallendrome or not.

Q6:- To find the sum of digits of a no.

Q.7: To check no. is prime or not.

Q.8: Write a program to print the n terms of Fabnocci series.

Date / /
Page No.
Shivalal

To display the table of no. ?
To check no. is perfect no. or not.
Write a program to input a no. and count the even no., odd no & zeros.
Write a program to input an odd no & count the positive, negative & zeros.
To check the no. is amstrong or not.
To check no. is Robinson or not.

```
#include<stdio.h>
void main()
{
int n, S=0, i=1
printf("Enter value of N");
scanf("%d", &n);
while(i<=n)
{
  S=S+i;
  i=i+1;
}
printf("Sum of natural no.=%d", S);
}
```

```
#include<stdio.h>
void main()
{
int n, S=0, i;
printf("Enter value of N");
scanf("%d", &n);
for(i=1, i<=n; i++)
{
```

```
        s = s + i;
    }
    printf("Sum = %d", s);
}
```

```
#include <stdio.h>
void main()
{
    int n, s = 0, i = 1;
    printf("Enter value of n");
    scanf("%d", &n);
    do
    {
        s = s + i;
        i = i + 1;
    } while (i <= n);

    printf("Sum = %d", s);
}
```

Ex no 2:

```
#include <stdio.h>
void main()
{
    int n, f = 1, i = 1
    printf("Enter no.");
    scanf("%d", &n);
    do
    {
        f = f * i;
        i++;
    } while (i <= n);
    printf("Factorial = %d", f)
}
```

Q3.

```
#include <stdio.h>
void main()
{
    int x, y, p=1, i=1
printf("Enter value of x & y");
scanf("%d%d", &x, &y);
while (i<=y)
{
    p = p * x;
    i++;
}
printf("Power = %d", p);
}
```

OR

```
int x, y, p=1
while (y!=0)
{
    p = p * x;
    y--;
}
printf("Power = %d", p);
}
```

Q4.

```
#include<stdio.h>
void main()
{
    int a, s=0, rem;
    printf("Enter no.");
scanf("%d", &a); int a=a;
    while (a>0)
{
    r = a%10;
    s = s*10 + r;
    a = a/10;
}
printf("Reverse no. = %d", s);
```

```
if (s==n)
printf("Palin.")
else
printf("not")
}
```

Page No.

Sol^n C:-

```
#include<stdio.h>
void main()
{
int a, s=0, r
printf("Enter no.");
scanf("%d", &a);
while(a>0)
{ r= a%10;
  s=s+r;
  a= a/10
}
printf("Sum of digits = %d", s);
}
```

Q.15: Write a program to multiply a 2 no. without multiply.

(Hint:- like 3*5.
Add 3 5 times
c = 3+3+3+3+3 = 15)

Ans:
```
#include<stdio.h>
void main()
{
int x, y, p=0, i=1;
printf("Enter value of x & y (x*y)= ");
scanf("%d %d", &x, &y);
while(i<=y)
{
p=p+x;
i++;
}
printf("Multiply of 2 no.(%d*%d)=%d"
, x, y, p);
}
```

Q.16: Write a program a no. have a factor by loop.

Ans:
```
#include<stdio.h>
void main()
{
int x, i=1;
printf("Enter value of ");
scanf("%d", &x);
```

(8)

```
for(; i<x; i++)
{
    if(x%i==0)
    {
        printf("%d\n", i);
    }
}
```

Q-17:- Write a program to convert decimal to binary, display binary.

Q.18:- Input the age of is student & count how many of them are still a baby, attending School, & adult age.

Sol^n 17:-

```
#include<stdio.h>
void main()
{
int a, r, b, i=1;
printf("Enter no.");
scanf("%d", &a);
while(a>0)
{
 r = a%2
b = b + (r*i);
a = a/2;
 i = i*10;
}
printf("Binary = %d", b);
}
```

Date / /
Page No.
Shivalal

Q.16): a) Still a baby 0-5
b) School going 6-17
c) Adult 18 & More.

Q.17):
```
#include <stdio.h>
void main()
{
int b=0, s=0, a=0, n, j;
for(j=1; i<=15; j++)
{
printf("Enter age of student no. %d= ", j);
scanf("%d", &n);
if(n<=5)
b++;
else if (n<=17)
s++;
else
a++;
}
printf(" Total baby= %d\n Total school going= %d\n
Total adult= %d\n", b, s, a);
}
```

Q.18: Write a program find odd no. b/w 5 & 97 and then average of odd no.

```
#include <stdio.h>
void main()
{
int j, s=5, d=97, sum, c=0;
float avg;
```

```
for(i=s; i<=d; i++)
{
  if(i%2==1)
  {
  sum=sum+i;
  c=c+1;
  }
  avg=(float)sum/c;
}
printf("Average=%d", avg);
}
```

```
#include<stdio.h>
void main()
{
  int n, i=1, s=0;
  printf("Enter no.");
  scanf("%d", n);
  while(i<n-1)
  {
    if(n%i==0)
    {
      s=s+i;
    }
    i++
  }
  if(s==n)
    printf("Perfect no.");
  else
    printf("Not Perfect");
}
```

(4 in)

```
#include<stdio.h>
void main()
{
  int n, i=1, s=0;
  printf("Enter no.");
  scanf("%d", &n);
  while(i<n-1)
  {
    if(n%i==0)
      s=s+i
  }
  i++
}
if(s==1)
    printf("Prime");
else
  printf("No. is not prime");
}
```

OR

```
#include<stdio.h>
void main()
int n, c=0, i=1
scanf("%d", &n)
while(i<n/2
```

(8)

```
#include<stdio.h>
void main()
{
  int n, a=0, b=1, c
  printf("Enter no.");
  scanf("%d", &n);

  if(n==1);
      printf("%d", a);
  else if(n==2)
      printf("%d%d", a, b);
  else
  {
      printf("%d %d", a, b);

  while(i<=n-2)
  {
      c=a+b;
      printf("%d", c);
      i++;
      a=b;
      b=c;
  }
}
}
```

(9)

```
#include<stdio.h>
void main()
{
int a, i=1;      scanf("%d", &a);
while(i<=10)
{
  s=a*i;
  i++;
```

```
printf("%d * %d = %d", a, i, s);
    }
  }
```

```
#include<stdio.h>
void main()
{
printf("Enter no.");
  scanf("%d", &n);

int n, s=0, d=0, m, r
  while(n>0)
{
    d++;
    n/=10;
}
while(m>0)
{
  r=m%10
  s=s+pow(r,d);
    m=m/10;

  }
if(p==s)
  printf("Amnstrong");
else
  printf("Not Amnstrong");
}
```

14.

```
#include<stdio.h>
void main()
{
    int n, s=0, r, f, i, m, S
    printf("Enter no.");
    scanf("%d", &n);
        m=n
    while(n>0)
{
    r= n%10;
    for(i=1, f=1; i<=r; i++)
    {
        f=f*i;
    }
        s=s+f;
    n=n/10;
}
    if(s==m)
        printf("Robinson");
    else
        printf("Not Robinson");
```

Q. Write a program to check a no. is prime using a break keyword?

Sol^n.

```
#include<stdio.h>
void main()
{
int i=2, n
printf("Enter no.");
scanf("%d", &n);

while (i<=n/2)
{
if (n%i==0)
break;
i++;
}
if (i==n/2+i)
printf("No. is prime);
else
printf("No. is not prime");
}
```

* Break and Continue :-

↓

Switch case

&

loop (for, while, do while)

@

```
int i=10;
if(i<15)
{
    i=i+2;
    if(i<20)
    {
        break;
    }
    printf("%d", i);
}
```

Output :- Error (misplaced break);

#

```
for(int i=1; i<=10; i++)
{
    if(i==5)
    {
        break;
    }
    printf("%d\n", i);
}
```

i	Output
1	1
2	2
3	3
4	4
5	5

```
for(int i=1; i<=10; i++)
{
    if(i==5)
    {
        continue;
    }
    printf("%d\n", i);
}
```

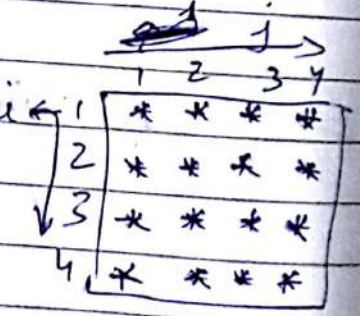
skip

i	output
1	1
2	2
3	3
4	4
5	6

Pattern Printing:-

Or

```
#include<stdio.h>
void main()
{
    int i, j;
    for(i=1; i<=4; i++)
    {
        for(j=1; j<=4; j++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```



Date / /
Page No.
Shivalal

Q

```
#include<stdio.h>
void main()
{
int i, j
for(i=1; i<=4; i++)
{
    for(j=1; j<=4; j++)
    {
        printf("%d", j);
    }
    printf("\n");
}
}
```

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

Q

```
#include<stdio.h>
void main()
{
int i, j;
for(i=4; i>=1; i--)
{
    for(j=4; j>=i; j--)
    {
        printf("%d", i);
    }
    printf("\n");
}
}
```

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

Q

```
#include<stdio.h>
void main()
{
int i, n
char j;
for(i=1; i<=4; i++)
{
    for(j='A'; j<=4; j++)
    {
        printf("%c", j);
    }
printf("\n");
}
}
```

```
A B C D
A B C D
A B C D
A B C D
```

Q.

```
#include<stdio.h>
void main()
{
    int i, j, k=1;
for(i=1; i<=4; i++)
{
    for(j=1; j<=4; j++)
    {
        printf("%d", k);
        k++;
    }
printf("\n");
}
}
```

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

```
#include <stdio.h>
void main()
{
  int i, j;
for(i=1; i<=4; i++);
  {
    for(j=1; j<=i; j++)
    {
      printf("*");
      printf("\n");
    }
  }
```

	1	2	3	4
1	*			
2	*	*		
3	*	*	*	
4	*	*	*	*

Q.

```
#include <stdio.h>
void main()
{
int i, j;
for (i=1; i<=4; i++)
{
  for(j=1; j<=i; j++)
  {
    printf("%d", j);
  }
  printf("\n");
}
```

	1	2	3	4
1	1			
2	1	2		
3	1	2	3	
4	1	2	3	4

Q.

```
#include<stdio.h>
void main()
{
    int i, j;
for( i=4; i>=1; i--)
{
  for(j=4; j<=i; j--)
  {
    printf("%d", i);
  }
  printf("\n");
}
}
```

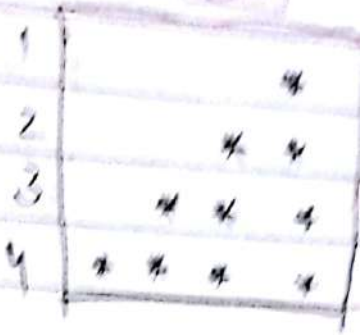
```
4
3 3
2 2 2
1 1 1 1
```

Q.

```
#include<stdio.h>
void main()
{
int i, j
for(i=1; i<=4; i++)
{
    a=i
for(j=1; j<=i; j++)
{
  printf("%d", a);
  a= a+4-j
}
printf("\n");
}
}
```

```
1
2 5
3 6 8
4 7 9 10
```

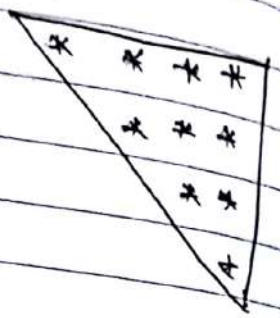
```
#include <stdio.h>
void main()
{
  int i, j, k

  for(i=1; i<=4; i++)
  {
    for(j=4; j>i; j--)
    {
      for(k=i; k<=i; k++)
      {
        printf("*");
      }
      printf("\n");
    }
  }
```



```
#include<stdio.h>
  void main()
{
  int i, j;
  for(i=4; i>=1; i--)
    {
      for(j=1; j<=1; j++)
        {
```

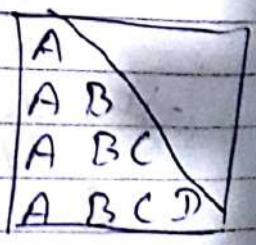
```
        printf("*");
    }
  }
}
```

Q.



```
#include<stdio.h>
void main()
{
  int i,j,k;
  for(i=4; i>=1; i--)
  {
    for(j=4; j>i; j--)
    {
      for(k=4; k<=1; k++)
      {
        printf("*");
      }
      printf("\n");
    }
  }
}
```

Q.



```
#include<stdio.h>
void main()
{
  int i,j;
  for(i=1; i<=4; i++)
  {
    for(j=1; j<=i; j++)
```

```
        {
            printf("%c", 64+j);
        }
        printf("\n");
    }
```

(OR)

```
#include<stdio.h>
void main()
{
char i, j
    for(i='A'; i<='D'; i++)
    {
        for(j='A'; j<=i; j++)
        {
            printf("%c", j);
        }
        printf("\n");
    }
```

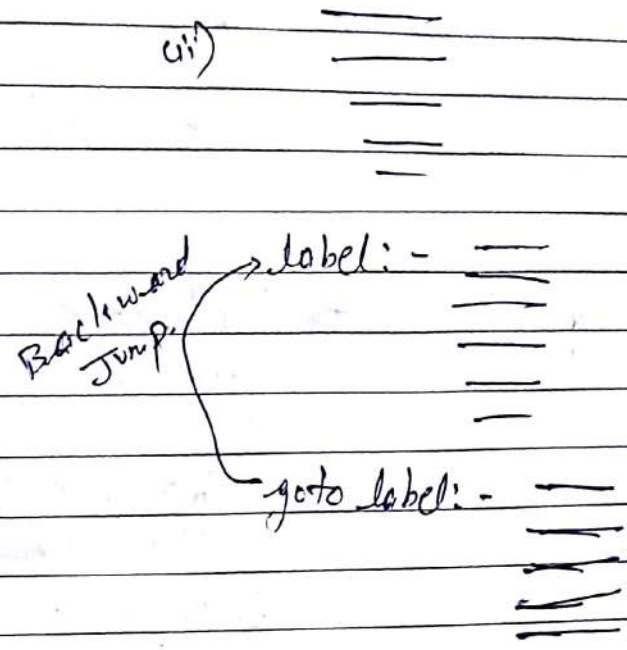
Date / /
Page No.
Shivalal

goto statement :-

→ Transfer the control

(i)

(ii)



Q. Write a program to input the numbers and find the sum of only positive no. and stops the proceds. if 999 encountes. by user.

sol":-

```
#include<stdio.h>
void main()
{
    int num, s=0;
    printf("Enter no. = ");
Read: scanf("%d", &num);
    if(num != 999)
    {
        if(num<0)
```

```
        goto Read;
        s = s + num;
        goto read;
    }
    printf("sum = %d", s);
}
```

(pr-2)

```
#include<stdio.h>
void main()
{
    int num, s=0;
    scanf("%d", &num);
    if(num != 999)
    {
        if(num>0)
        s = s + num;
        goto Read
    }
    printf("sum = %d", s);
}
```

(pr-3)

```
#include <stdio.h>
void main()
{
    int num, s=0;
    printf("Enter no. = ");
    scanf("%d", &num);

    for(; num != 999; ).
    {
        if(num>0)
```

```
    s = s + num;
}
scanf("%d", &num);
```

Array: → No Garbage

(c) int a[5];

	0	1	2	3	4
	1	2	3	4	5
	100	104	108	112	116

(a) printf("%d", a[0]);

Output:- 1

(b) printf("%d", a[5]);

Output:- Garbage

* (a) int a[5] = {1, 2,} ✓

(b) int a[5] = {5, 7, 2, 9, 1} ✓

(c) int a[] = {5, 2,} ✓

(d) int a[5] = {1, 2, 3, 4, 5, 6} ✗

* Determine how much have element :-

Address of a[index] = BA + (index - LB)

Lower Bound

Eg:

-1	-2	-3	-4	-5	0	1	2
2	7	5	9	3	1	6	4
100	104	108	112	116	120	124	128

Address of a[0] = 100 + (0 - (-5))

= 105

Q.1: Write a program to input the elements of an array and display them.

```
#include <stdio.h>
int main()
{
int a[100], n, i;
printf("Enter the size (No. of element)");
scanf("%d", &n);
printf("Enter array elements");
for(i=0; i<=n-1; i++)
{
scanf("%d", &a[i]);
}
printf("Elements are");
for(i=0; i<n-1; i++)
{
printf("%d", a[i]);
}
}
```

Q.2: Write the program to input, calculate the sum and average of all elements.

Q.3: ~~Write to input~~ Reverse of Array.

```
#include<stdio.h>
int main()
printf("Enter the size (No. of element)");
scanf("%d", &n);
printf("Enter array elements");
```

```
for(i=0; i<=n-1; i++)
{
    scanf
}
printf("Reverse order")
for(i=n-1; i>=0; i--)
{
    printf("%d", a[i]);
}
```

Q no 2:

```
#include <stdio.h>
int main()
{
    int a[100], n, i, s=0, avg;
    printf("Enter the size (No of element)");
    scanf("%d", &n);
    printf("Enter array elements");
    for(i=0; i<=n-1; i++)
    {
        scanf("%d", &a[i]);
        s = s+a[i];
    }
    avg = (float) s/n;
    printf("Sum= %d, Average= %d", s, avg);
}
```

WAP -> Write a program

Date / /
Page No.
Shivalal

WAP to input the no. & display max. b/w them.

```
#include <stdio.h>
int main()
{
int a[100], n, i, max;
printf("Enter the size (*No. of element)");
scanf("%d", &n);
printf("Enter array elements");
for(i=0; i<=n-1; i++)
{
scanf("%d", &a[i]);
max = a[0];
if (a[i] > max)
{
max = a[i];
printf("No. is max. = %d", max);
}
}
}
```

Q.4:- WAP to input array and count the (+)ve, (-)ve & Zero's. store

Q.5: WAP to input array and then display Even & Odd in different array & display different them.

Ed"4:-

```
#include<stdio.h>
int main()
{
int a[100], n, i, b=0, c=0, s=0;
```

```
printf(" Enter the size (No: of element)");
scanf("%d", &n);
printf(" Enter array element:");
for(i=0; i<n-1; i++)
{
    scanf("%d", &a[i]);
if (a[i]>0)
{
  b++;
}
else if (a[i]<0)
{
   c++;
}
else
   s++;
}
printf(" Total positive: %d.
        Total negative : %d
        Total zeros = %d", b, c, s);
}
```

Q. Write a program to stored the odd and even no. in two separate array and display them.

```
#include<stdio.h>
void main()
{
int a[100], e[100], o[00], n, i, k, j
```

```
for(i=i-1; i>=0; i--)
printf("%d", a[i]);
a[j] = n%2
```

```
printf("Enter the size (No. of element)");
scanf("%d", &n);
printf("Enter array elements");

for(i=0; i<=n-1; i++)
{
    scanf("%d", &a[i]);
    if(a[i]%2==0)
    {
        e[j] = a[i];
        j++;
    }
    else
    {
        a[k] = a[i];
        k++
    }
}
for(i=0; i<=j-1; i++)
{
    printf("%d", e[i]);
}
for(i=0; i<=k-1; i++)
{
    printf("%d", a[i]);
}
```

Q. WAP, to input array in a[i] print in reverse order of b[i].

```
#include
    void
{
    printf("
    scanf("
    printf("
    scanf("
for (i=n-1;  j=0; i>=0;  i--; j++)
    {
        b[j] = a[i]
    }

printf("Elements are");
    for(
    {
    printf("%d", b[j]);
    }
```

Date / /
Page No.
Shivalal

Operations on Arrays :-

1) Searching — Linear, Binary, Hashing
2) Sorting — Bubble, Insertion, Merge, Heap
3) Insertion
4) Deletion

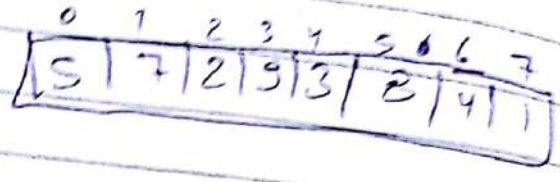
Searching :-

Q. W.A.P. to search a no.

```
#include<stdio.h>
void main()
{
  int n; a[i]; i, ele
  printf("Enter the size (No. of element)");
  scanf("%d", &n);
  printf("Enter elements");
  for(i=0; i<n-1; i++)
  {
    scanf("%d", &a[i]);
  printf("Element need to be searched");
  scanf("%d", &ele);
  for(i=0; i<n-1; i++)
  { if(a[i]==ele)
    {
      printf("%d formed at %d position", ele, i++);
      break;
    }
  }
}
```

if (i == n)
{
printf("Element not found");

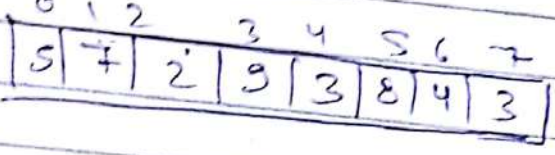
Ex:-



i) element = 3
Output: 3 found at 5.

ii) element = 11
Output:- Element not found.

Q. Ex:-




```
#include<stdio.h>
void main()
{
int a[i], n, i, de, f
printf("Enter the size (No. of elements)");
scanf("%d", &n);
printf("Enter elements");
for(i=0; i<=n-1; i++)
{
scanf("%d", &a[i]);
if(a[i]==de)
{
```

printf("%d found at %d position")

f=1;

if (f==0)

printf("Element not found");

Sorting:-

To arrange the data either in ascending or descending or decking order.

Ex:- | 5 | 7 | 1 | 4 | 3 | 9 | 2 | → convert Ascending → | 1 | 2 | 3 | 4 | 5 | 7 | 9 |

```
#include<stdio.h>
void main()
{
    int a[100], n, i
printf("Enter the size (No: of elements)");
  scanf("%d", &n);
printf("Enter elements");

for(i=0; i<=n-1; i++)
{
    scanf("%d", &a[i]);
```

```
{ Array
for ( i=0; i<=n-2; i++)
{
    for(j=0; j<n-i-2; j++)
    if( a[j] > a[j+1])
    {
        temp = a[j];
        a[j] = a[j+1];
        a[j+1] = temp;
    }
}
printf (" Sorted Array (Ascending Order)");
for (i=0; i<n-1; i++);
{
    printf (" %d", a[i]);
}
```

Hint:-

5	7	1	4	3	9	2
0	1	2	3	4	5	6

Pass 1:

5 7 1 4 3 9 2

5 7 1 4 3 9 2

5 1 7 4 3 9 2

5 1 4 7 3 9 2

5 1 4 3 7 9 2

5 1 4 3 7 2 9

Pass 2:

5 1 4 3 7 2 9

1 5 4 3 7 2 9

1 4 5 3 7 2 9

1 4 3 5 7 2 9

Date / /
Page No.
Shivalal

1 4 3 5 2 7 9

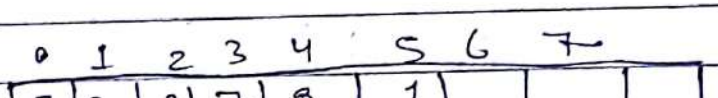
Pass 3:
1 4 3 5 2 7 9
1 3 4 5 2 7 9
1 3 4 2 5 7 9.

Pass 4:-
1 3 4 2 5 7 9
1 3 4 2 5 7 9
1 3 2 4 5 7 9

Pass 3:
1 3 2 4 5 7 9
1 2 3 4 5 7 9

Insertion:-
Means Insert a no.

Initial:-



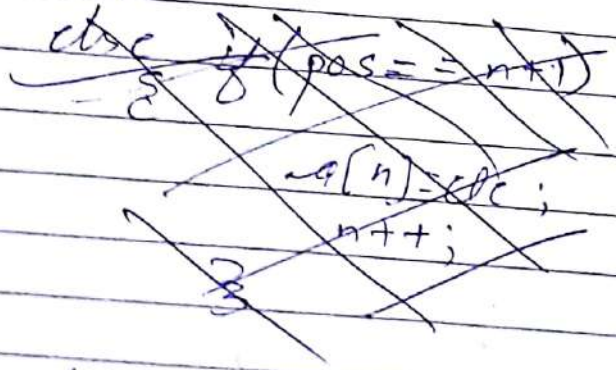
n=6
clc=4
pos=3

1) 5 2 3 7 9 1 1
2) 5 2 3 7 9 9 1
3) 5 2 3 7 7 9 1
pos=2 4) 5 2 3 3 7 9 1

5 2 4 3 7 9 1

```
#include<stdio.h>
void main()
{
int a[100], n, clc, pos
```

```
printf(" Enter the size(no. of elements):");
scanf(" %d", &n);
printf(" Enter elements");

for (i=0; i<=n-1 ; i++)
{
    scanf(" %d", &a[i]); }

scanf("%d", &pos);
printf("Enter ;insert element ");
scanf(" %d", &ele) ;

if( pos<0 || pos >= n+2)
{
printf(" Can't insert (wrong pos
```



```
else
{
    pos = pos-1;
for (i=n; i>pos ; i--)
{
    a[i] = a[i-1];
}
    a[pos] = ele;
    n++;
}
```

for(i=0; i<n; i++)
{
printf("%d\t", a[i]);
}

Deletion of an element from an array:-

0	1	2	3	4	5	6
5	2	5	7	3	1	4

n=7

Pass 4

Pass=4-1=3

1) 5 2 5 3 3 1 4

2) 5 2 5 3 1 1 4

3) 5 2 5 3 1 4 4

n=6

code:

```
#include<stdio.h>
void main()
{
    int i,pos,n,a[100];
    printf("
    scanf("
    printf("
    for(i=0;i<n-1;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("Enter element of that position");
    scanf("%d",&pos);
```

```
if(pos<0 || pos>=n+1)
{
    printf("Invalid pos (can't delete)");
}
else
{
    pos = pos-1;
    for(i= pos; i<n-1; i++);
    {
        a[i] = a[i+1];
        n = n-1;
    }
    printf("Array of a deletion \n");
    for(i=0; i<n-1; i++)
    {
        printf("%d\t", a[i]);
    }
}
```

Q. If position is not given, we take a element and then searching & delete that element. WAP.

Sol^n.

```
#include<stdio.h>
void main()
{
    int a[100], n, de, pos.
    printf("
    scanf("
    printf("
```

```
for (i=0; i<n-1; i++)
{
scanf("%d", &a[i]);
}
    printf("Enter element");
    scanf("%d", &de);
for (i=0; i<n-1; i++)
{ if (a[i]==de)
    {
      pos=i;
      break;
    }
if (i==n)
{
  printf("Can't delete (not delete)");
}
else
{
  for(i=pos; i<n-1; i++)
  {
    a[i] = a[i+1];
  }
  n=n-1;
  printf("Array of a deletion \n");
  for (i=0; i<n-1; i++)
  {
    printf("%d\t", a[i]);
  }
```

Date 12/10/18
Page No

* 2-D Arrays :-

(Array of Arrays)

→ To arrange data in form of tables/matrix

Declaration :-

datatype Arrayname [Row size] [Col. size]

- Row size → No. of Array
- Col. size → size of every Array.

Ex: ① int marks [3] [5];

TM = RS * CS * sizeof(dt)

3 * 5 * 4

60 Byte.

		0	1	2	3	4
No. of Rows	0	00	01	02	03	04
		100	104	108	112	116
	1	10	11	12	13	014
		120	124	128	132	136
	2	20 140	21 144	22 148	23 152	24 156

→ No. of Col. (* size of Every)

Ex: ② float B[20] [50]

⇒ 20 x 50 x 4

⇒ 4000 Byte.

Date / /
Page No.
Shivalal

Initialization:—

Compile time Initialization:—

✓ (i) int a[2][3] = {1,2,3,4,5,6};

✓ (ii) int a[2][3] = {{1,2,3},{4,5,6}};

✓ (iii) int a[][3] = {1,2,3,4,5,6};

✓ (iv) int a[][3] = {1,2,3,4,5,6,7};

✓ (v) int a[2][3] = {1,2,3,4};

1	2	3
4	0	0

✓ (vi) int a[2][3] = {{1},{2,3}};

1	0	0
2	3	0

✗ (vii) int a[2][3] = {{1,2,3,4},{5,6,7}};

:→ (Error) Invaid: too many initializer.

✗ (viii) int a[2][] = {1,2,3,4,5,6};

Error (Compile Time)

✗ (ix) int a[2][] = {{1,2,3},{4,5,6}}

✓ (x) int a[2][3] = {0};

✓ (xi) int a[2][3] = {1}; (OR)

(xii) int a[][] = {1, 2, 3, 4}

2) Run Time Initialization :-

Ex:- (4). int a[3][4]

a	0	1	2	3
0	100	104	108	112
1	116	120	124	128
2	132	136	140	144

Row :- for(i=0; i<=2; i++)
{

Column :- for(j=0; j<=3; j++)
{
scanf("%d", &a[i][j]);
}
}

Print (2D) Array :-

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

Address Calculation in 2-D Array :-

1.) Row Major Implementation
2.) Column Major Implementation

Row Major Implementation:-

Ex: int a[3][4];

LBR → 0	1 (100)	2 (104)	3 (108)	4 (112)
1	5 (116)	6 (120)	7 (124)	8 (128)
UBR 2	9 (132)	10 (136)	11 (140)	12 (144)

LBC = 0, UBC = 3

100	104	108	112	116	120	124	128	132	136	140	144
1	2	3	4	5	6	7	8	9	10	11	12
Row 0				Row 1				Row 2			

Add. of a[1][2] = [(1-0)*4+(2-0)] * 4 + 100

(i = 1, j = 2)

Add. of a[i][j] = BA + size of dte(i) [(i - LBR) * N + (j - LBC)]

- BA: Base Address
- size of dte(i): Word Size
- (i - LBR): Row Index
- N: Column Index
- (j - LBC): Lower Bound Column

* int a[ ][ ]

M N

N = UBC - LBC + 1

M = UBR - LBR + 1

Ex: int a[-1:1, -1:2]

LBR LBR

Q. WAP to input 2-D Array & print it in the form matrix.

```
#include<stdio.h>
void main()
{
  int a[100][100], m, n, i, j;
  printf("Enter the row & column size");
  scanf("%d%d", &m, &n);
  for(i=0; i<m-1; i++)
  {
    for(j=0; j<=n-1; j++)
    {
      scanf("%d", &a[i][j]);
    }
  }
  printf("Array elements:\n");
  for(i=0; i<=m-1; i++)
  {
    for(j=0; j<n-1; j++)
    {
      printf("%d\t", a[i][j]);
    }
    printf("\n");
  }
}
```

NULL → ('\0')

Strings

→ Collection of character terminated by NULL('\0') character

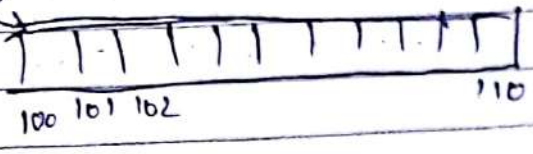
Ex: "C string class"

C		S	t	r	i	n	g		C	l	a	s	s	\0

- **Declaration:-**

  char stringname [size]

Ex:: char string [50]



- **Initialization:-**

1) **Compile time Initialization:-**

char str[6] = "Hello";

str →	0	1	2	3	4	5
	H	e	l	l	0	\0
	100	101	102	103	104	105

2) char str[] = "Hello world";

str →	0	1	2	3	4	5	6	7	8	9	10	11
	H	e	l	l	0		w	0	r	l	d	\0

3) char str[6] = {'H', 'e', 'l', 'l', '0', '\0'}; ✓

4) char str[6] = {'H', 'e', 'l', 'l', '0'} ✗

(No & is used in string)

Date / /
Page No.
(Shivalal)

c) char [ ] = {'H','e','l','l','o','\0'} ✓

d) char str[ ] = {'H','e','l','l','o'} ✗

e) char str[5] = "Hello";

compiler

gcc :→ Hello

DevC :→ Hell.f → Garbage Value (Warning)

Old compiler :→ Error

2) Run time Initialization :-

↓

Input & Output functions for string :-

(1) scanf :-

%s → string

Eg :→

```
char str[50];
scanf("%s", str);
printf("%s", str);
```

* Drawback of scanf()

Stores only single word.

(2) gets() :-

- Declaration :- gets(stringname);

gets(str);

(4) puts :-

Initial :- puts (string name);

puts(str); → Hello.world → Cursor Newline में आता है यदि printf में Cursor Sentence के बाद)

Ans: sconf ("%[^\n]s", str)

जबतक User Enter press नहीं करेगा, तबतक Value store करेगा)

Eg: scanf("%[^a]s", str);
↳ a के बाद Value not store.

str

| H | e | l | l | a | w | o | r | l | d | \0 | | |

Output :- Hell.

Q) WAP to input a string and display it.

Sol:
```
#include <stdio.h>
#include <string.h>
void main
{
    int str[50];
    printf("Input a String");
    gets(str); //scanf("%[^\n]s", str);
```

Date / /
Page No.

//scanf(" %[^\n]", str);

printf(" %[", str);

}

* Operations on Strings :-

1) Length Calculation
2) Copy one string to another.
3) Concatenation of one string into another.
4) Reverse of a Strings
5) Comparison of 2 strings

1) Length Calculation :- [strlen()]

n-1

```
#include <string.h>
main()
{
    int len;
    char str[100];
    gets(str);
    len = strlen(str);
    printf(" Length = %d ", len);
}
```

// Length :- Exclude Null

Date / /
Page No
Shivalal

```
#include <stdio.h>
#include <string.h>
void main()
{
    int  char str[100], i;
printf(" Enter \n.");
    char str[100];
    gets(str);

for (i=0; str[i] != '\0'; i++)
  {
    printf("%d", i);
  }
```

2) Copy one string to another.

```
#include <stdio.h>
#include <string.h>
    void main()
{
    int char str1[100], str2[100]
        gets (str1);

strcpy (str2, str1);
printf(" String after copy = %s", str2);
}
```

→ strcpy (destination string, Source String)

M-2

```
#include<stdio.h>
void main()
{
    char str1[100]
    gets(str1);

for(i=0; str[i]!='\0'; i++)
{
    str2[i] = str1[j];
}
for(i=0; str2[i], i++);
{printf("Copy = %s", str2[i]
}
}
```

3) Concatenation :-

Initialization :-

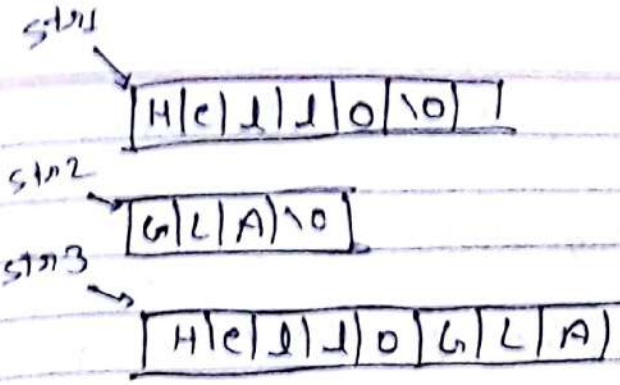
strcat

Syntax :- strcat(Output concatenate, string

string

M-3

```
#include<stdio.h>
#include<string.h>
void main()
{
    char str1[100], str2[100], str3[100
    gets(str2);
    gets(str1);
    strcpy(str3, str1)
    strcat(str2, str2)
printf("Second string after concatenate = %s", str3);
```




```
#include <stdio.h>
void main()
{
char str1[100], str2[100],

        gets (str2);
        gets (str1);

  strcat(str1, str2),
}
printf("Second string after concatate=%s", str1);
}
```

```
#include<stdio.h>
void main()
{
char str[100] , i, j;

        gets(str2);
        gets(str1);

for( i=0; str1[i] != '\0'; i++)
{
  for (j=0; str2[j] != '\0', j++)
{
```

```
    str1[i+j] = str2[j];
}
    str[i+j]='\0';
for(i=0; str[i]!='\0'; i++)
{
printf("%c",str[i]);
}
}
```

4) Reverse of Strings :-

↳ [strrev( )]

Syntax :- strrev (syntax-name)

A-I

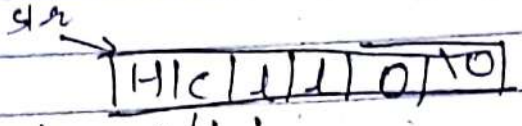
```
#include<stdio.h>
#include<string.h>
void main()
{
    char str[100];
    gets(str);
    strrev(str);
printf("String Reverse=%s", str);
```

M-II

```
#include<stdio.h>
#include<string.h>
void main()
{
```

Date / /
Page No
(Shivalal)

```
char str[100]
gets(n);
gets(str1);

for(i=n-1; str[i]!='\0'; i--)
{
    str[i] = str[i].
    printf("%s", str[i]);
}
}
```



(n-3)

```
#include<stdio.h>
#include<string.h>
void main()
{ int str[i], i, j;

for(i=0; str[i]!='\0'; i++){
    i=i-1;
    j=0;
}
for(; i>j; i--, j++)
{
    temp = str[i];
    str[i] = str[j];
    str[j] = temp;

printf("%s", str[j]);
}
```

(char में Enter का character आ न जाए इसलिए
Use "%*c")

Q. WAP to input a character & print.

```
#include<stdio.h>
void main()
{
    char ch[100], c;
    int i=0
    printf("Enter character");
    while(1)
    {
        scanf("%c %*c", &c);
        if(c!='\n');
        {
            ch[i]=c;
            i++;
        }
        else
            break;
    }
    ch[i]='\0';
    printf("%s\n", ch);
}
```

Comparison of 2 strings:-

```
#include<stdio.h>
void main()
{
    char str1[100], str2[100]

    for(i=0; str1[i] != '\0' &&
        str1[i] == str2[i]; i++)
    {
    }
```

strcmp()
↓
strcmp(string1, string2).
↓0 → Non zero
if same if diff.

```
if ( str1[i] > str2(i))
    printf(" string 1 > string 2);
else
    if ( str2[i] > str1[i])
        printf ("string 2 > string 1");
    else printf (" Identical");
}
```

trg

M-II)

```
#include<stdio.h>
#include<string.h>
void main()
{
    char str1[100], str2[100];
    gets(str1);
    gets(str2);
    if (strcmp(str1, str2)==0)
        printf (" Identical)
    else printf (" Not Identical")
}
```

Q. WAP to input a string & count total no of vowels, consonants & special character.

```
#include<stdio.h>
    void main()
{   char str[100]
    int v=0, c=0, d=0, s=0;
for( i=0; str[i]!='\0'; i++)
{
    if (str[i]='A'||'a'||'e'||'i'||'o'||'u')
                v++;
else if( str[i]>='a' &&
                || (str[i]>='A'||str[i]<=
            c++;
else if(str[i]>='0' && str[i]<='9');
            d++;
    else
            s++;

    printf("%d %d %d", v, d, s);
```

Q. WAP to input string to convert Uppercase to lowercase & lowercase to Uppercase.

```
#include<stdio.h>
    void main()
{
    char str[100].
    int i
```

```
for(i=0; str[i]='\0' ; i++)
{
    if(str[i]>='A' && str[i]<'Z')
        str[i] = str[i]+32;
    else if (str[i]>'a' && str[i]<='z')
        str[i] = str[i]-32;
}
```

Q. WAP to input a string & remove the extra spaces?

Soln:

```
#include<stdio.h>
    void main()
{
    char
     int

for(i=0; str[i]!='\0' ; i++)
{
    if (str[i]==" " && str[i+1]==" ")
{
for(j=i+1; str[j]='\0' ; j++)
        str[j]= str[j+1]
    i--;

printf("%s", str[j]);
}
```

Q. WAP to input a string & check the string is pallindrome or not.

sol":

```
#include <stdio.h>
#include <string.h>
void main()
{
    int
    char

for(i=0; m[i] != '\0'; i++)
{
    }
        while(i>j)
    {
            i = i - 1;
        if(str[i] = str == str[j])
            j=0
            i-- || i = 3;
            j++ || j=j+1;
        else
            break;
    }
    if(i==j || j==(i+1))
        printf(" Pallindrome");
    else
        printf("\nNot Pallindrome");
```

WAP to input a string & no. of words.

```
#include<stdio.h>
void main()
{
   int c=0, flag=1
char str[100]

for(i=0; str[i]!='\0'; i++)
{
  if(str[i]==' ' || str[i] '\t')
          flag=1;
  else
      if(flag==1)
      {  c++;
         flag=0;
      }
```

Q. WAP to input a string and check in string Palindrome is not.

```
#include<stdio.h>
void main()
{
   char str[100];
   int  c=0, Alphabet[27]={0};

for(i=0; str[i]!='\0'; i++)
   if(str[i]>='a' && str[i]<='z')
```

string Alphabet ... not recognize special character

frequency → ... Word ...

```
Alphabet [ str[i] - 97] ++ ;

else if (str >= "A" && str <= 'Z')
     Alphabet [str[i]-65]++;
for(i=0; i<=26 ; i++);
{ if(Alphabet [i]!=0)  c++; }
```

~~Q. WAP to input a string and find the frequency of each character in the string.~~

```
{
     if(Alphabet [i]!=0)
          c++;
}

if (c==26)
     printf(" Panagram");
else
     printf(" Not Panagram");

}
```

Q. WAP to input a string and find the frequency of each character in the str...

```
#include<stdio.h>
#include <string.h>
void main()
{
     int
     char
```

Scanned by CamScanner

Special characters in Byte Stream 0-255 Unsigned

Page No.

Shivalal

```
for( i=0 ; i<=127 ;i++)
{
    c=0;
    for(j=0; str[j]!='\0' ; j++)
    {
        if (str[j]==i)
            c++;
    }
    if(c>0)
    printf("The frequency of %c is %d", i, c);
}
}
```

Function:-

Date 23/10/18
Page No.

* **Function:-**
  set of statements designed for a particular task.

* **Advantages of function:-**
  (i) To reduce the length of code
  (ii) ~~To code~~ Reuseability of code.
  (iii) Reduce the complexicity.
  (iv) Easier Debuging and test.
  (v) Balanced load distribution among resources.

* **Types of Functions:-**
  1.) Pre defined / implicit / Built-in / library functions / Readymade function
  2.) User-defined functions

  Ex:- (i) printf(), scanf() → stdio.h
  (ii) pow(), sqrt(), sin(), asin() → math.
  (iii) strcmp(), strrev(), strlen().
  ↳ string.h

Ex:-(12):- Ex:- main()

Date / /
Page No
Shivalal

Function Declaration / Function Prototype :→

→ Syntax :-

Return-type Function_Name (Argument list);

datatype1 var1, datatype var2, ...

Optional.

Ex ① :

int add (int a, int b);

(or)

② int add (int, int);

③ char upper to lower (char);

float sub (float, int);

All conditions are mandatory

✓ No. of Arguments
✓ Types of Argument
✓ Order of Argument
✓ Return type
✓ function_name

→ Function defination :→

→ Syntax :-

Return_type function_name (Argument list)
datatype var. 1,
datatype var2, ...

{
Body
return var_name; //Optional;
}

Return_type: char, int, float, void

Body

3

Library → Function definition

Ex:-

```
int add(int a, int b)
{
int c;
  c = a+b
  return c;
}
```

③ Function Call:- (To use Function):-

→ Syntax:-

datatype result = function_name (Arguments);

Optional.

Var1, Var2, ...

(or)

Const1, const2, ...

Ex:- int result = add(4, 5)

or

int result = add(a, b)

Date / /
Page No.
Shivalal

Q. WAP to input 2 integer no. & add them using function.

```
#include<stdio.h>
int add (int, int);
void main()
{
  int a, b, s;
  printf("Enter 2 no.");
  scanf("%d%d", &a &b);
        s= add (a, b); printf("sum= %d", s);
}
     int add(int a, int b)
     {  int c;
        c = a+b;
        return c;
     }
```

Q. WAP to input & make factorial of No. using function.

(A)

```
#include<stdio.h>
  int fact(int);  // declaration function
  void main()
{
  int a, s;
  printf(" Enter no. = ");
  scanf("%d", &a);

      s= fact(a);  // calling function
  printf(" Factorial = %d\n", s);
}
```

```
int fact(int a)    // define function
{
    int c=1, i;
    for(i=1; i<=a; i++)
        c = c*i;
    return c;
}
```

Q. WAP to input & find a^b Using function

```
#include<stdio.h>
int power(int, int);
void main()
{
    int a,b,s;
    printf("Enter 2 no.=");
    scanf("%d %d", &a, &b);
    s = power(a,b);
    printf("Factorial = %d\n", s);
}
    int power(int a, int b)
    {
        int c=1, i;
        for(i=1; i<=b; i++)
            c = c*a
        return c;
    }
```

Categories of functions Based on Argument & Return type :-

	Return type	Argument	
(1)	✓	✓	Function with Argument & Return type
(2)	✓	×	Function without Argument but with Return type
(3)	×	✓	Function with argument but no return type
(4)	×	×	Function with no argument & return type

(2) Function without Argument but with Return type :-

```
int add( );          //declaration function.
void main()
{
   int s;
   s = add( );       // define function
   printf("%d", s);
}

int add( )           //define function
{
   int a, b, c;
   scanf("%d%d", &a, &b);
   c = a + b;
   return c;
}
// To save data on take less type to use this Function.
```

Date / /
Page No.

Shivalal

2) Function with Arg. but no return type:-

```
void add (int, int);
void main()
{
    int a, b;
    scanf ("%d%d", &a, &b);
    add (a, b);
}
void add (int a, int b);
{
    int c;
    c = a+b;
    printf ("%d", c);
    return c;
}
```

I/P → Call

Process o/p → [Add() called]

Argument	Return type	main() calling	add() called
✓	✓	I/P, call, O/P	Process
✓	×	I/P, call	Process, O/P
×	✓	call, O/P	Process/I/P
×	×	call	I/P, Process, O

Module: → Function

Shivalal

*) Function without any Arg & No Return type:-

```
void add( );
void main( )
{
    add( );

(O/P →

void add( )
{
  int a,b,s;
  scanf(" %d%d", &a, &b);

I/P → s= a+b;

  printf(" %d", s);
}
```

Q. WA to input & find a^b Using No Argument with return type.

```
#include<stdio.h>
  int power( );
    void main( )
  {
    int s;   s=power( );
     printf(" Factorial= %d\n", s);
  }
    {
      int power( )
      {
        int c=1, i, a, b;
```

```
printf("Enter 2 no.");
scanf("%d %d", &a, &b);
for(i=1; i<=b; i++)
    c= c*a;
    return c;
}
```

Q. WAP to input & find a^b Using Argument & no return type.

```
#include<stdio.h>
int power(int, int)
void main()
{ int a,b,s;
printf("Enter 2 no.");
scanf("%d %d", &a, &b);
        s= power(a,b)
} printf("Power = %d\n", c);
int power(int a, int b)
{
    int c=1, i;
    for(i=1; i<=b; i++)
    { c= c*a;
    }
    return c;
}
```

Date / /
Page No.
Shivalal

Q. WAP for input & find a^b Using No Argument & return type.

```
#include<stdio.h>
int power (int, int)
    void main()
{
   int s;
       s = power(a,b);
}
       int power(int a, int b)
{
    int c=1, i, a, b;
printf("Enter 2 no.");
 scanf("%d %d", &a, &b);

for(i=1; i<=b; i++);
{
        c = c * a;
}
printf("Factorial = %d\n", c);
        return c;
}
```

Only 1 time return code is execute.

we call by value only for 1 time return

Date 26/10/18
Page No.

Shivalal

* Call by Value:-

Q. WAP to multiply a no. by 10 using function

```
void multiply (int);
void main()
{
    int n;
    scanf("%d", &n);
    multiply(n)
}
void multiply(int n)
{
    n = n * 10;
    printf("%d", n);
}
```

(OR)

```
#include<stdio.h>
int multiply(int);
void main()
{
    int n;
    scanf("%d", &n);
    n = multiply(n);
    printf("%d", n);
}
int multiply(int n)
{
    n = n * 10
    return;
}
```

Q. WAP to input 2 values & swap them by call by value.

```
#include<stdio.h>
void swap(int, int);
{
    int n, m;
    scanf("%d %d", &n, &m);
        swap(n, m)
}
void swap(int n, int m)
{
    int t;
    t = n;
    n = m;
    m = t;
  printf("%d %d", n, m);
}
```

Q. Points to Remember:-

(i) In call by value a copy of actual argument is passed to the formal argument.

(ii) The changes made in the formal parameters will not change the actual parameter.

(iii) We can only return in call by value in parameter passing technique.

(iv) It is a time consuming process to copy actual argument to formal arguments if their are multiple no. of argument.

(v) It takes more memory to create duplicate / double the actual argument in a duplicate copy.

It is a secure parameter passing technique as the changes made in first module(function) will not reflect in second function or module.

Or. WAP. There are 2 friends A and B. Friend A enter no. and asks to find a no. next no. that no. is palindrome & even.

Q. WAP to largest among 3 no.

```
#include <stdio.h>
void largest (int, int, int);
  int main()
{
   int j, k, l;
printf("Enter 3 no.");
scanf("%d %d %d", &j, &k, &l);
      largest(j, k, l);
}
  void largest (int j; int k, int l)
{
     if (j>k && j>l)
        printf("%d is largest no.", j);
   else if (k>l)
        printf("%d is largest no.", k);
   else
     printf("%d is largest no.", l);
  }
```

Q. WAP जहाँ पर Bit Position Differ in

Pointers :

Date 13/12/18
Page No.

Variable that stores address of another variable.

* Declaration :- datatype *pvar_name;

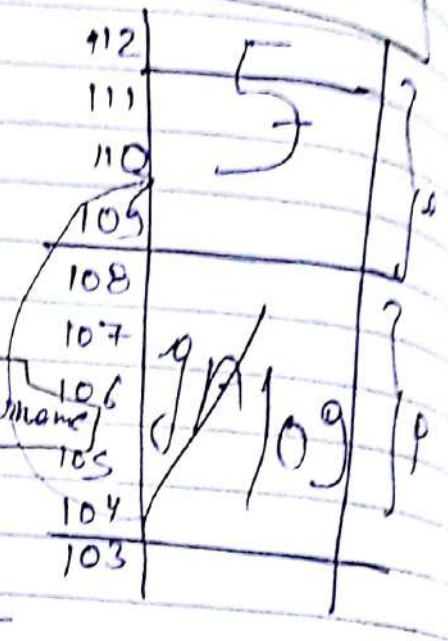
Eg:- int *p;

int a = 5;

Initialization :-

datatype *pvar_name = &variable;

*p = &a



109 [9A] 104 [5] 109

printf("%d", a); ⇒ 5

printf("%d", *p); ⇒ 5

printf("%d", &p); → 104

printf("%d", p); → 109

%u → Unsigned int

For +ve byte

%p, %u, %x → Address for pointer (To Print Address)

Result on different format specifier:—

```
#include<stdio.h>
void main()
{
    int *p;
    int a=5;
    p = &a;
    printf("%d\n", a);
    printf("%d\n", *p);
    printf("%d\n", &p);
    printf("%d\n", p);
    printf("%u\n", *p);
    printf("%p\n", *p);
    printf("%x\n", *p);
    printf("%o\n", *p);
    printf("%d\n", &a);
    printf("%c\n", &a);
}
```

→ 0x7bf9678a8
→ bf96788
→ 0x1.9cb8 ---
→ -1080657752
→ bf96788

Output:

```
5
-1080657752
-1080657756
-1080657564
3214309544
```

Q. WAP to input two floating no & add by pointer.

```
#include <stdio.h>
  void main()
{
  float a, b*p;
   scanf("%f %f", &a, &b);
        a += b;
        p = &a
  3 printf("%f\n", *p);
```

Q. WAP to input P, R, T
Find S.I.

$$S.I. = \frac{P * R * T}{100}$$

```
#include<stdio.h>
  void main()
{
  float a, b, s, si;
   float *p.a, *pb, *cs, *psi;
  p.a = &a;
   pb = &b;
   cs = &s;
  psi = &si;
   scanf("%f %f %f", p.a, pb, cs);
   * psi = (*p.a * *pb * *cs) / 100;
   printf("%f", *psi);
  3
```

Q WAP to find factorial.

```
#include<stdio.h>
void main()
{
   int a, f=1, i;
    int *pa, *pf, *pi;
pa = &a;
 pf = &f;
 pi = &i;
  scanf("%d", pa);
 for(*pi=1; *pi<=*pa; (*pi)++)
  {
    *pf = *pf * *pi;
  }
  printf("%d\n", *pf);
}
```

Q WAP Suppose an employee works for 10 hrs. a day and get the salary 10,000 for that particular week. Now we have to input no. of days employee for a particular company for particular week and find his total salary based on the following condition.

Over time	Bonus
1 & 0,<5	5000
6-10	10,000
>11	15,000

<0, ⟶ 10,000

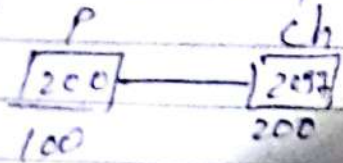
Date 4/11/15

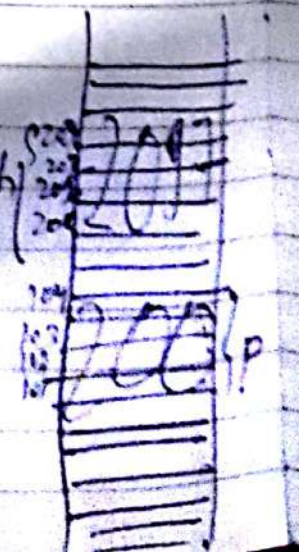
*

```
int *p;
char ch = 'c';
p = &ch; X ⟶ गलत लिखा Type मत्त
p = (int *) &ch; ✓

printf("%d", ch);   ⟶ 67
printf("%d", *p);   ⟶ garbage
printf("%d", p);    ⟶ 200
printf("%d", &ch);  ⟶ 200
```

*

```
char *p;
int ch = 2097
p = &ch; X ⟶ गलत लिखा Type Not Same
p = (char *) &ch; ✓

printf("%d", ch)    ⟶ 2097
printf("%d", *p)    ⟶ 49
printf("%d", p)     ⟶ 200
printf("%d", *(p+1)) ⟶ 8
```





2097 ⟶ 100000 11 0001

0000 1000 0011 0001

201 200

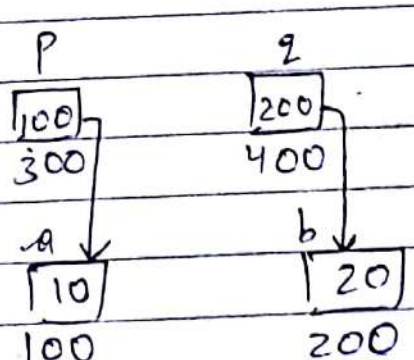
Date 6/11/15
Page No.
Shivalal

* (p+2) → 0
* (p+3) → 0
* (p+4) → garbage.

Pointer Arithmetic :-

(a) Addition of pointer with a constant value.
(b) Subatraction of pointer with a constant value.
(c) Subtraction of 2 pointer variables.
(d) Comparison of 2 pointers.
(e) Assignment one pointer to another.
(f) Increament & decrement on pointer.

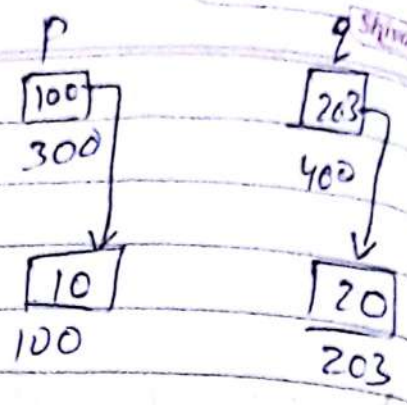
Ex:(1)
```
int *p, *q
int a=10, b=20, c;
p = &a;
q = &b;
c = q-p
printf("%d", c); → 25
```



⇒
c = q-p
= 200-100
= 100

No of integer = 100/4 = 25

*



$c = q - p$

$= 203 - 100$

$\frac{103}{4} = 25$

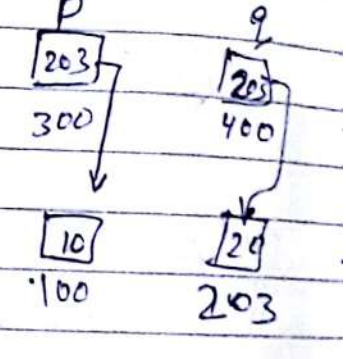
Ex (i)

```
int *p, *q
int a=10, b=20, c, d;

if (q>p)
    pf ("Hello")
else
    pf ("World")
```

⇒ Output :- Hello

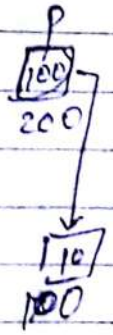
*

```
pf("%d", p) → 203
q → 203
&p → 300
&q → 400
*p → 20
*q → 20
```

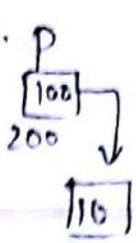


Ex (ii).

```
int *p;
int a=10;

p++;    p=104
++p;    p=108
p--;    p=104
p--;    p=100
```



Pre-increment
↓
Print Value
& Increase
decrease address.




```
int *p;
int a = 10;
printf("%d %d %d", *p, *p++, *p--, --*p);
```

Output :- 9 0 99

garbage

```
#include <stdio.h>
int main()
{
    int *p;
    int a=10;
    p = &a;
    printf("%d %d", *p++, *(--p));
}
```

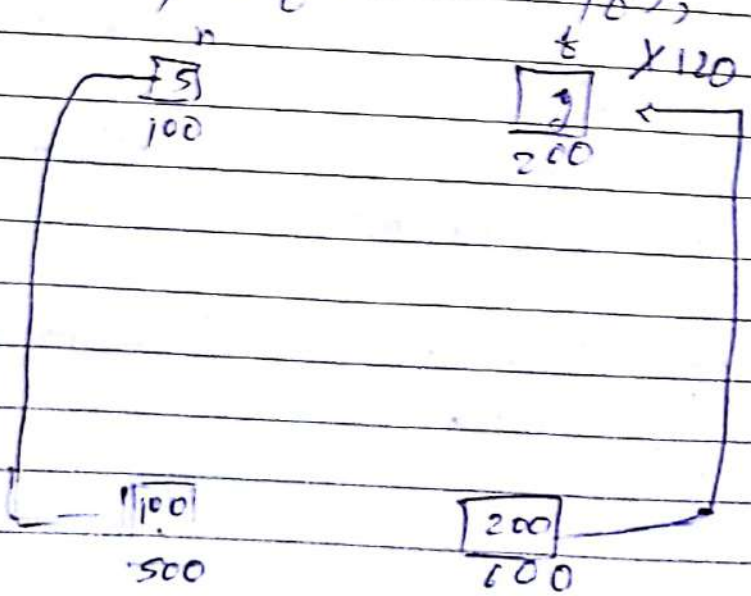
Output :- Garbage. Garbage.

Call by Value :-

1. In call by reference address of actual arguments not the values.
2. The change made by the formal paranders will also be reflected to the actual parameter.
3. In call by reference we can return 2 or more value in a function.
4. It takes less memory & less time compare to call by value.

Q. WAP to find the factorial of a no. using call by reference.

```
#include<stdio.h>
void fact (int *, int *);
void main()
{
    int n, f, *pn, *pf, i, *pi;
    pn = &n;
    pf = &f;
    scanf("%d", pn);
    fact(pn, pf);
}
void fact (int *pn, int *pf)
{
    *pf = 1;
    int i = 1, *pi = &i;
    for(*pi = 1, *pi <= *pn; (*pi)++)
    {
        *pf = *pf * *pi;
    }
    printf("%d", *pf);
}
```

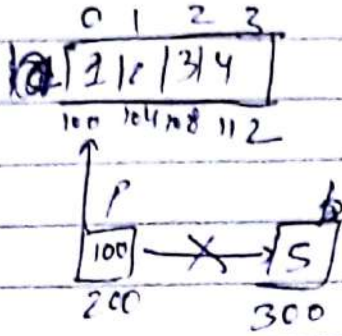


T* a = * a+1), ~(VCC).

Date 19/11/18
Page No.

* Pointers and 1-D Array :— (Point to an array)

int a[4] = {1, 2, 3, 4};
int *p;
*(p+1) = 2
a[0] = *(a+0)



# p = a+3; → 112
p[0] = *(p+0); → 4
p[-1] = *(p-1); → 3
p[+1] = *(p+1); → garbage

#

1-D Array	Pointer
(i) Array name is a constant pointer you cannot modify it	Pointer is a variable so it can verify
Ex: a = a + 1 not allowed (Error) Error (lvalue required)	Ex: p = p+1 allowed
(ii) We can not use increment & decrement on array	We can use increment & decrement on pointer.
Ex: a++ not allowed	Ex: p++ is allowed.

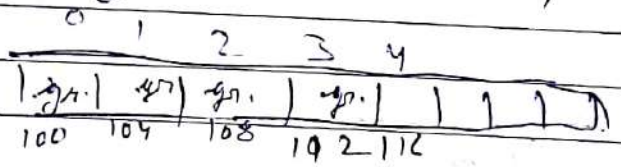
Scanned by CamScanner

Q: WAP to input a array & and display it using pointer 2 an array

Sol":

```
main()
{
    int a[100], n, *p;
    p=a;

printf("Enter size");
    scanf("%d", &n);
printf("Enter Array");
    for(i=0; i<n; i++)
        scanf("%d", p+i);

for(i=0; i<n; i++)
{
    printf("%d", *(p+i));
}
```

Ans":



Q.1: WAP to input a array & using pointer and find total no. of ~~pointer~~ positive & zero & also count.

Q.2: WAP to input a array using pointer and store all even value on even add. & also stord at odd with odd add.

Q.3: Insertion
Q.4: Deletion. → using Array or Pointer.
Q.5: Searching
Q.6: Sorting

printf("%d", a7 i+1); x

Page No. Shivalal

d''
```
main()
int a[100], n, *p, c=0, n=0, z=0
    p=a;
    printf("Enter size");
    scanf("%d", &n);
    printf("Enter Array");
    for(i=0; i<n; i++)
    {
        scanf("%d", p+i);
    }

    for (i=0; i<n; i++)
    {
        if(*(p+i) >0)
            c++;
        else
        if ( *(p+i) <0)
            n++;
        else
            z++
    }
        printf("%d %d %d", c, n, z);
```

soln

```
main()
int a[100], n, *p, b[100], c[100], *q, *r, j=0, k=0
    p=a;
    printf("Enter size");
    scanf("%d", &n);
    printf("Enter Array");
    for(i=0; i<n; i++)
    {
        scanf(" %d", p+i);
    }
```

a[] = *(a+0)
= *a

a[0] = *(a+0)
= *a

[ ]

Page No. / /

Shivalal

```
q = b;
r = c;

for(i=0; i<n; i++);

{
    if( *(p+i) % 2 == 0)
    {
        *(q+j) = *(p+i);
        j++;
    }
    else
    {
        *(r+k) = *(p+i);
        k++;
    }
}

for(i=0; i<j; i++)
{
    printf("%d\n", *(q+j));
}
for(i=0; i<k; i++)
{
    printf("%d\n", *(r+k));
}
```

Passing Array to function

Array & pointer All are same

Ex: WAP to input an array & find the average of all the elements by passing array to function.

0	1	2	3	4	5
4	7	2	9	3	6
100	104	108	112	116	120

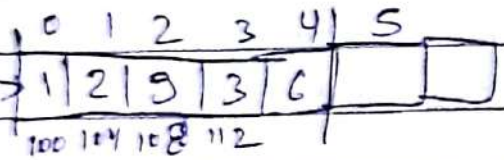
Soln:

```
float Arraysum (int *, int) , int [ ]);
void main()
{
    float avg;
    int a[100], n, i;
    printf("Enter size");
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        scanf("%d", & a[i]);
        avg = Arraysum (a, n);
        printf("%f", avg);
    }
float Arraysum (int a[100], int n)
               (a*a)
{
    int i, s=0;          -> printf("%d", sizeof(a));
    for(i=0; i<n; i++)
    {
        s = s + a[i]
            *(a+i)
        avg = (float) s/n;
        return avg;
    }
```

→ call by Reference

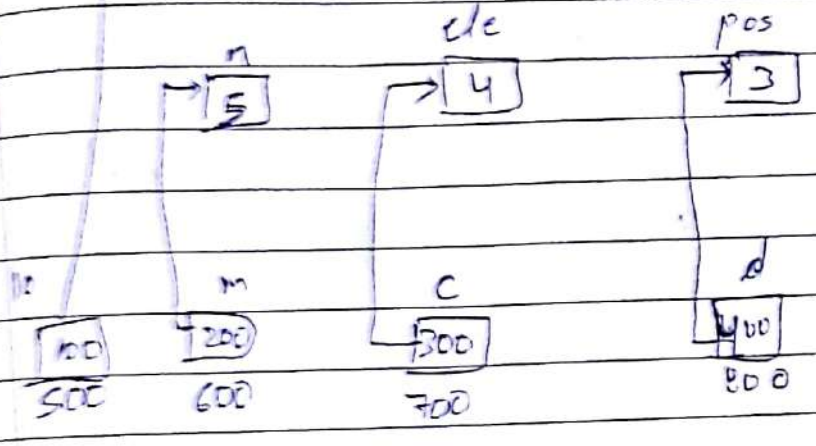
Insertion by passing array to function

```
void insert (int [], int *, int, int *);
main()
{
    int a[100], n, ele, pos, i;
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
    scanf("%d %d", &ele, &pos);
    insert(a, &n, &ele, &pos);
    for(i=0; i<n; i++)
    printf("
}
insert (int b[100], int *m, int *c, int *d)
{
    if(*d<0 || *d>=*m+2)
    {
        printf("Invalid position");
    }
    else
    {
        for(i=*m; i>*d; i--)
        b[i]
        *(be+i) = *(b+i-1);
        *(b+*d) = *c;
        *m = *m+1;
    }
```



| 1 | 2 | 3 | 4 | 5 | 6 |

n = 6



Q. WAP to input a no. and check whether no. is Perfect or prime not.

(

(i) Perfect no.
(ii) Robinson no.
(iii) Palindrome no.
(iv) Armstrong.

Pointer to String :-

```
char str[100];
char *p;
p = str;
```

✗ scanf("%s", p); Hello world → Hello

✓ scanf("%[^\n]", p); Hellowadd

(or)

```
gets(p);
printf("%s", p);
```

(or)

```
puts(p);
```

* **Length :-**

```
main()
{
    char str[100];
    char *p;
    int i=0;
    p = str;
    gets(p);
    for(; *(p+i) != '\0'; i++)
    {
        pl.
    }
    printf("%d", i);
}
```

0	1	2	3	4	5	6	7	8
H	e	l	l	o		H	i	\0

Length = 8.

Q. (i) Copy
(ii) Comparison
(iii) Reverse

Reverse.

by Pointer of String -

```
{
  char str[100], t;
  char *p;
    int i, j=0;
  p = str
  for(i=0; *(p+i) != '\0'; i++)
  {

  }
  i = i-1;  //i=4
while (i>j)
{
  t = *(p+i)
*(p+i) = *(p+j)
   *(p+j) = t
      i--;
      j++;
}

   puts(p);
```

(OR)

```
main()
{
  char str[100], t;
    int i, j=0;
  char *p, *q;

  p = str;
```

Comparison:

```
for(i=0 ; *(p1+i) != '\0' && *(p1+i) == *(p2+i); i++)
if(*(p1+i) > *(p2+i))
    printf("1");
else if(*(p1+i) < *(p2+i))
```

Date / /
Page No.
Shivalal

```
for(i=0; *(p+i) != '\0'; i++)
{

}
```

```
i = i-1;    // i=4
q = p+i;

while(q>p)
{
    t = *p;
    *p = *q;
    *q = t;
    p++;
    q--;
}
p = str;
puts(p);
}
```

Q.1: WAP to count the frequency of particular character

Q.2: to count the frequency of character present in string

Q.3: WAP to remove do extra white space.

Q.4: WAP to input String show UPPERcase, lowercase & special character.

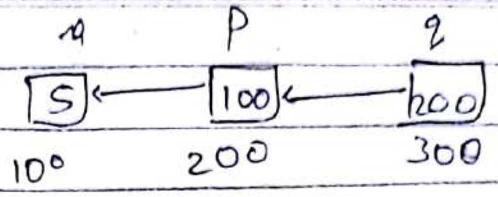
Q.5: WAP to convert lowercase to Uppercase

[Upto 12 levels :- → तक चल सकते हैं OR depend on Compile]

Date 26/11/2018

Pointer to Pointer:-

int a=5;
int *p;
p=&a;
int **q;
q=&p;

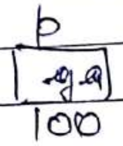


Output:-
*q=100
*q=5

printf("%d", a); → 5
&a → 100
p → 100
&p → 200
*p → 5
q → 200
&q → 300

NULL Pointer:-

→ If: No Initialise:- uninitialized pointers
datatype
int * p;



DevC :- → Runtime error;
GCC :- → Print the address.

p = 0 ~~or Address~~
&p = Address
*p = Runtime Error [Always]

Date / /
Page No.
Shivalal

(ii) Uninitialized Pointer:-

An uninitialised pointer is a pointer which points to a garbage address. Dereferencing uninitialized pointer may acess the value of that garbage address and Hence Our memory is not secure in this case.

(iii) NULL POINTER:-

NULL Pointer is a pointer which does not point anywhere. Deferencing pointer with result in run time error like Segmentation fault.

The Applications of NULL pointer is in designing of complex data structures like linked lists, trees, graphs, etc.

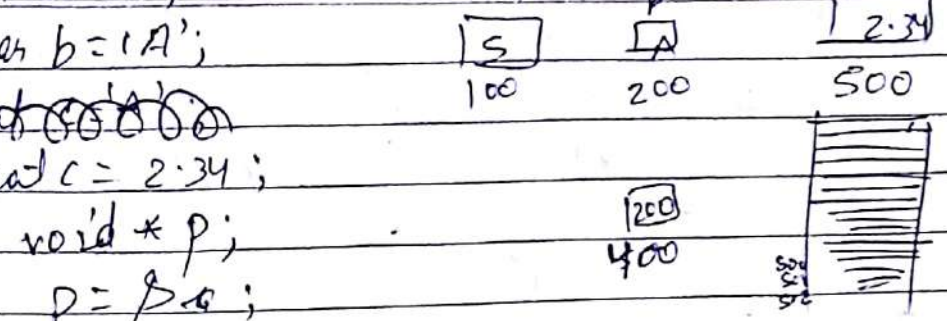
Here NULL is a predefined MACRO in header file (stdio.h).

(iv) Void Pointer:-

Void pointer is a pointer which can points to anywhere or variable of any type.

Date / /
Page No.
Shivalal

```
int a=5;
char b='A';
float c=2.34;
void *p;
p=&a;
```



in void.

Output

pf("%d", p); → 100

```
*(int *)p = 5;
p = &b;
*(char *)p = Run time Error
```

* Advantages of pointer :-

(i) Less Memory
(ii) We can access the memory directly with the help of pointer.
(iii) It makes program efficient and faster in excusing due to less memory or time (call by reference)
(iv) We can design the complex data structure like link list, trees, graph etc.
(v) We can allocate the memory dinamically In memory dinemication of memory hence there is no wastage like ~~wasting~~ in strink and increase of size.
(vi) We can also resize (expand or shrink) our memory block as per our changing requirement in dynamic memory allocation.

(vii) We can return 2 or more through a function indirectly [Ex. Call by Reference].

(viii) We can access the data structure like array, string efficiently using pointers.

* Drawbacks :-

(i) It makes our program complex and hence it make hard to debug or analyse.

(ii) In case of dynamic memory allocation we have to free the memory explicitly otherwise it will lead to memory likage

(iii) Wrong updation of uninitialized pointer may result of memory Corruption.

Pointer to 2-D Array :-

```
int a[4][3] = {1,2,3,4,5,6,7,8,9,10,11,12};
int (*p)[3];
p = a;
```

a	0	1	2
a[0] → 100	0	1 (00)	2 (01)	3 (02)
a[1] → 112	1	4 (10)	5 (11)	6 (12)
a[2] = 124	2	7 (20)	8 (21)	9 (22)
a[3] = 136	3	10 (30)	11 (31)	12 (32)

Output

pf("%d", a); → 100

a[0] → 100

*(a+0) → 100

**(a+0) → 1

p: 100 (200)

a[1] → 112

*(a+1) → 112

a+1 ⟷ 112 (100+1*3*4)

***a = *(*(a+0)+0) ⟶ 1

```
# for(i=0; i<4; i++)
  {  for(j=0; j<3; j++)
     scanf("%d", &a[i][j]); }
```

Date / /
Page No
Shivalal

Q. Write code of program :: ASCII?

```
#include<stdio.h>
void main()
{
char a;
printf(" Enter character = ");
scanf(" %c", &a);
printf("ASCII = %d", a);
}
```

Str :- Hello | GLA U

Q. int
char str[100], str1[100];

2nd :-

→ &a[0][0] = 100
&a[0][1] = 104

a[i][j] → *(*(a+i)+j)

a[i][j] = *(a[i]+j) = *(a+i)[j] = *(*(a+i)+j)

printf("%d", sizeof(p)); → 2 or 4 or 8

printf("%d", sizeof(*p)); → (1)

Q. Suppose the population of a city is 20,000 it gets increased by 10% during 1st year and 15% during 2nd year. Then find the population after 2 years.

sol^n:

```
# include<stdio.h>
void main()
{
int p=20000, r=10, t=15, f, s;
    f= p+ (p*10)/100;
      s= f + (f*15)/100;
printf("Total population= %d", s);
}
```

Q.

sol^n:

```
# include<stdio.h>
void main()
{
  float a;
  printf("Enter float no=");
    scanf("%f", &a);
  printf("FLOAT NO. = %e", a);
    return 0;
```

Input

↳ 432.92 ⟶ 4.329200e+002

Date / /
Page No
Shivalal

Q. Write a program to input integer value & display it.

Sol^n:

```
#include<stdio.h>
void main()
{
int a;
printf("Enter no.")
scanf("%d", &a)
printf("Display = %d", a);
}
```

→ Rules :-

(i) alphabets, digits and underscore (_)
(ii) can't use keywords
(iii) all variable are case sensitive.
(iv) first letter will not be a digit

Q. Write a program to input a 2 no. & sum of 2 no.

Sol^n:

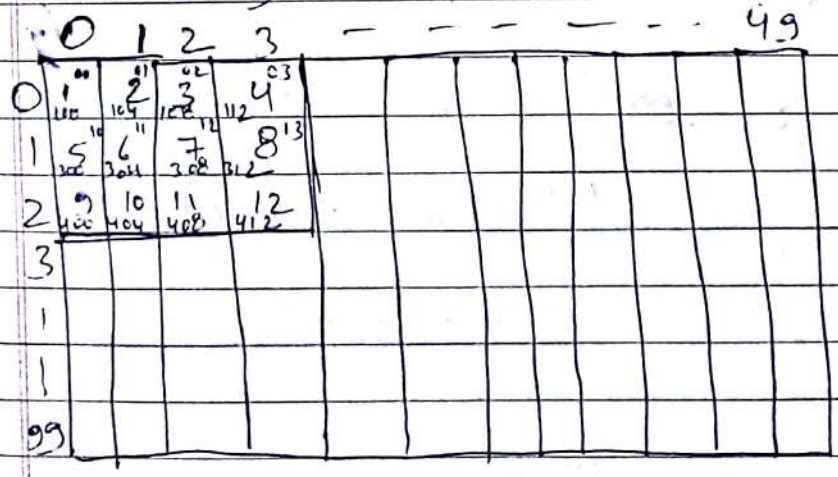
```
#include<stdio.h>
void main()
{
int a,b,c;
printf("Enter two no. a and b");
scanf("%d %d", &a, &b);
c=a+b;
printf("sum of two no. =%d", c);
}
```

Date 26/11/18
Page No.
Shivalal

2. WAP to input a 2-D Array and display it using ~~paid~~ pointer to 2-D Array.

pointer with 1-D Array

```
main()
{
  int a[100][50], (*p)[50], r, c, i, j;
  p = a;
  scanf("%d %d", &r, &c);
  for(i=0; i<r; i++)
  {
    for(j=0; j<c; j++)
    {
      scanf("%d %d", &a[i][j])
                      → *(p+i)+j
    }
  }
  for(i=0; i<a; i++)
  {
    for(j=0; j<c; j++)
    {
      printf("%d\t", *(*(p+i)+j));
    }
    printf("\n");
  }
```



Q.1: WAP to input the 2-D Array Using pointer to 2-D array. Find the sum of all the elements.

Q.2: WAP to input the 2-D Array Using pointer to 2-D array. Find the sum of all individual rows and all the individual columns.

Q.3: WAP to add matrices.

Q.4: WAP to find transpose

Q.5: WAP to print diagonal elements only.

Q.6: WAP to find sum of diagonal elements.

Soln 1:

```
main()
int a[100][50], i, j, r, c, sum=0
  p = a;
  scanf("%d %d", r, c);
  for(i=0; i<c; j++)
  {
    for(j=0; j<c; j++)
    scanf("%d %d", (*(p+i)+j));
    sum = sum + *(*(p+i)+j);
  }
}
printf("%d", sum);
```

Q.9: To print boundary elements.

Q.10: To check matrix is identity or not.

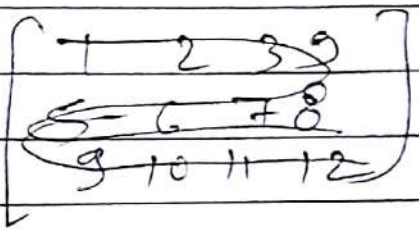
Q.11: To check matrix is Sparse or not.

——— Dense ———

Q.10:

Q.11: To print Matrix in zig zag form :-

Hint:



Q.12: Suppose their class in this class max. 10 students, Conduct 3 quiz for all the student & store the marks of all the students.
The students roll no. is 4 digit no. in increasing order (like 1001, 1002, - -)

Find...

(a) Average of all the quizes.
(b) min. & max in all the quizes. with marks & Roll no.
(c) Highest average of all 3 quizes.
(d) Average of students of 3 quizes.

Q:

1	2	3	10
4	5	6	11
7	8	9	12

Hint:- `i==0||i==m-1||j==0||j==n-1`

```
1  2  3  10
4           11
7  8  9  12
```

Soln:-

```
void main()
{
  int a[100][100], r, c, i, j, (*p)[100], ;
  p=a;
  printf("Enter size");
  scanf("%d %d", r, c);
  for(i=0; i<r; i++)
  {
    for(j=0; j<c; j++)
    {
      scanf("%d", *(p+i)+j);
    }
  }
  if(r==c)
  {
    for(i=0; i<r; i++)
    {
      for(j=0; j<c; j++)
```

```
if ( i==j || i+j==r-1)
{
    printf("%d\t", *(*(p+i)+j));
}
else
    printf("\t");
}
printf("\n"); }}
else
{
    printf(" Diagonal element of this
            matrix not define");
}
```

Sol^n:-

```
void main()
{
int a[100][100], r, c, i, j, (*p)[100], s=0;
  p=a;
printf(" Enter size");
    scanf("%d %d", &r, &c);
printf(" Enter matrix");
    for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        scanf("%d", *(p+i)+j);
    }
}
```

```
if(r==c)
{
for(i=0; i<r; i++)
    {
        for(j=0; j<c; j++)
        {
            if(i==j || i+j==r-1)
        {
    s=s+ *(*(p+i)+j);
        }
    else
        {
        printf("\t");
        }
    printf("\n");
    }
}
else
printf("Sum of diagonal elements
                    =%d", *(*(p+i)+j));
}

sol(): void main()
(print Elements)  {
int a[100][100], r, c, i, j, (*p)[100];
    p=a;
printf("Enter size");
    scanf("%d%d", &r, &c);

printf("Enter matrix");
```

```
for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        scanf("%d", *(p+i)+j);
    }
}

for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        printf("%d", *(*(p+i)+j));
    }
}
```

Soln:

```
void main()
{
    int a[100][100], r, c, i, j, (*p)[100];
    p = a;
    printf("Enter size");
    scanf("%d %d", &r, &c);
    printf("Enter matrix");

    for(i=0; i<r; i++)
    {
        for(j=0; j<c; j++)
        {
            scanf("%d", *(p+i)+j);
        }
    }
```

```
for(i=0; i<r; i++)
{
    for (j=0; j<c; j++)
    {
    if((i==0|| i==r-1 || j==0 || j==c-1)
    {
        printf("\n Print Boundary element\n");
        printf("%d", *(*(p+i)+j));
    }
}
}
```

Soln:

```
void main()
{
int a[100][100], b[100][100], c[100][100], r, c, i, j,
        (*p)[100], (*r)[100], (*s)[100];
    p=a;
    r=b;
    s=c;

for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        scanf("%d", *(p+i)+j);
    }
}
for(i=0; i<r; i++)
{
    for(j=0; j<c; j++) {
    scanf("%d", *(r+i)+j);
}
}
```

```
for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        scanf("%d", *(s+i)+j);
    }
}
for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        *(*(s+i)+j) = *(*(p+i)+j) + *(*(m+i)+j);
        printf("%d", *(*(s+i)+j));
    }
}
```

Soln:

```
int main()
{
    int a[100][50], *p[100], i, j, r, c;
    printf("Enter size");
    scanf("%d %d", &r, &c);
    printf("Enter matrix");
    for(i=0; i<r; i++)
    {
        for(j=0; j<c; j++)
        {
            p[i] = a[i];
            scanf("%d", p[i]+j);
        }
    }
```

```
for(i=0;i<r;i++)
{
    for(j=0;j<c;j++)
    {
        printf("%d", *p[i]+j);
    }
    printf("\n");
}
```

→ Using pointer

Array of pointers :-

// int a[5] = {5, 2, 9, 7, 13};

int a=5, b=9, c=13, d=11, e=6;

int *p[5] = {&a, &b, &c, &d, &e};

(04)

p[0] = &a;

p[1] = &b;

printf("%d", p); → 600

p[0] → 100

*(p+2); → 300

*(p-1); → 900

**(p+2); → 13

*3[p]; → 11

[2]p; → Error

5	9	13	11	6
100	200	300	400	500

p	100	200	300	400	500
600	602	604	606	608	610

(2).

int a1[3] = {5, 2, 7};

int a2[3] = {1, 3, 9};

int a3[3] = {4, 8, 10};

p	100	200	300
500	502	504	506

a1 → | 5 | 2 | 7 | — 2100 104 108

a2 → | 1 | 3 | 9 | — 200 204 208

a3 → | 4 | 8 | 10 | — 300 304 308

int *p[3] = {a1, a2, a3};

for(i=0; i<3; i++)

{

pf("%d", *(p+i));

Output: 100, 200, 300

pf("%d", **(p+i));

output: 5, 1, 4

pf("%d", *(*(p+i)+1));

output: 2, 3, 8.

*p[2] ✓
*p[2][2] ✓

(Use Double Pointer):-

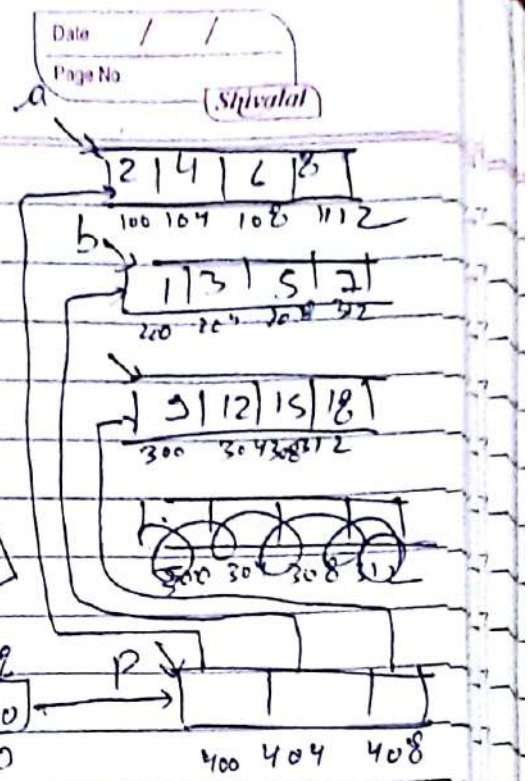
WAP to input a 2-D array & display.

```
{
int a[100][100], (*p)[100], i, j, r, c;

printf("Enter the size ");
scanf("%d %d", &r, &c);

for(i=0; i<r; i++)
    p[i] = a[i];
    (or)
    &a[i][0]

for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        scanf("%d", p[i]+j);
    }
}

for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    printf("%d", p[i]+j);
    printf("\n");
}
```

[Pointer to ... *(p+i)+j]

Date / /
Page No
Shivalal

```
int a[4]= { 2,4,6,8};
int b[4]= {1,3,5,7};
int c[4]= {9,12,15,18};
int *p[3]={a,b,c};
int **q= p.
```




```
printf("%d" p); →410
*(p+1); →200
p+5; →420
**(p+3); →gar.
**(p+1); →1
*2[p]; →9
(*1[p]+7); →8
q; → 400
&q; →500
*q; →100
**q; →1
q+1; →404
*(p+8); → gar.bagr.
```

Dynamic Memory Allocation:-

Date 3/12/18

1) malloc() ← memory allocation (For Normal Variables)

2) calloc() → Contiguous allocation (Array, String)

(i) stdlib.h (ii) alloc.h (iii)

3) realloc() → Re allocation (To expand or shrink memory).

4) free() → Release or delete or deallocate memory.

1) malloc:-

| default value | Garbage |

| datatype *p_var_name = (datatype *) malloc (sizeof... |

Ex:

(i) int *p = (int *) malloc (2 × sizeof(int));
*p = 20
*(p+1) = 20

(ii) char *p = (char *) malloc (2 × sizeof(int));

2) calloc:-

| default value = 0 |

| datatype *p var name = (datatype *) calloc (no. of blocks, size of each block |

Ex: (i) int *p = (int *) calloc (4, sizeof(float));

double :- Size :- 8 byts.

Date 3/12/15
Page No.
Shivalal

printf("%d", p); → 100
&p; → 0
p+1 = 104;

ii) double *p = (double *) calloc(4, sizeof(float));

3) ~~realloc~~ →

3) realloc() :→

p_var_name = realloc(p_var_name, New Size);

Ex: ~~p= realloc(p,20);~~
p = realloc(p, 20);

4) free() :→

free(p_var_name); (or) p = realloc(p,0);

* Dangling Pointer :-
To delete pointer if you delete block.

p = NULL;
p = realloc(p,0);

Q2- WAP to input an array and find the sum of elements using dynamic memory allocation.

Soln:

```
#include<stdlib.h>
#include<stdio.h>
main()
{
    int x, *p, i, s=0;
    printf("Enter size");
    scanf("%d", &n);
    p = (int*)malloc(n * sizeof(int));
                (or)
                n * 4
                (or)
    p = (int*) calloc(n, sizeof(int));
    printf("Enter elements");
    for(i=0; i<n; i++)
    {
        scanf("%d", p+i);
        s = s + *(p+i);
    }
    printf("sum = %d", s);
    free(p);
}
```

n
[5]
100

Date / /
Page No
Shivalal

Q: WAP to find the sum of n1 elements input by user & displayed & Now user enters a new size n2 and remaining no. of elements & find total sum of new using dynamic memory allocation.

```
#include <stdlib.h>
#include <stdio.h>
void main()
int n1, *p, i, s=0, n2;
printf("Enter size");
scanf("%d", &n1);
p=(int*) malloc(n1*sizeof(int));
(or)
s=(int*) calloc(n1, sizeof(int));
for(i=0; i<n1; i++)
{
scanf("%d", p+i);
s=s+*(p+i);
}
printf("Sum1=%d", s);
printf("Enter the new size")
scanf("%d", &n2);
p=realloc(p, n2*sizeof(int));
printf("Enter remaining elements");
```

n2 | 10 | n1 | 5 | 0

```
for(i=n1; i<=n2-1; i++)
{
    scanf("%d", p+i);
    s=s+*(p+i);
}
printf("sum2=%d", s);
free(p);
}
```

Q1. WAP to input an array & display the elements using DMA (Dynamic ...)

Q.2:- WAP to input an array & store even add in separate array.

Q.3:- WAP to input an array & count +ve, -ve or zero.

Q.4:- Insertion, deletion, Sorting, Searching.

Q.5:- Find max. & min. element in array & swap them.

Sol^n 5:

```
#include<stdio.h>
#include<stdlib.h>
{
    int *p, n, i, s, t
    p=(int *) malloc (n, sizeof(int));
    printf("Enter size");
    scanf("%d", &n);
    p=(int *) malloc (n*sizeof(int));
```

```
for(i=0 ; i<n; i++)
    {
scanf("%d", p+i);
    }
l = *p;
s = *p;

for(i=1; i<n ; i++)
    {
        if( l < *(p+i))
                l = *(p+i);
        if( s > *(p+i))
                s = *(p+i);
    }
printf(" Small=%d \n large=%d", s, l);
        free(p);
}
for(i=0; i<n ; i++)
{
    if(.
```

Date / /
Page No.

2D Array (Using Dma()):-

Eg:- `int *p = (int *) malloc ( m * n * sizeof(int) );`

```
for(i=0; i<m; i++)
{
    for(j=0; j<n; j++)
    {
        scanf("%d", p+i*n+j);
    }
}
```

Explain:-

	0	1	2
0	100	104	112
1			
2			
3			