*Find out the next element of the series:-

(i). 12, 11, 13, 12, 14, 13 15

(ii) 36, 34, 30, 28, 24 22

(iii). A, B, A, C, A, D, A, E, A

(iv). 3, 4, 6, 7, 9, 10, 12, 13, 15, 16, 18

(v) 1, 2, 3, ω, 9, t, 27, q, 81

(vi) JKLMNO, JKLMON, JKLOMN, JKOLMN. JOKLMN

→ (i). j+1, j j=11

j = j+1

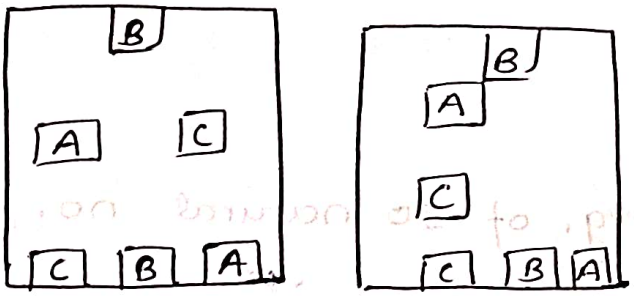
j+1 = j+2

(vii).

N	Output
0	0 2
1	3 5 7 9
2	6 8 10 12 14 16
3	9 11 13 15 17 19 21 23
4	12 14 16 18 20 22 24 26 28 30

find for no. 5.

*



Match the same alphabet without crossing the line.

* Computer programming and problem solving:-

→ Set of steps and instructions known as Algorithm. It is written in English like language also known Pseudocode.

Problem-1:- Calculate the average of 3 numbers.

Algorithm:- 1. Start

2. Read 3 nos. in variable n_1, n_2 and n_3.

3. Take variable sum.

4. Sum = $n_1 + n_2 + n_3$

5. Take variable average.

6. Avg = sum/3

7. Print Avg

8. End.

input/read/take

output/write/print.

Problem-2:- Calculate the avg of first 10 natural no.

Algorithm:- 1. Start

~~2. Take first~~

2. Read 1 as a.

3. Take variable sum as a+a+1+a+2+---+a+9.

4. Sum = a+a+1+ ---- +a+9

5. Take variable avg.

6. Avg = $\frac{sum}{10}$

7. Print Avg

8. End.

Problem-3:- Calculate the avg. of 10 natural no. (1 to 10)

Algorithm :- 1. Start

2. Take a variable 'counter' to hold 10 natural no. from 1 to 10.

counter = 1 — Initialise counter by 1.

3. Take a variable sum

sum = 0 — 4. Initialise sum by 0

5. Repeat while counter is less than or equal to 10.

5.1 sum = sum + counter

5.2 counter = counter + 1

6. Take a variable Avg.

7. Avg = $\frac{Sum}{10}$

8. Print Avg

9. End.

Algorithm and Flowcharts

Problem - Write an algorithm to sum the square of 5 numbers.

Algorithm:

1. Start
2. Read 5 nos. in variable a, b, c, d and e.
3. Take variable $a_1 = a \times a$, $b_1 = b \times b$, $c_1 = c \times c$, $d_1 = d \times d$ and $e_1 = e \times e$.
4. Take variable sum.
5. sum = $a_1 + b_1 + c_1 + d_1 + e_1$
6. Print sum
7. End.

Algorithm:

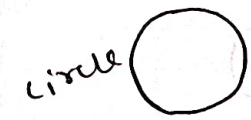
1. Start
2. Take variable 'n' to hold the 5 numbers.
3. Take variable sum to hold the addition of numbers.
4. Initialise sum = 0
5. Repeat upto 5 elements
   - 5.1 Input no. in 'n'
   - 5.2 sum = sum + n^2
6. print sum
7. End.

Problem - Write an algorithm to check greater no. between two.

Algorithm:

1. Start
2. Take two nos. a and b.
3. If a is greater than b.
   print a is greater
   else
   print b is greater
4. End.

* Flowchart are graphical representation of algorithm program.



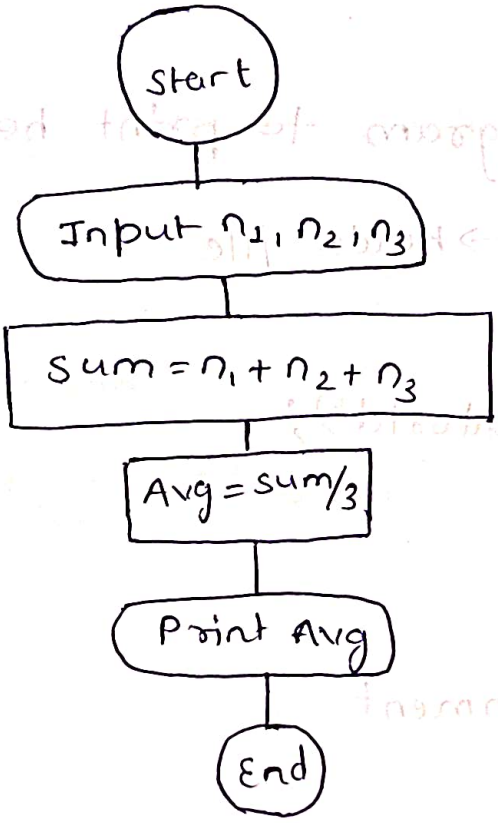
→ start

→ Input/output

→ Processing / calculation

→ ~~conclusion~~ condition. / check

→ ~~conclusion~~ connectors.



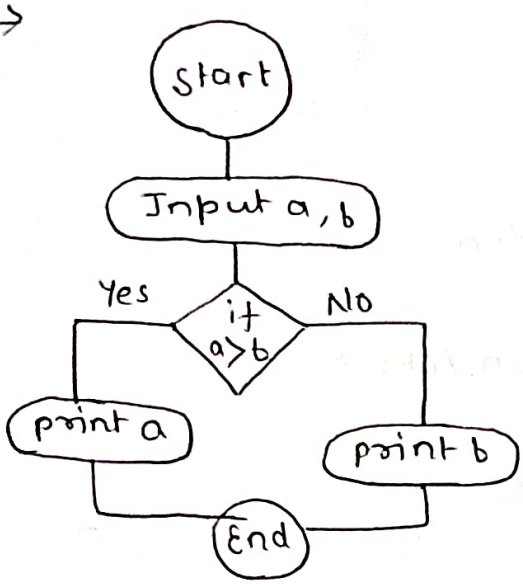
Q. Write an algorithm and draw a flowchart to calculate the area and perimeter of a rectangle.

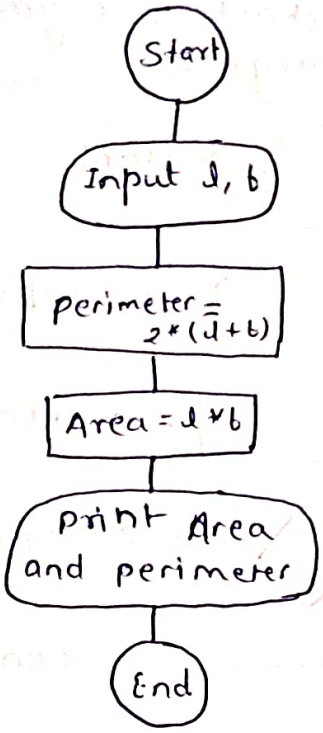
Algorithm:

1. Start
2. Take variables l and b
3. Take two new variables perimeter and area
4. perimeter $= 2*(l+b)$
5. area $= l*b$

6. print perimeter.
7. print area.
8. End.

→





*

```
/*this is the program to print hello world*/
#include<stdio.h>          → header file
int main()
{
   printf("Hello World");
   return 0;
}
```

⇒ /* -- */ → comment

⇒ # → preprocessor

⇒ stdio → standard input output.

⇒ int main()
   ↳ → prototype

⇒ main function does not give anything in return.

printf() → output

scanf() → input

```c
/*this is the program for sum of two numbers*/
#include <stdio.h>
int main()
{
  int a,b;
  int sum;
  printf("Enter first number:");
  scanf("%d",&a);
  printf("Enter second number:");
  scanf("%d", &b);
  sum=a+b;
  printf("Sum of two numbers is=%d",sum);
  printf("Sum of two numbers:%d",a+b);
  return 0;
}
```

⇒ & → Empersand

- Contents of C language:-

Constants
Variables
Keywords

Symbols/→
characters used in C language,

A-Z 0-9
a-z %, #, ', ;, :, -, +, |, *, ~, >, <, (),
{}, [], $, & ---- =,

* char c='A'; (single inverted commas)

Variable :- that can change or vary. It is an identifier.

e.g.- xyz_abc , a , abc, a2a,

Keywords :- the meaning of these words are already defined.

int	const	switch	do	case	struct
float	char	default	repeat	return	void
for	signed	union	until	long	enum
if	unsigned	while	goto	double	define

Data type :-

① Integer - simple members

1.1 Signed int → -ve & +ve

1.2 unsigned int → +ve

* only int can be written for signed int.

② char -

- Conditional statement :-

Q. Write a program to check given number is ~~keyword~~ even (or) odd

- Format specifier :-

```
int a;
char c;
float b;
scanf ("%f", &b)
scanf ("%d", &a)
scanf ("%c", &c)
printf ("Integer values %d", a)
printf ("Floating values %f", b)
```

- operators in C:-

① Arithematic operators - +, -, *, /, %

② Relational operators - ==, >, <, <=, >=, !=

③ Logical operators - &&, ||, !

1). Arithematic operators:- : ( ) ↑ /^

(i) Precedence :- highest : *, /, % ⎫ priority
lowest : +, - ⎭

(ii) Associativity:- (left to right) ⇒ in expression.

a↑b↑c ⇒ right to left. (in case of exponents)

* write a program to calculate interest given Principle, Rate & Time
* Calculate the area of circle, square, triangle, rectangle
* WAP to add digits of three digit number.
* WAP to add first and last digit of five digit number.

2). Relational / Conditional operators:-

a=5, b=3

```
if ( a == b)
    a > b
    a < b
    a <= b
    a >= b
    a != b
```

Relational operators
() me aayenge.

3). Unary operators :-

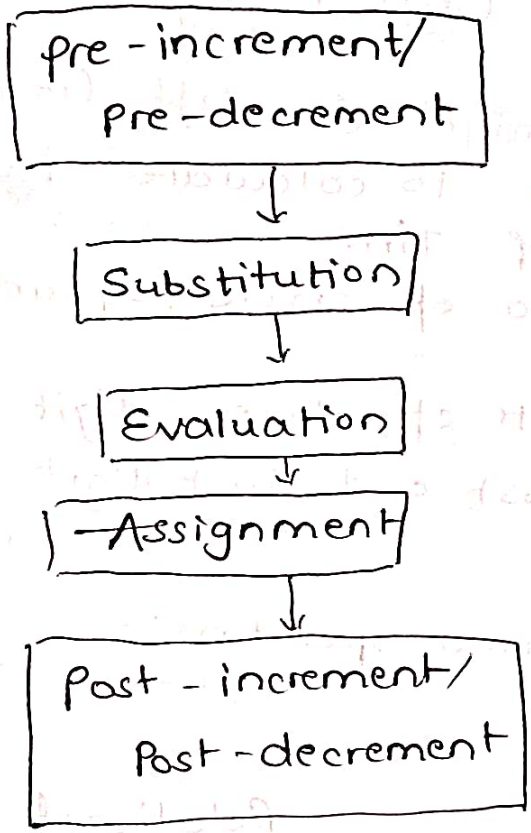
Increment → [Pre ++a, Post a++]

Decrement → [pre --a, post a--]

pre- means first increament,
-then assign it to another variable

post- means first assign it to another variable -then

assignment
evaluation
sum = a+b



* priority is ~~post~~ increment.

```
#include <stdio.h>
int main()
{
  int x=10, y;
  y = ++x;
```

```
    printf ("%d %d", x, y);
    return 0;
}
```

```
#include <stdio.h>
int main ()
{
  int x=10, y=20, z;
  z = x++ * --y;
  printf ("%d %d %d", x, y, z);
  return 0;
}
```

→

- Logical operators:-

  && = AND

  || = OR

  ! = NOT

→ ① Integers - simple numbers (int) 0-9

a. signed int - values can be taken in positive as well as negative

b. unsigned int - values can be taken positive

② float - 4 bytes values can be stored in float.

③. double - 8 byte values can be stored.

④. long int - 2 - 4

4 - 8

⑤. char - Data type in which alphabets are stored at a time only on character is stored.

- Conditional Statement :-

```
if (a > b)
    printf ("Greater no. is %d", a)
else
    printf ("Greater no. is %d", b)
```

- Escape character :-

```
printf ("Hello world \n");
printf ("Hello MANIT);
```

\n ⇒ new line

\t ⇒ tab (esc. character)

%f ⟶ %d ⇒ integer part

%c ⟶ %d ⇒ ASCII value

- Logical operators :-

* && (AND) → Returns true when all conditions under consideration are true and returns false when any one or more conditions false

* || (OR) → Returns true when one or more conditions under considerations is true and returns false when all conditions are false

```
int main ()
{
 int a = 10;
 if (a == 5 || a > 0 || a != 10 || a > 5)
     printf ("Hello MANIT");
     return 0;
}
```

* ! (NOT) → Return true if condition is false and false if condition is true (compliment of the condition)

- Conditional operators :-

1). If (conditional statements) :-

Note :- if ( ) ke andar me any value other than 0 then all results are printed.

0 → false.

1/Any no. → True

ex :-

```
int main ()
{
  int n=5;
   if (n>4)
   printf ( "%d", n);
    n++;
   printf ( "%d", n);
  return 0;
  }
```

* if (condition)
  statement;
  else
  statement;

* if (condition)
  {
  if (condition)
  statement;}
  else
  {statement;}

* if (condition)
  {statement;}
  else
  if (condition)
  statement;
  else
  statement;

* if (condition)
  {if (condition)
  statement;
  else
  statement;
  }

* else

```
{if (condition)
 statement;
else
    statement;
}
```

Q. Find Highest no. among 3 numbers:-

```
int main()
{int a,b,c;
 scanf("%d %d %d," &a,&b,&c);
 if (a>b)
  {if (a>c)
    printf("highest no. is %d",a);
   else
    printf("highest no. is %d", c);
  }
 else if (b>c)
   {
    printf("highest no. is %d," b);
   else
    printf("%d", c);
   }
    return 0;
  }
```

Q. Find out armstrong numbers less than 1000 ($153 = 1^3 + 5^3 + 3^3$) except 0 & 1.

```
int main()
{
int a,b,c, num,num1,num2, sum;
Scanf ("Enter a number = %d", &num);
if (1<num<1000)
 {
  a= num %10;
  num1 = num/10;
  b = num1 %10;
  num2 = num1/10;
  c = num2 %10;
  sum = a*a*a + b*b*b + c*c*c
  if (sum==num)
  {
  printf("Entered num is aomstrong:");
  else
   printf (" Not Armstrong");
   }
  else
   printf ("Condition not satisfied");
  }
  return 0;
 }
```

- BITWISE OPERATORS:-

& , | , ~ , ^ , << , >>

AND OR NOT XOR left shift Right shift

→ & (AND):-

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

→ ~ (NOT):-

a	~a
0	1
1	0

→ | (OR):-

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

```
#include<stdio.h>
int main(){
  char x=1, y=2;
  if (x&y)
    printf("result of x and y is 1");
  else
    printf("Nothing");
  return 0;
}
```

Nothing

→ left shift :- (<<)

4<<2

4<<1

→ 00000100 ← keval zero se replace karenge.

⇒ 00001000 → after 1 bit shift.

(5).

```
* #include<stdio.h>
  int main()
  {
float ~~int~~ u, a, b, c, num;
    printf("Enter a ~~number~~ unit: ");
    scanf("%f", &num);
    a = (50*0.0 + 50*0.30)
    b = a + (50*1.50 + 50*0.40)
    c = b + (100*2.50 + 100*0.50)

    ~~if (num>0~~
    if (num=<50){
      ~~pri~~ u = (num*0.0) + (num*0.30);
        printf("~~Unit~~ Bill is %f", u);
      }
    else if (50<num<=100){
        u = a + ((num-50)*1.50) + ((num-50)*0.40);
        printf
      }
    else if (
```

(6). Make appropriate conditions.

Q. WAP to multiply first 5 number and print the answer

```
#include<stdio.h>
int main ()
{
 int i, n;
 n=1;
 for (i=1; i<=5; ++i){
    n=n*i;
 } ~~main~~
 printf("Multiplied number is %d", n);
 return 0;
}
```

```
* int main ()
 { int num=1; //initializing the variable
  do  // do-while loop
  {
   printf("%d\n", 2*num);
   num++;    //incrementing operation
  }
   while (num<=10);
  return 0;
 }
```

Output

2
4
6
8
10
12
14
16
18
20

- Nested Loops :-

```
for (int i=0; i<=5; i++)
{
  for(int j=0; j<=5; j++)
  {
    printf("%d", i*j);
  }
}
```

*
```
for(int i=1; i<=5; i++)
{
  for (int j=1; j<=i; j++)
  {
    printf("%d\t", j);
  }
  printf("\n");
}
```

Output

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

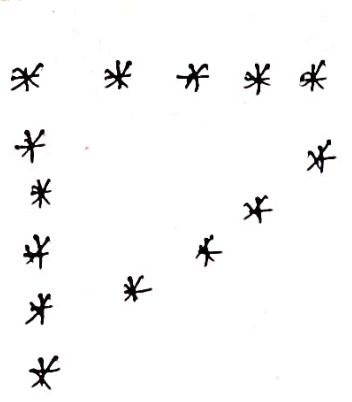
```
1
2 3
3 4 5
4 5 6 7
5 6 7 8 9
```

Q. 1 WAP in C to print the following patterns.

(a)
```
*   *   *   *   *
*                   *
*   *   *   *   *
```

→
```
#include<stdio.h>
int main()
{
  int i, j;
  for (i=1; i<=3; i++)
  {
    for (j=i; j<=i; j++){
      if (j%2==0){
        printf("*          *");
      }
      else {
        printf("*****");
      }
    printf("\n");
  }
  return 0;
}
```

(b).



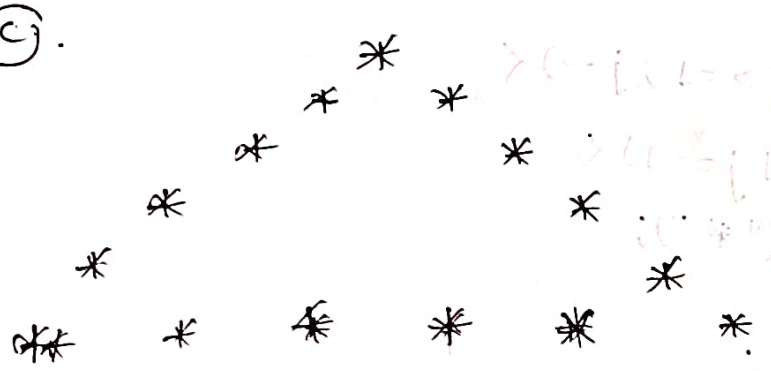
→
```
#include <stdio.h>
int main()
{
 for (int i=1; i<=6; i++)
 {
```

```
# include<stdio.h>
int main()
{
 for (int i=6; i>=1; i--)
 {
  ~~for(int j=i;~~
  if (i==6){
     printf("* * * * * *");
  }
  else{
     ~~for(j=1~~
     for(int j=i; j>=1; j--){
        if (j==i || j==1){
           printf("*");
        }
        else{
           printf(" ");
        }
  }
```

```
        printf("\n");
    }
    return 0;
}
```

④.
```
#include<stdio.h>
int main()
{
    for(int i=1; i<=6; i++){
        for(int j=1; j<=6; j++){
            if(i==1 || j==1 || i+j==7){
                printf("*");
            }
            else{
                printf(" ");
            }
        }
        printf("\n");
    }
    return 0;
}
```

⑤.



(c).

Q. WAP to swap two numbers using function parameter. without using 3rd variable

```
int a, b;
void swapno(a, b){

    a = a+b;
    b = a-b;
    a = a-b;
// after swapping the numbers.
    printf(" No. after swapping %d, %d ", a, b);

}
int main(){
void swapno(5, 8);

void swapno(9, 2);

return 0;

}
```

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

1 2 3 4 5

```
int n=5,
int n1, n2, n3, n4, n5;
```

Q- Sum all the value of array & print the sum:-

```
#include<stdio.h>
int main(){
 int a[5]={2,3,4,5,6};
 int sum=0;
 for (int i=0; i<5; i++)
 {
   sum = sum+a[i];
 }
 printf("sum of array : %d", sum);
 return 0;
}
```

3x3 → matrix. print in matrix form.

```
#include<stdio.h>
int main()
{
  int a[3][3];i,j;
  printf("Enter matrix elements:\n");
  for (i=0; i<3; ++i){
    for (j=0; j<3; ++j)
    {
      printf("Enter element a %d%d:", i+1, j+1);
      scanf("%d", &a[i][j]);
    }
  }
  print
  // Entered in matrix form
  for (
```