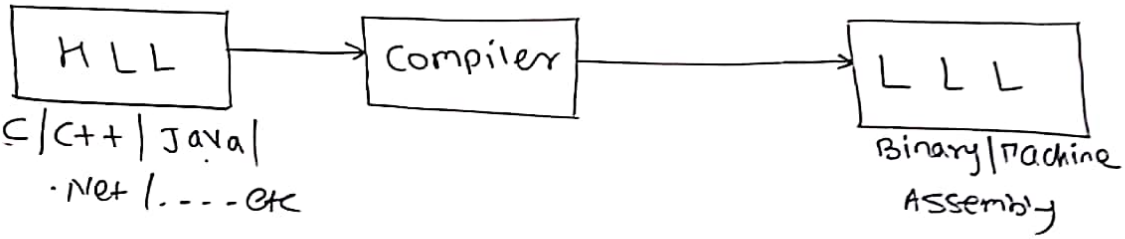
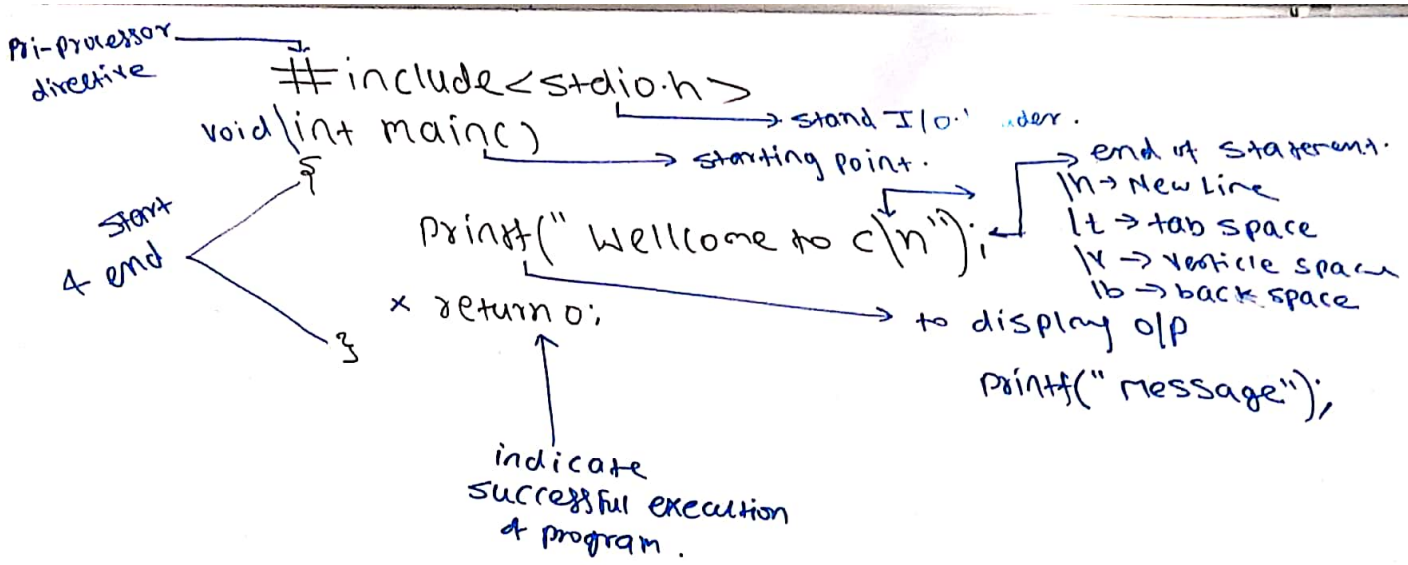
What is Programming

set of instruction  OS



Applications

1) Scientific calcy
2) Virus & Antivirus.
3) Mobile & PC games.
4) Mozila
5) Adobe



WAP to find area of circle

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int r;
    float a;
    printf("Enter radius\n");
    scanf("%d", &r);
    a = 3.14*r*r;
    printf("Area = %f", a);
    getch(); <= stop //
}
```

getch() ↑ Get character.

a = pi*r*r;

%d int → 5, 2, -10 ...

%f float → 5.2, 3 ...

%c char → a, A, b ...

O/P

Enter a radius

=

Area = 154.00

WAP to accept basic salary and calculate gross salary including : % DA 40% HRA.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float bs, gs, da, hra;
    printf("Enter basic salary\n");
    scanf("%f", &bs);          // 10000
    da = 50/100 * bs;          // 5000
    hra = 40/100 * bs;         // 4000
    gs = bs + da + hra;        // 19000
    printf("Gross Salary = %f", gs);
    getch();
}
```

O/P

Enter basic salary

10000

Gross Salary = 19000.00

WAP to calculate roots of quadratic equation by accepting coefficients from user.

$ax^2+bx+c=0 \Rightarrow x_1, x_2$

$= \frac{-b \pm \sqrt{b^2-4ac}}{2a}$

```
#include<math.h>
#include<stdio.h>
#include<conio.h>
void main()
{
    int a, b, c;
    float x1, x2;
    printf("Enter coefficient of quad. equation\n");
    scanf("%d %d %d", &a, &b, &c);
    x1 = (-b + pow((b*b - 4*a*c), 0.5)) / 2*a;
    x2 = (-b - pow((b*b - 4*a*c), 0.5)) / 2*a;
    printf("Roots are x1 = %f x2 = %f", x1, x2);
    getch();
}
```

Tokens

1) Character set :
   Alphabet : A-z, a-z;
   Digits : 0-9
   Symbol : @, #, $, % ....

2) Keywords :- It is reserve words.
   Total 32 keywords.
   eg :- int, float, if, break

3) Identifier :- Name given to memory, array, function, pointer.

Rules :
1) Combination of alphabet, digit & underscore.
2) Special symbols are not allowed except underscore.
3) Keywords are not allowed.
4) Blank space is not allowed.
5) Name cannot start with digit.

Variable & Constant :-

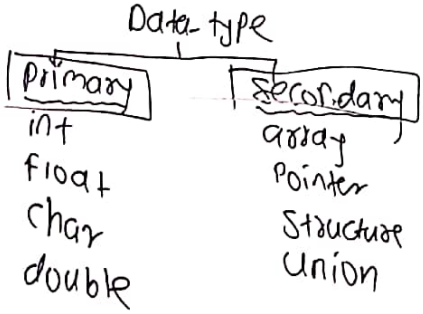
Variable :- place in computer memory to store data.

Syntax
data_type variable_name;

Constant : special variable whose data remain fixed.

Syntax
const datatype variable_name = value;
eg :
int x;
const float Pi = 3.14;

Data type : type of data.



operator :- operator is symbol used to perform operations on operands.

operator
- unary
- Binary
- Ternary

i) Unary minus (-)
eg:- int x = 4;
int y = -x;

ii) Address operator (&)
eg:- int x = 10;
int *y = &x;

iii) Casting operator :
conversion from one data type to another data type
eg:- int x = 65;
char ch = (char)x;

iv) sizeof() :-
It calculate size
eg:- int x = 4;
4 bytes ← sizeof(x);

v) Increment/Decrement
- Increment (++)
  It increases value by '1' digit
  - Pri-Increment
    eg:- int x = 4;
    int y = ++x;
    x = 5
    y = 5
  - post-increment
    eg:- int x = 4;
    int y = x++;
    x = 5
    y = 4
- Decrement (--)
  It decreases value by '1' digit.
  - Pri decrement
    eg:- int x = 3;
    int y = --x;
    x = 2;
    y = 2
  - post-Decrement
    eg:- int x = 3;
    int y = x--;
    x = 2
    y = 3

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int x=4,y=9;
    int z;
20  z=(x++)+(--y)+y;
    printf("Value=%d",z);
16  z=(--x)+x+(y--);
    printf("Value=%d",z);
    getch();
}
```

O/P

Value = 20

Value = 16

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a=10,b,c;
    c=b=a;
    b-=a--;      OR  b=b-(a--);
    c-=--a;      OR  c=c-(--a);
    a-=(--a)-(a--);  a=a-(--a-a--);
    printf("a=%d b=%d c=%d",a,b,c);
    getch();
}
```

6	0	2
a	b	c

Right then Left

O/P

a=6 b=0 c=2

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a=1,b=2,c=3,d=4.75,x;
    x=(++a)+(b++)*(++c)%d++;
    printf("%d %d %d %d %d",a,b,c,d,x);
    getch();
}
```

4) 8 (2
   8
   0

a	b	c	d	x
2	3	4	5	2

2 + 2 * 4 % 4

+ 8 % 4

o/p

2 3 4 5 2

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int x = 1;
    printf("%d %d %d", x, x=x+2, x<<2);
    x<<2;
    printf("%d %d %d", ++x, x++, ++x);
    getch();
}
```

6
x

1<<(2)
0001 ⇒ 0100

x 3<<2
0011 ⇒ 1100

O/P
3 3 4
6 4 4
//

Binary operator :-

1. Arithmatic operator :-

+, −, *, /, % are arithmatic operator.

'%' Mod, used to get remainder.

2. Relational operator :-

<, <=, >, >=, !=, ==

used to compare two operand.

	8	4	2	1
7 =	0	1	1	1
9 =	1	0	0	1
5	0	1	0	1

3. Logical operator :-

1. AND (&&) — if (condition1 (12th exam) && condition2 (Entrance))

2. OR (||) — if (condition1 (JEE) || condition2 (CET))

3. Not (!) — if (!(marks >= 40))

4. Bitwise operator :-

Operator which works on bit of data.

i) Bitwise AND (&)

5 & 3 = (1)

5 ⇒ 0 1 0 1

3 ⇒ 0 0 1 1

1 ⇒ 0 0 0 1

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

ii) Bitwise OR (|)

9 | 5 = (13)

9 = 1 0 0 1

5 = 0 1 0 1

13 = 1 1 0 1

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

iii) Bitwise Not (~)

~7 = −(8)

7 ⇒ 0 1 1 1

1 0 0 0

iv) Bitwise Ex-OR

6 ^ 4 = (2)

6 ⇒ 0 1 1 0

4 ⇒ 0 1 0 0

0 0 1 0

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

v) Bitwise left Shift (<<)

3 << 2 = (12) (2 times)

3 ⇒ 0 0 0 0 0 0 1 1

1st ⇒ 0 0 0 0 0 1 1 0

2nd ⇒ 0 0 0 0 1 1 0 0

vi) Bitwise Right Shift (>>)

13 >> 3 = (1)

Ternary operator :

It is also known as conditional operator.

Syntax

```
(condition) ? (Expr 1) : (Expr 2);
              True      False
```

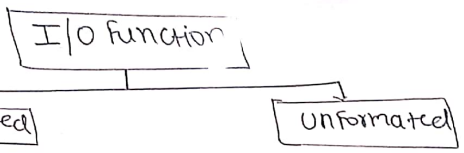
eg :-

```
int n1 = 10;
int n2 = 15;
max = (n1 > n2) ? (n1) : (n2);
15      10  15
```

WAP to find Largest of three number using conditional operator.

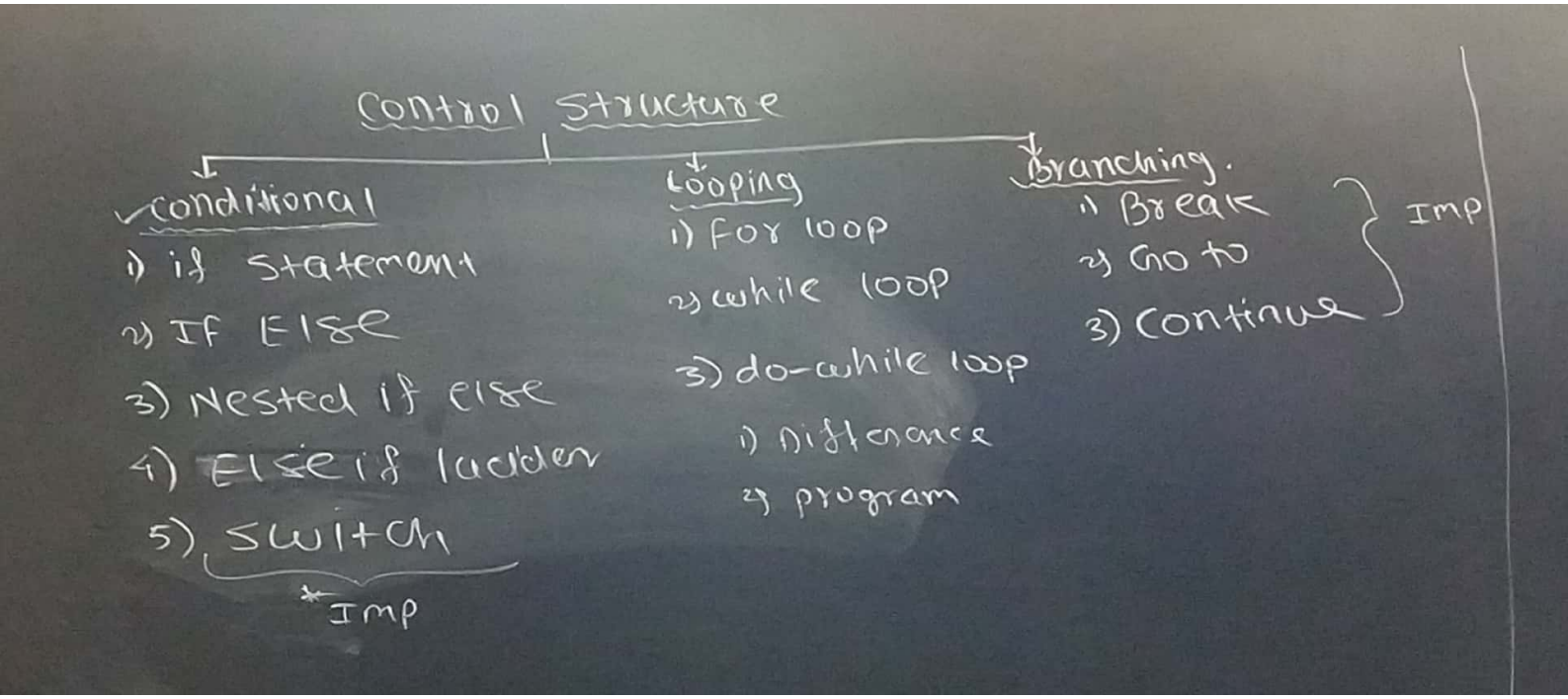
```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n1, n2, n3, max;
    printf("Enter three number\n");
    scanf("%d %d %d", &n1, &n2, &n3);
    max = (n1 > n2) ? ((n1 > n3) ? n1 : n3) : ((n2 > n3) ? n2 : n3);
    printf("Largest number = %d", max);
    getch();
}
```

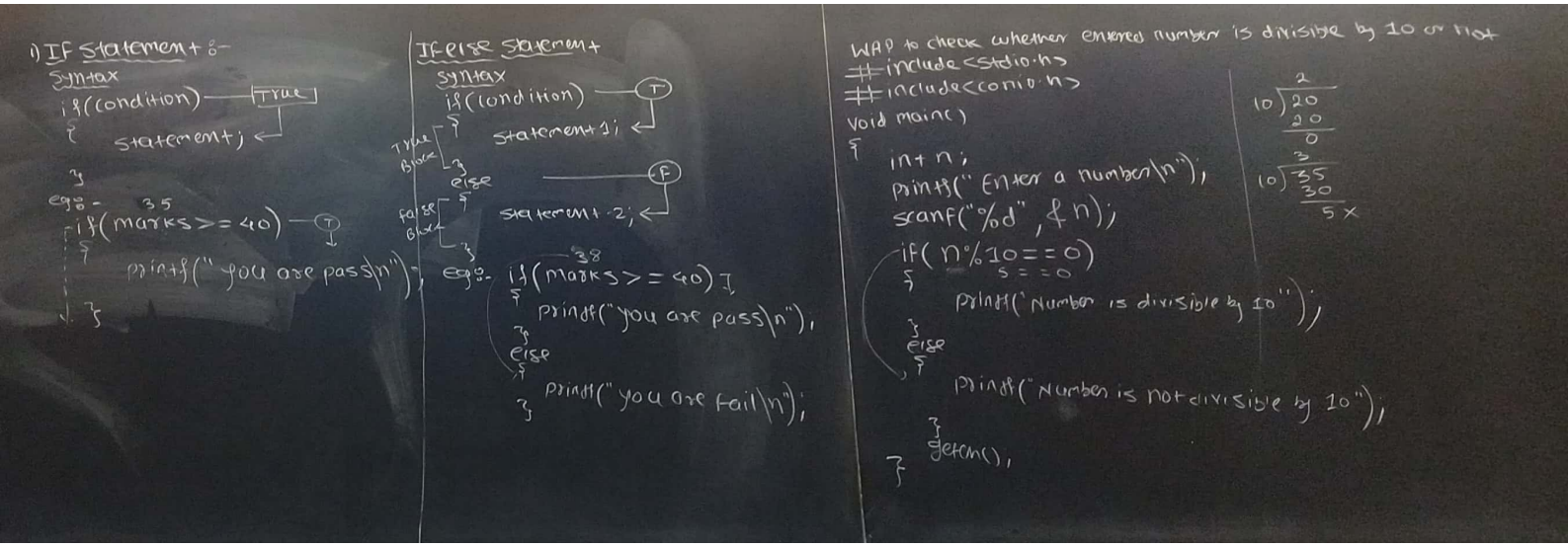
(n1 = 10, n2 = 15, n3 = 20; max = 20; (n1 > n2) is false, so the true branch ((n1 > n3) ? n1 : n3) is not taken; the false branch ((n2 > n3) ? n2 : n3) is taken.)

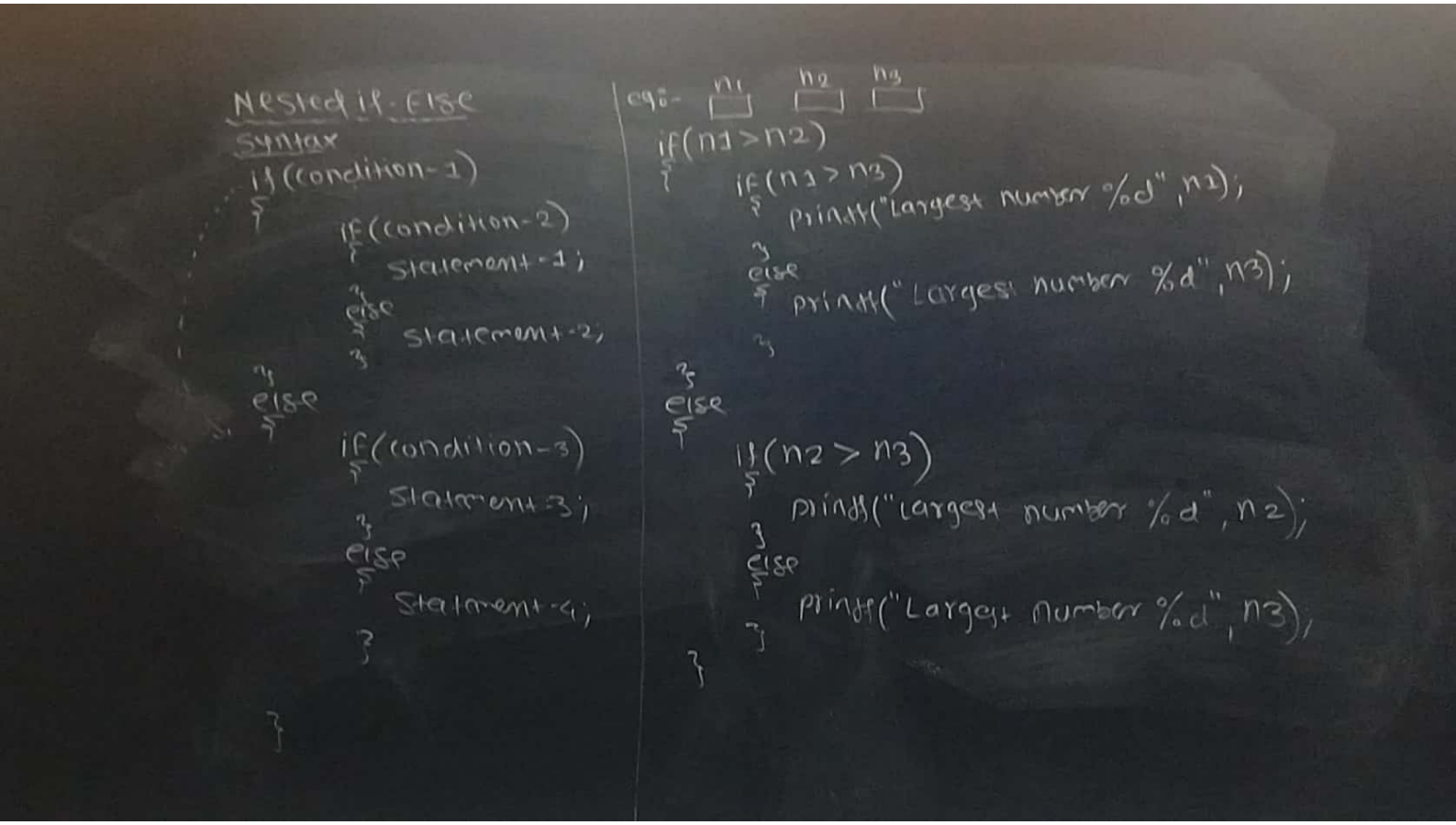


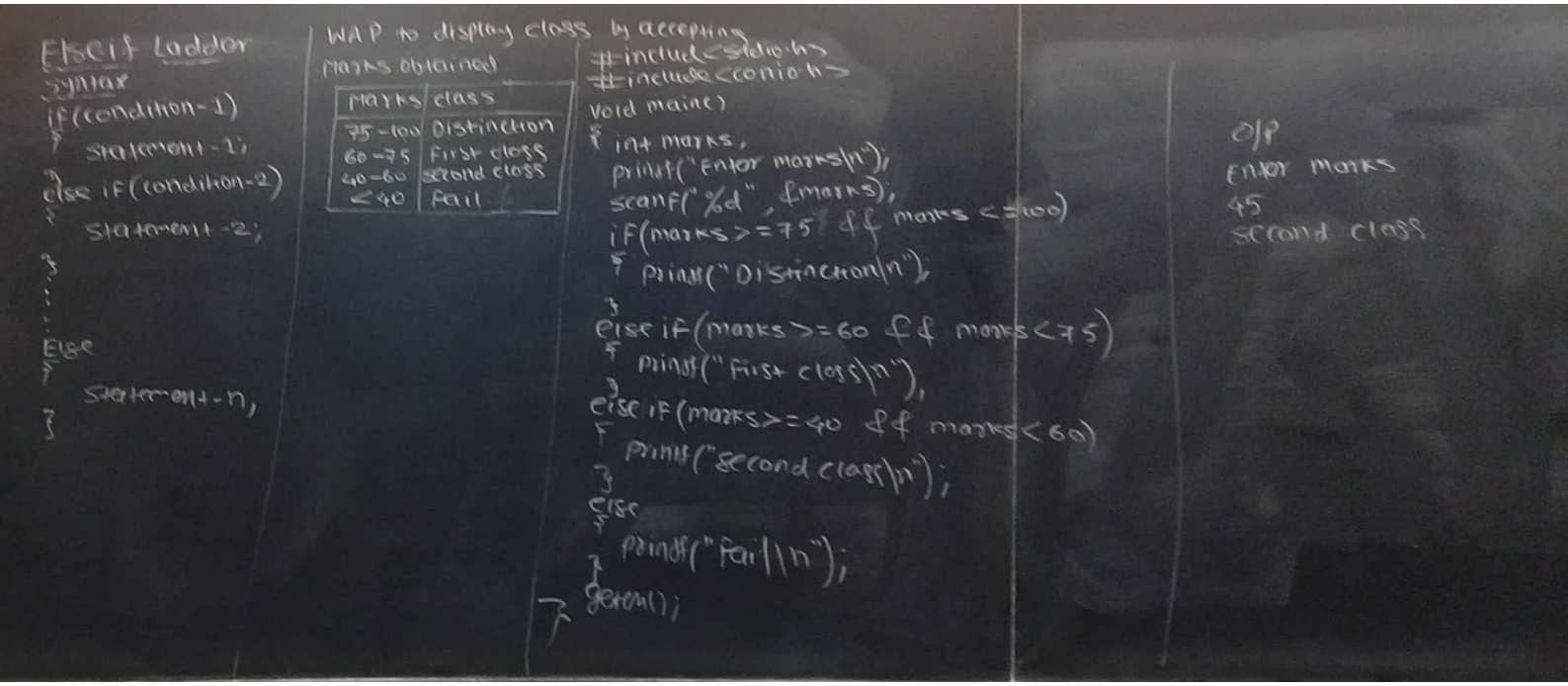
1) printf( );
2) scanf( );

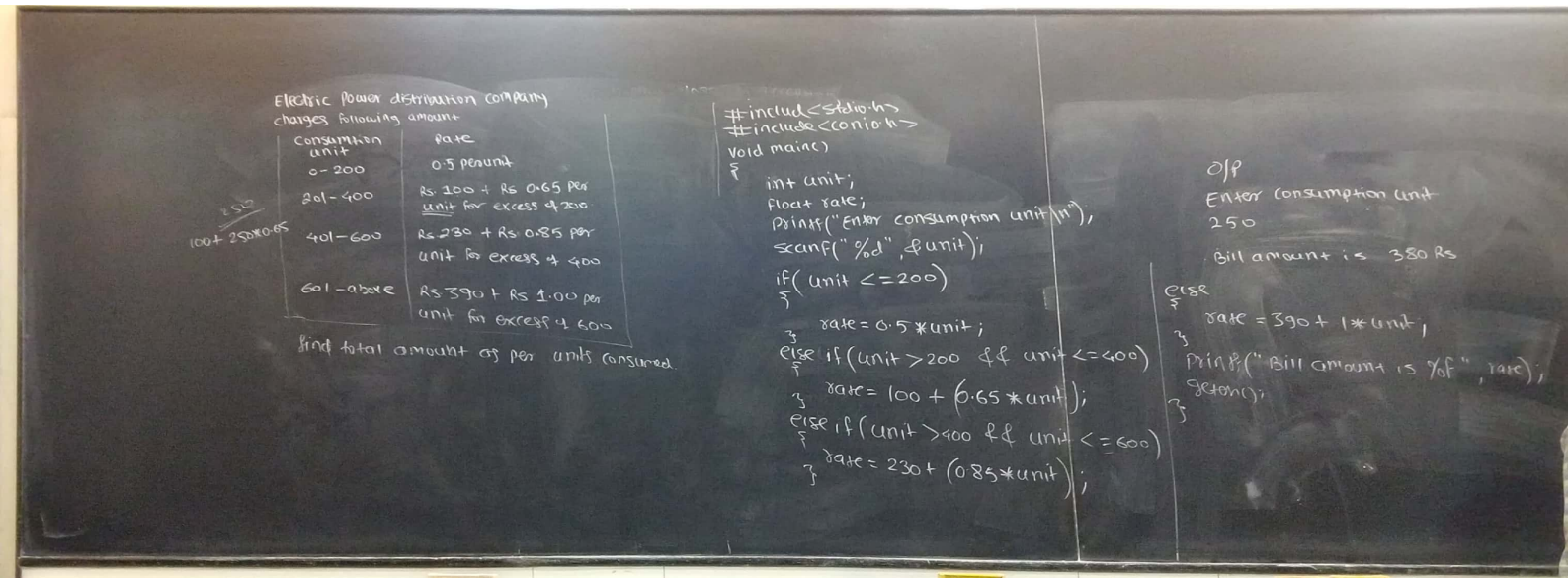
1) getchar() : It accept single character
2) getch() : It accept single character doesnot echo it to screen.
3) getche() : It accept single character echo it to screen but doesnot wait for Enter key to press.
4) putchar() : It display single character
5) gets() : It accept string (collection of character)
6) puts() : It display string.











Switch Statement

Syntax:-

```
switch (variable or expression)
{
    case label1 : statement-1;
                  break;
    case label2 : statement-2;
                  break;
    case label3 : statement-3;
                  break;
    .
    .
    .
    default : statement-n;
}
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int m;
    printf("Enter month number\n");
    scanf("%d", &m);
    switch(m)
    {
        case 1 : printf("JAN");
                 break;
        case 2 : printf("FEB");
                 break;
        case 3 : printf("MAR");
                 break;
        .
        .
        .
        case 12 : printf("DEC");
                  break;
        default : printf("Invalid month number");
                  break;
    }
    getch();
}
```

Switch Statement

Syntax:-

WAP to display number of days in a month using switch case.

Switch fall through.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int m;
    printf("Enter month number\n");
    scanf("%d", &m);
    switch(m)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: printf("31 days");
                 break;
        case 4:
        case 6:
        case 9:
        case 11: printf("30 days");
                 break;
        case 2: printf("28/29 days");
                break;
        default: printf("Invalid month number");
                 break;
    }
    getch();
}
```

Write a menu driven program to perform following operation
Addition/subtraction/Multiplication/
Division
by accepting choice from user.

O/P
Menu
1-ADD
2-Sub
3-Mult
4-Div
Enter your choice
4
Enter two number
5
2

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n1,n2,ch;
    float ans;
    printf("Menu\n");
    printf("1-ADD\n 2-sub\n 3-mult\n
        4-Div\n");
    printf("Enter your choice\n");
    scanf("%d",&ch);
    printf("Enter two number\n");
    scanf("%d %d",&n1,&n2);
```

```
switch(ch)
{
    case 1:
        ans=n1+n2;
        printf("Addition=%f",ans);
        break;
    case 2:
        ans=n1-n2;
        printf("Subtraction=%f",ans);
        break;
    case 3:
        ans=n1*n2;
        printf("Multiplication=%f",ans);
        break;
    case 4:
        ans=(float)n1/n2;
        printf("Division=%f",ans);
        break;
    default:
        printf("Invalid choice\n");
        break;
}
getch();
}
```

Break Statement

Break is used to transfer the control to end of loop.

Syntax

```
break;
```

eg:-

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

O/P

1 2 3 4

Continue:

continue is used to transfer the control to begining of loop.

Syntax

```
continue;
```

eg:-

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

O/P

1 2 3 4 6 7 8 9 10

Goto:-

goto is used to transfer the control from one part of program to another part.

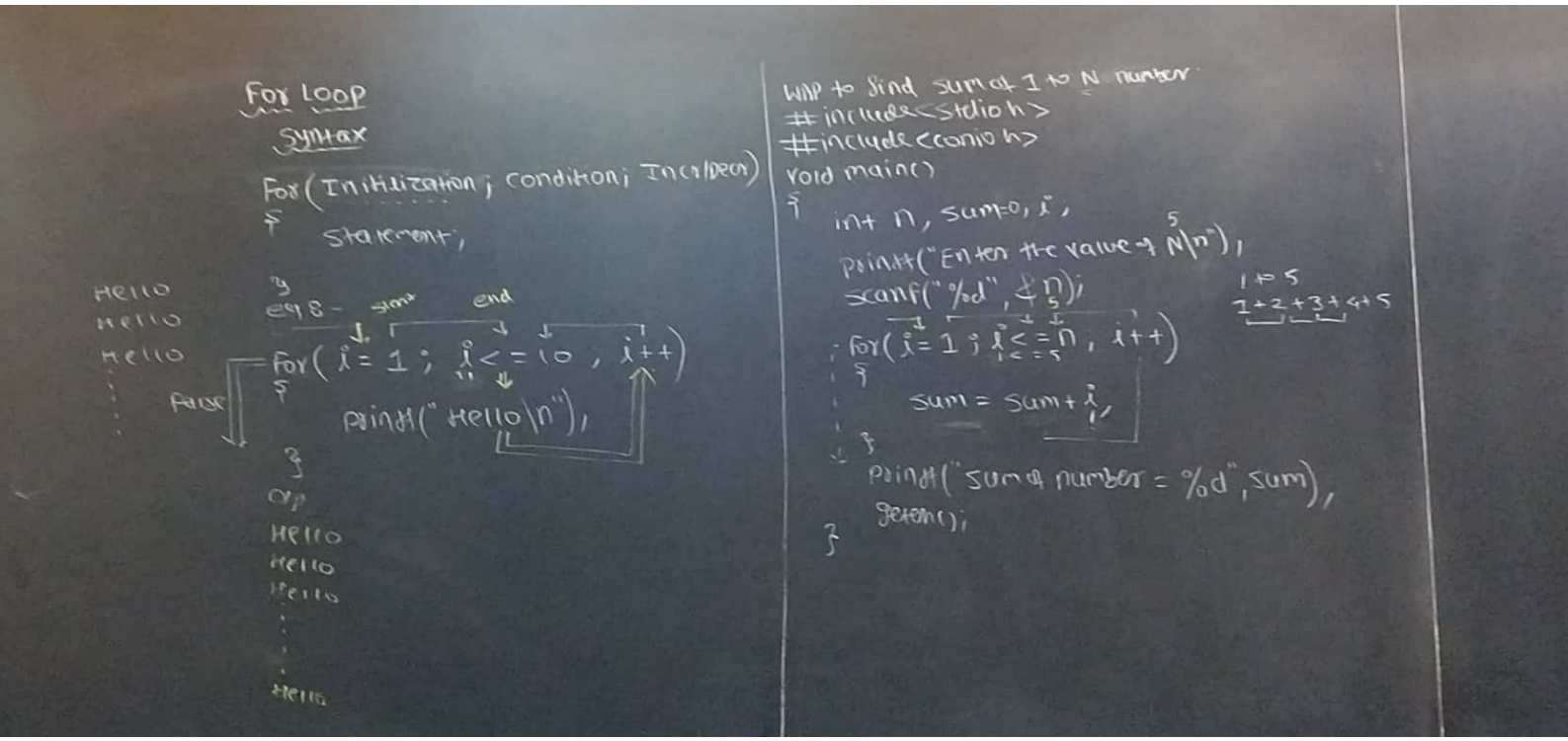
Syntax

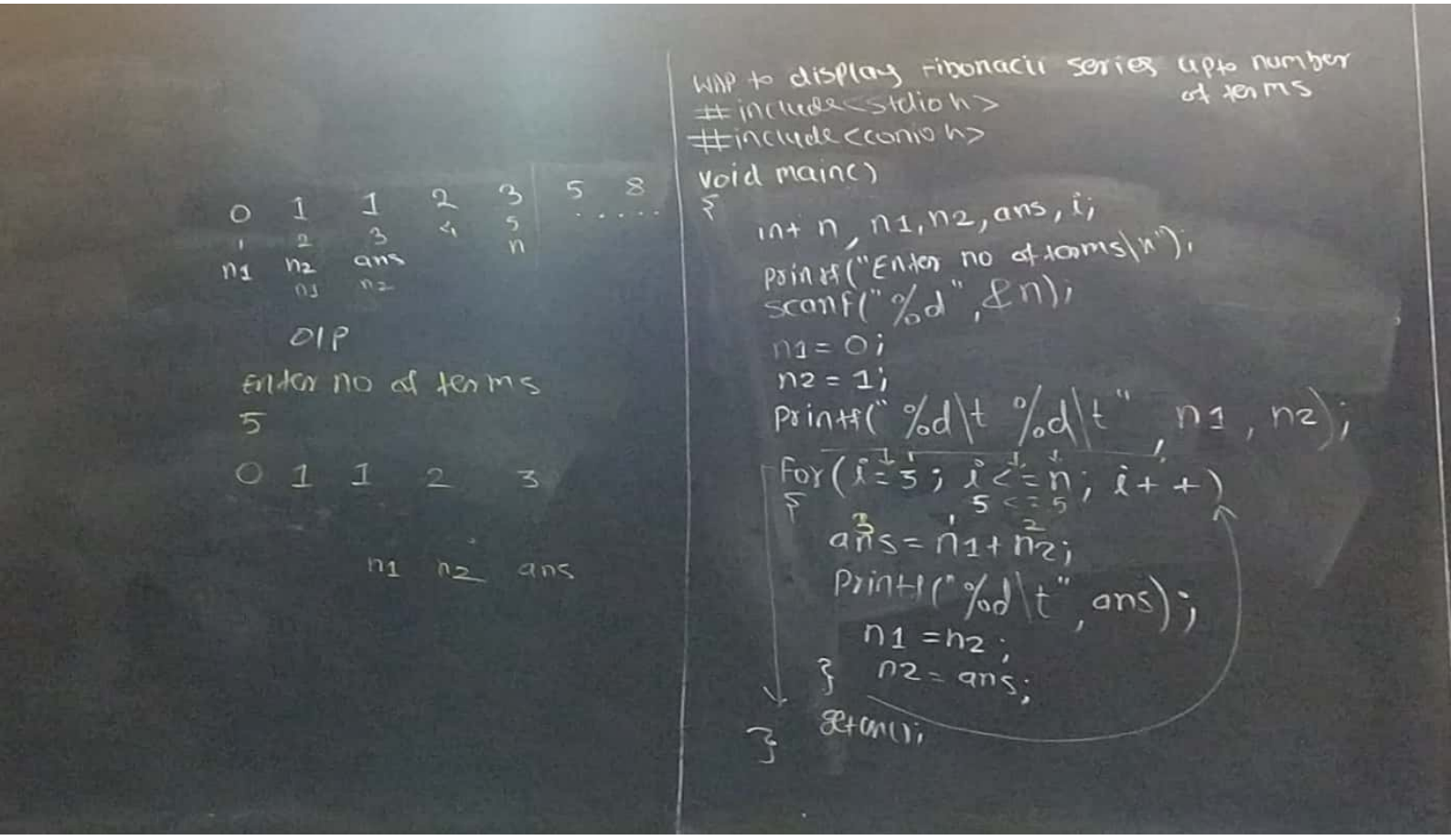
```
goto (label);
label:
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n;
    accept:
    printf("Enter a number");
    scanf("%d", &n);
    if(n<0)
        goto accept;
    printf("%d", n);
}getch();
```

O/P

Enter a number
-5
Enter a number
-2
Enter a number
5



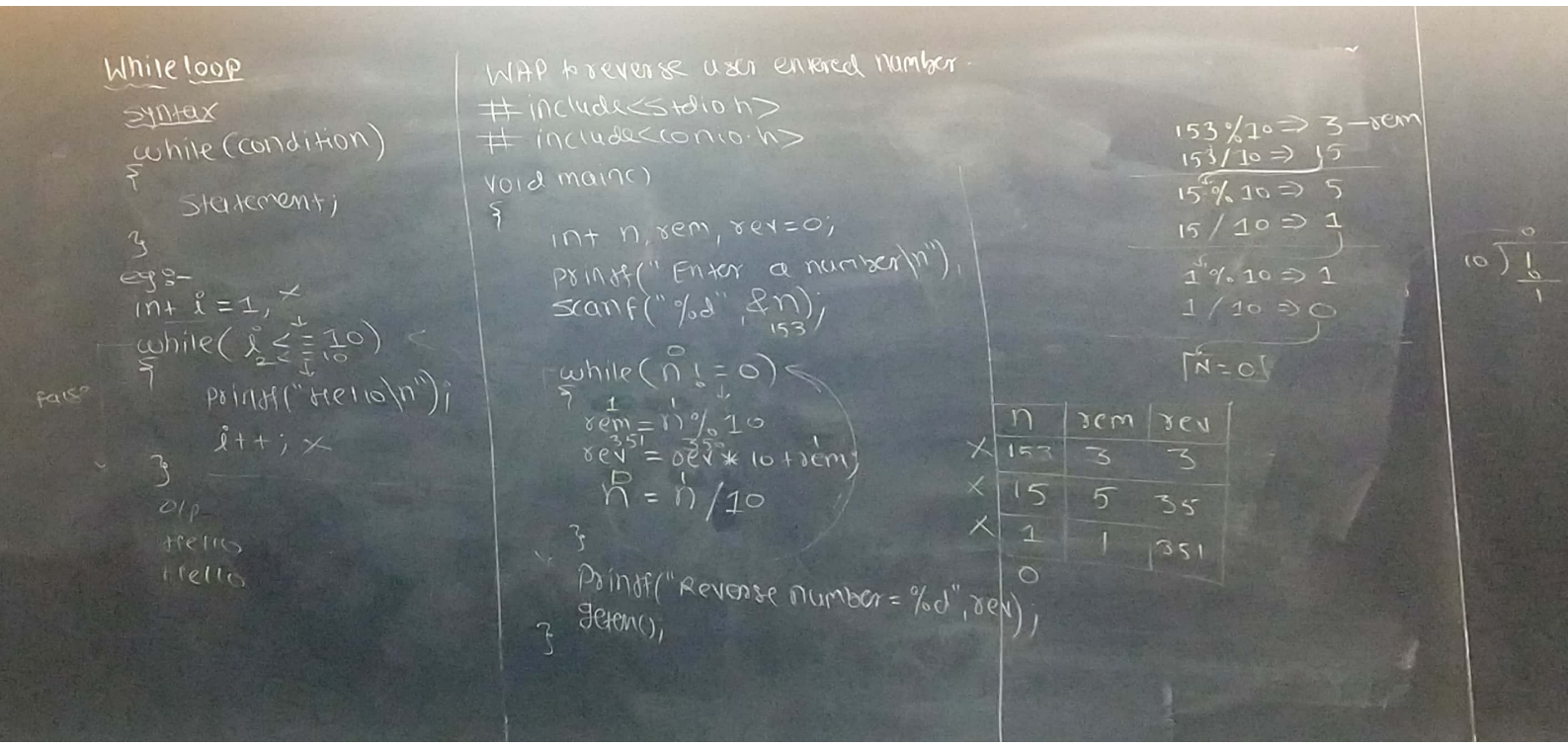


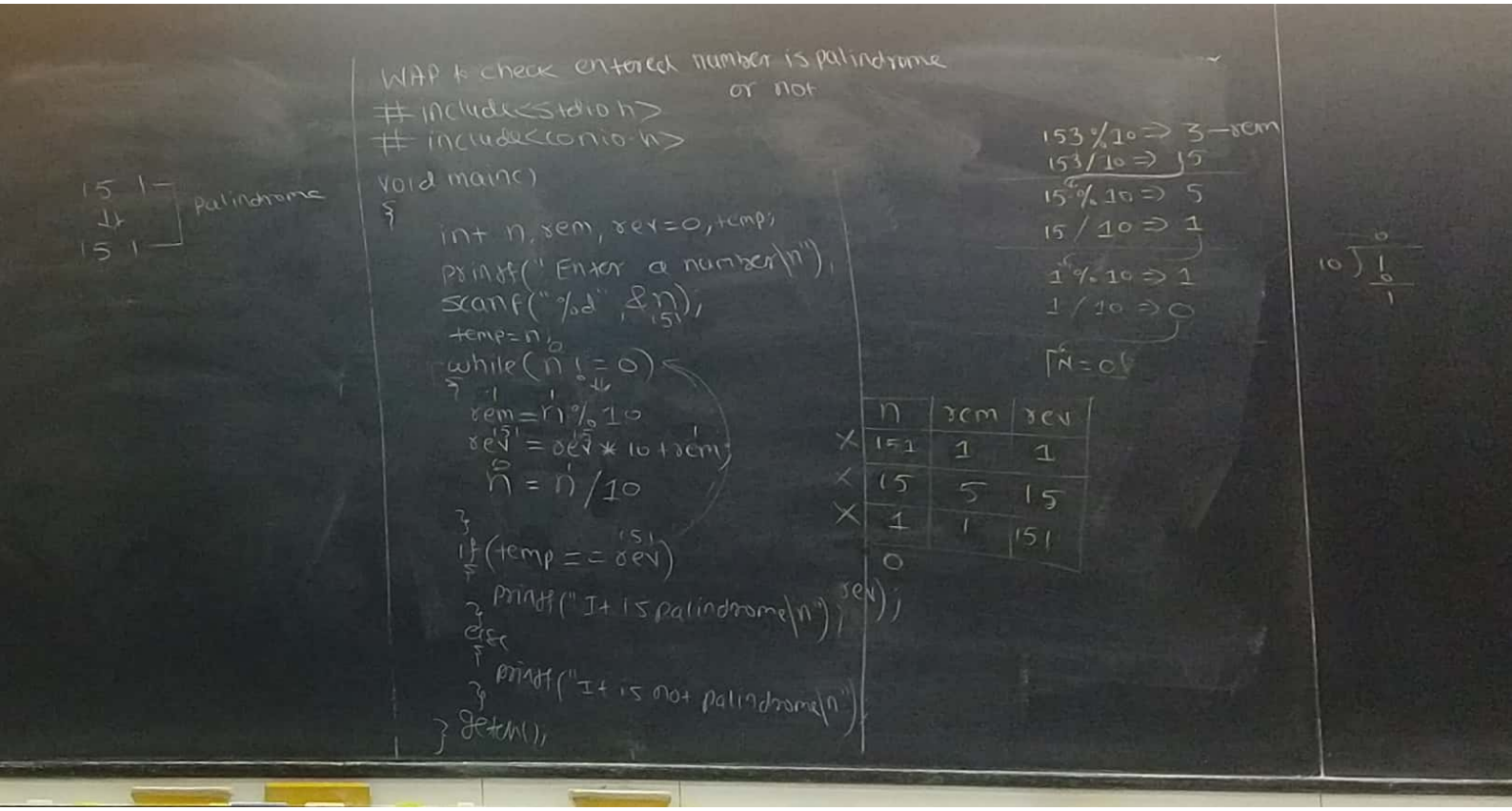
WAP to check entered number is prime number or not

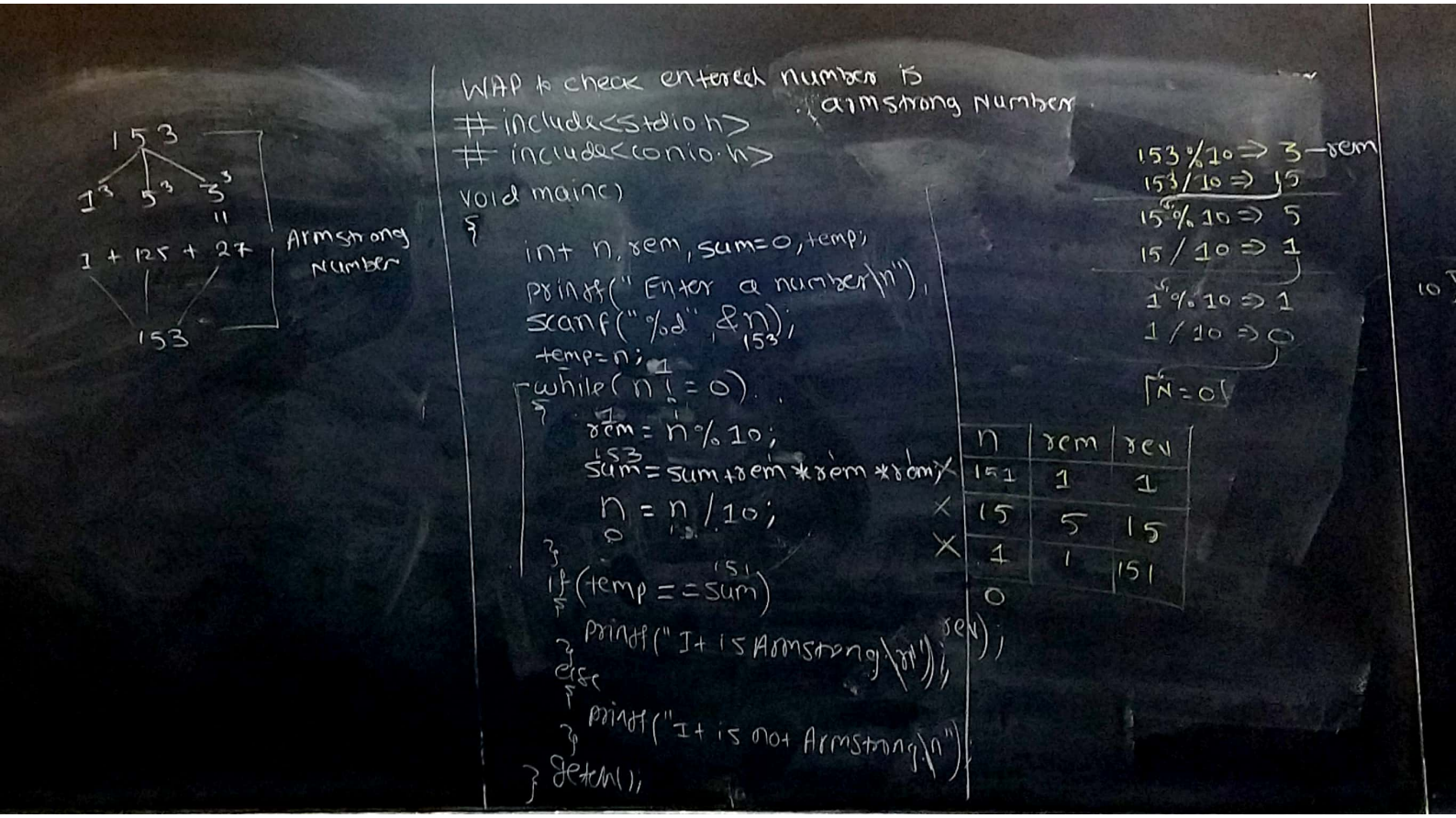
O/P

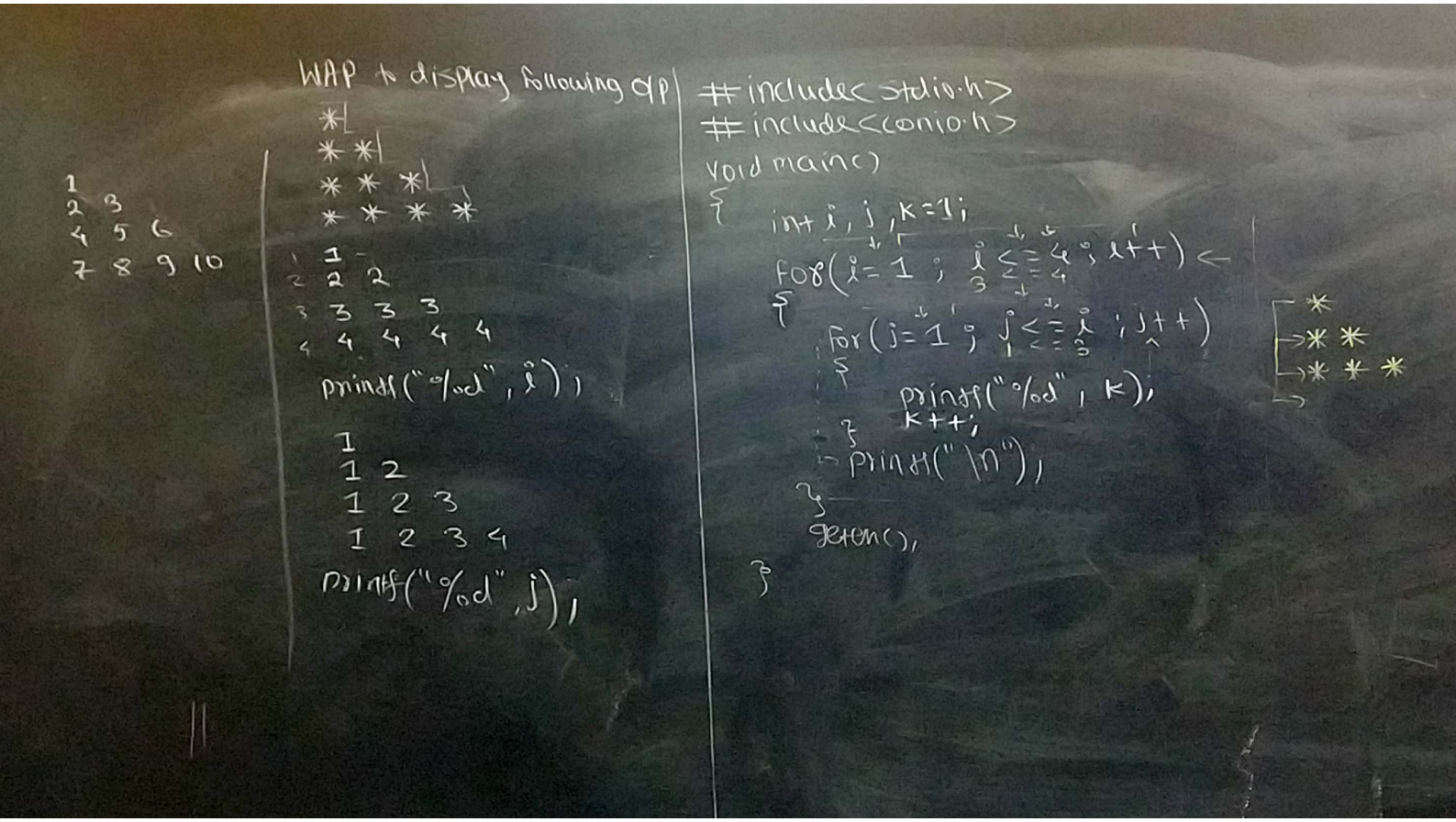
Enter a number
5

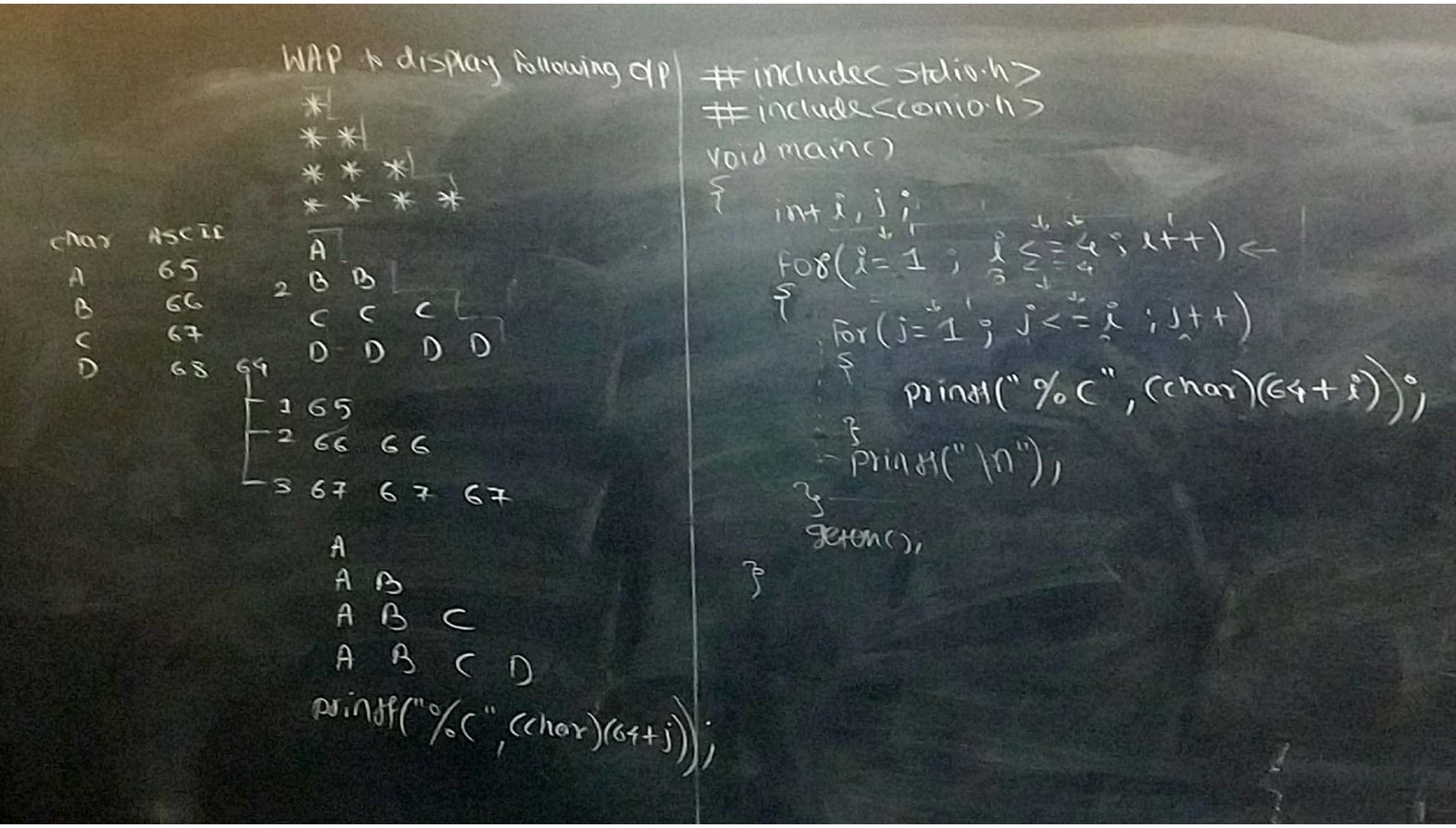
```
#include<stdio.h>
#include<conio.h>
void main()
{
  int n, i, flag=0;
  printf("Enter a number\n");
  scanf("%d", &n);
  for(i=2; i<=n/2; i++)
  {
    if(n%i==0)
    {
      flag=1;
      break;
    }
  }
  if(flag==0)
    printf("Number is prime\n");
  else
    printf("Number is not prime\n");
  getch();
}
```



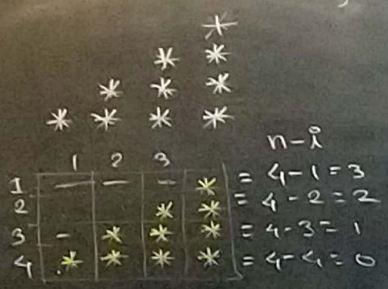






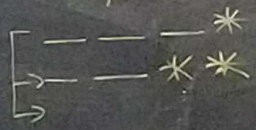


WAP to display following o/p




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

o/p



WAP to display following o/p

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

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

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

```
printf("%c", (char)(64+i));
```

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

o/p

```
    *
  * *
```

WAP to display following o/p

```
1 _ _ _ *
2 _ _ * * *
3 _ * * * * *
4 * * * * * * *
```

2-1
3-1=2
4-1

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

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

printf("%c", (char)(64+1));

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

o/p

```
_ _ _ *
_ _ * * *
_ * * * * *
```

WAP to display following o/p

```
    1
  1 2 A
1 2 3 A B
1 2 3 4 A B C
```

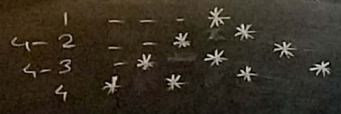
```
      A
    A B
  A B C
A B C D
```

```
printf("%c", (char)(64+j));
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int i, j;
  for(i=1; i<=4; i++)
  {
    for(j=1; j<=4-i; j++)
      printf(" ");
    for(j=1; j<=i; j++)
      printf("%d", j);
    for(j=1; j<=i-1; j++)
      printf("%c", (char)(64+j));
    printf("\n");
  }
  getch();
}
```

o/p

WAP to display following o/p



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

o/p

WAP to display following o/p

odd
1 1
2 2 1
3 1 2 3
Even
4 3 2 1

```
void main()
{ int i, j,
  for(i=1; i<=4; i++)
  {
    if( i%2 != 0)
    {
      for(j=1; j<=i; j++);
      {
        printf("%d", j);
      }
    }
    else
    {
      for(j=i; j>=1; j--);
      {
        printf("%d", j);
      }
    }
    printf("\n");
  }
getch();
}
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int i, j;
  for(i=1; i<=4; i++)
  {
    for(j=1; j<=4-i; j++)
      printf(" ");
    for(j=1; j<=i; j++)
      printf("%c", (char)(64+i));
    printf("\n");
  }
);
```

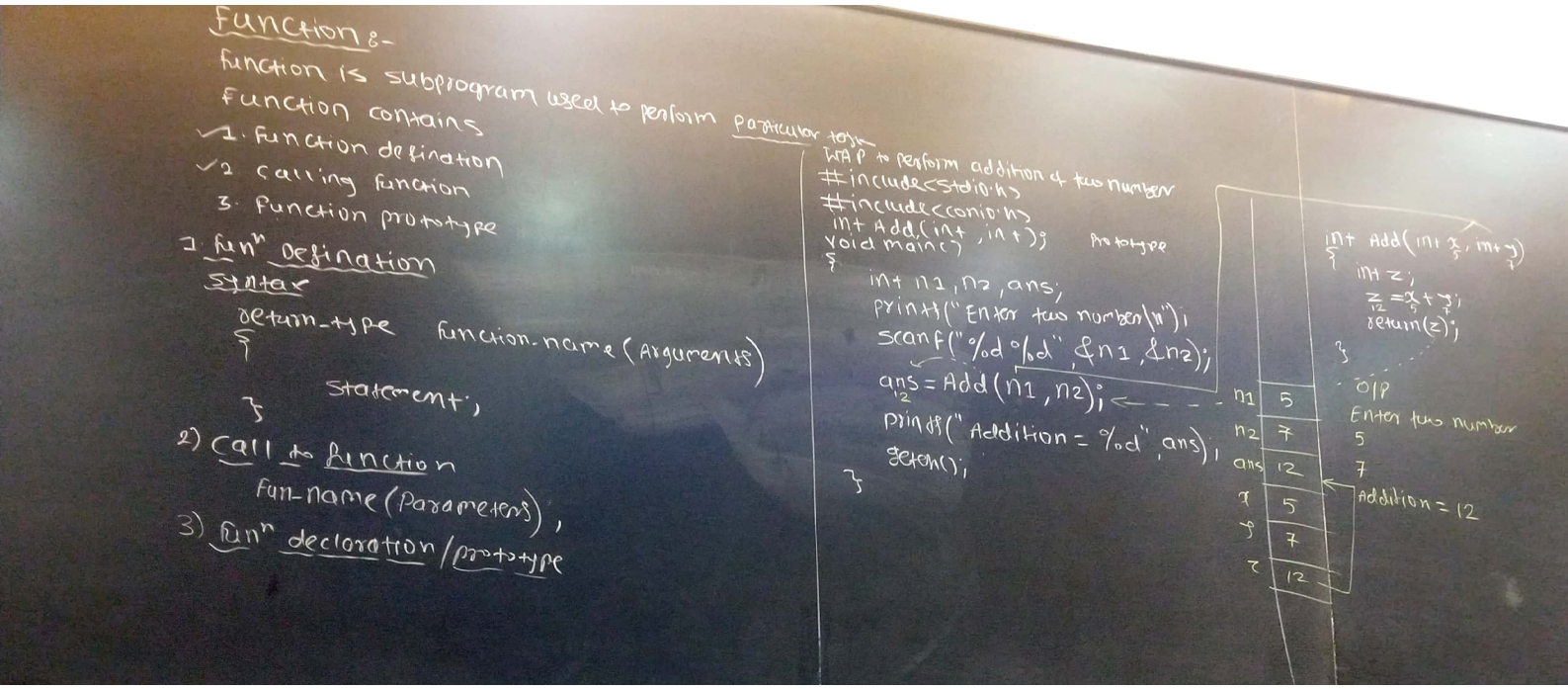
o/p

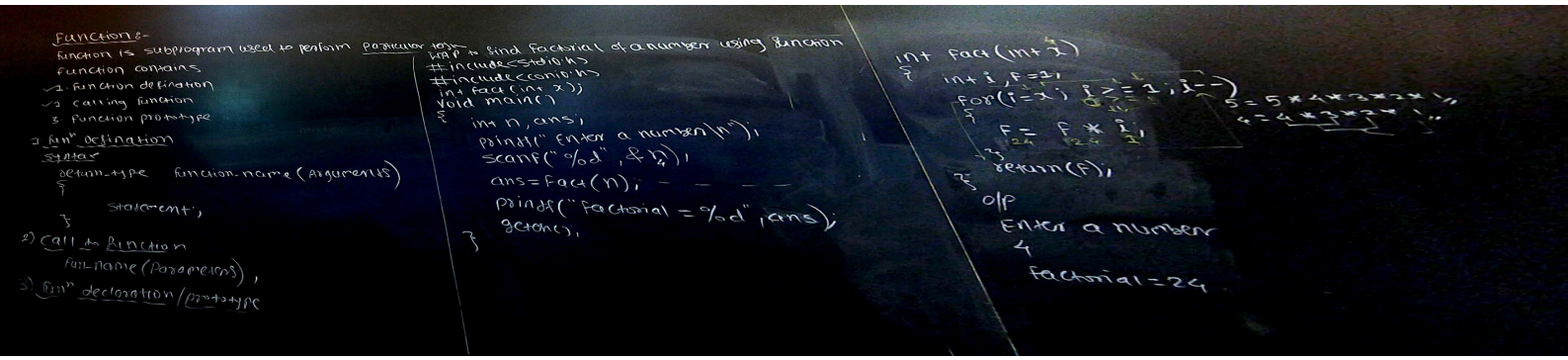
WAP to display following o/p

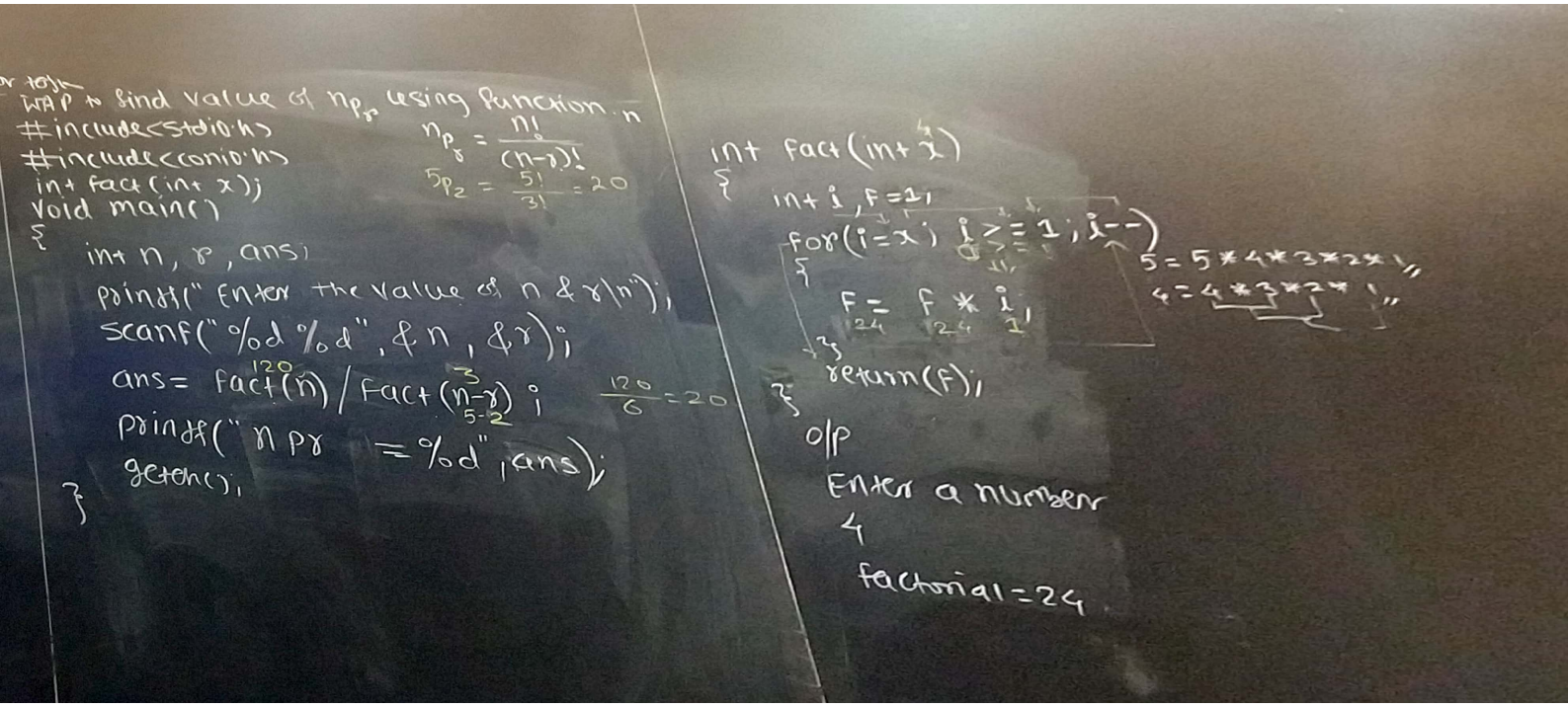
```
- - - - A 1
      C B 2
    F E D 3
  J I H G 4
O N M L K 5
```

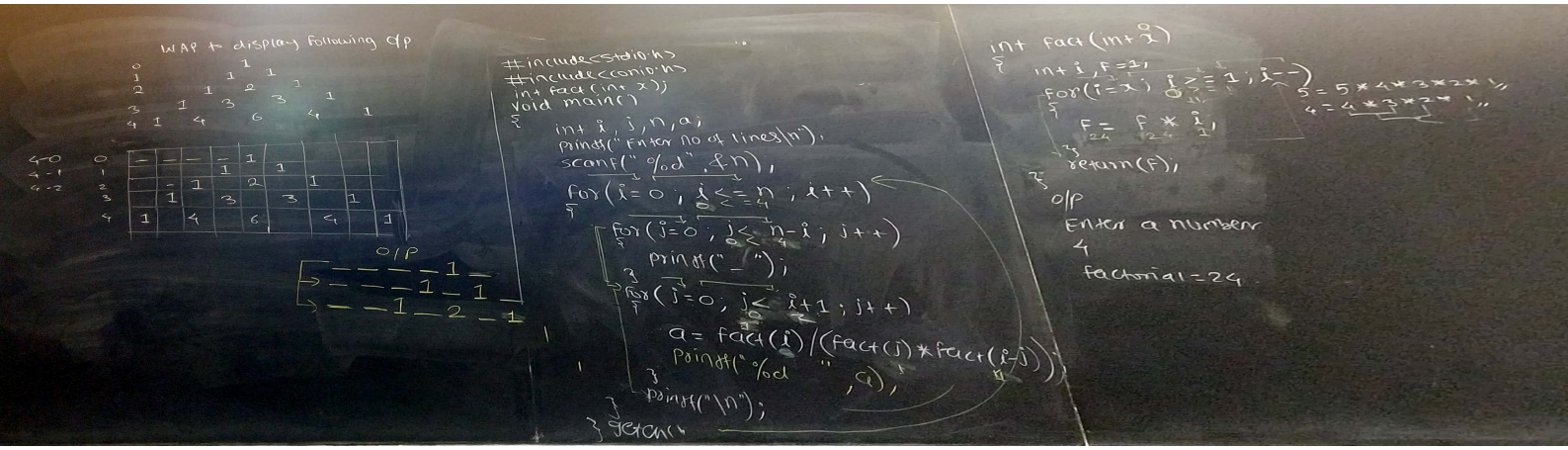
```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, j, k=0, m;
    for(i=1; i<=5; i++)
    {
        k=k+i;
        m=k;
        for(j=1; j<=5-i; j++)
            printf(" ");
        for(j=1; j<=i; j++)
        {
            printf("%c", (char)(64+m));
            m--;
        }
        printf("\n");
    }
```

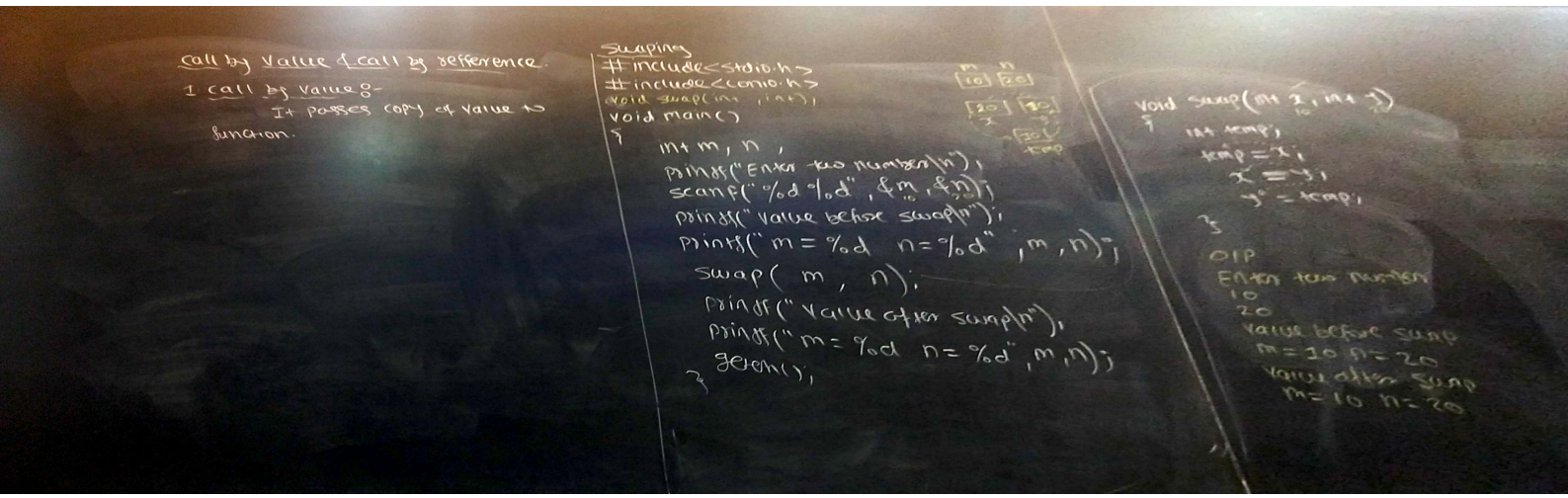
```
- - - - A
- - - C B
```

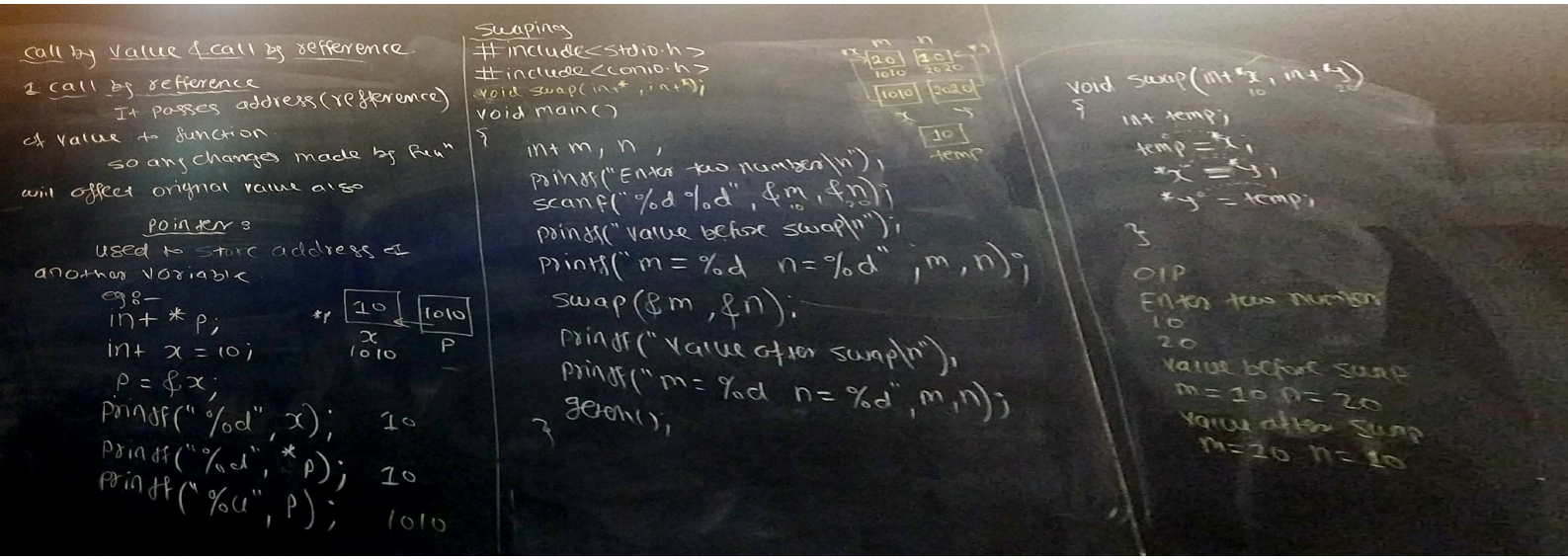


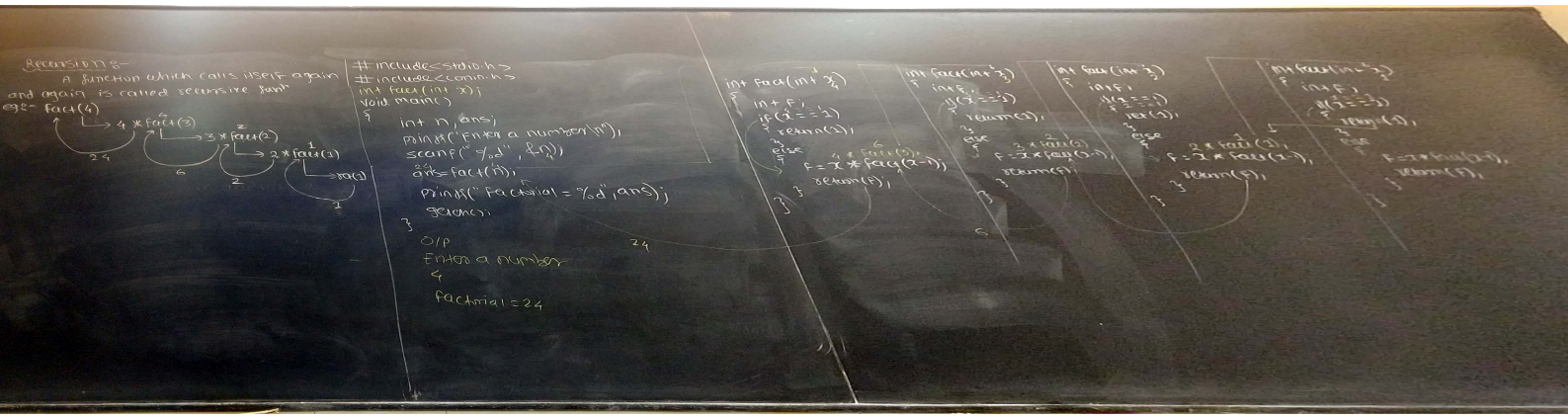


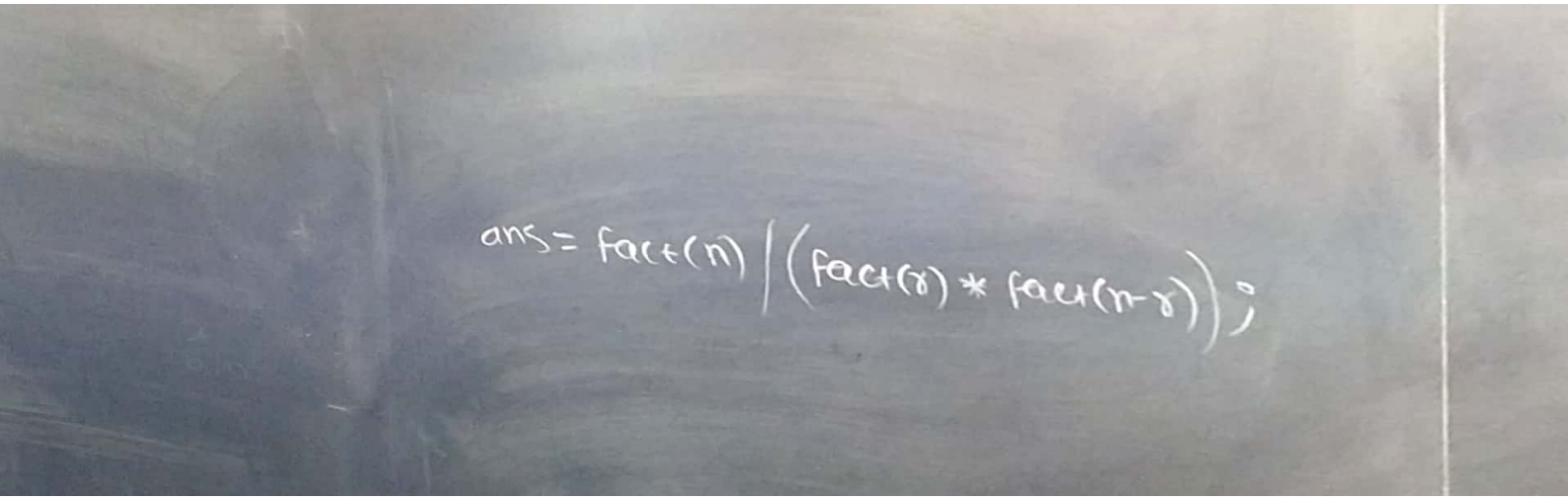


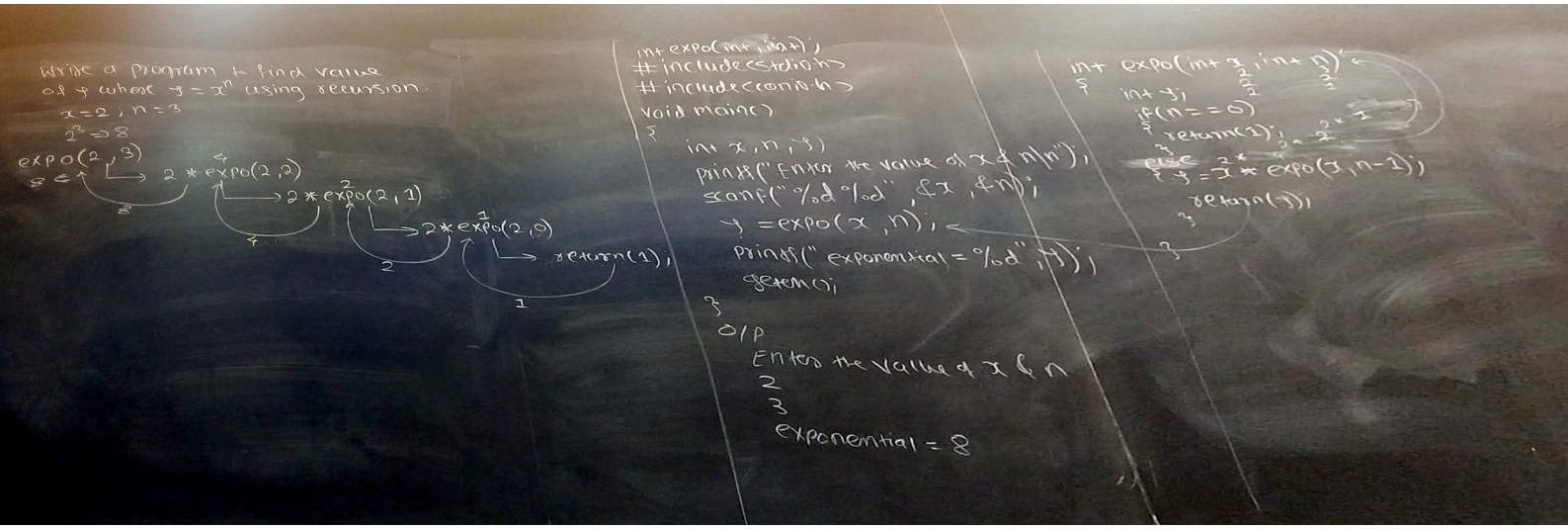












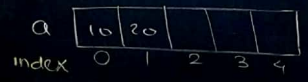
Array:- array is used to store collection of values of similar data type

Syntax

```
data_type array_name[size];
```

eg:-

```
int a[5];
```



Starting index = 0

Last index = n-1

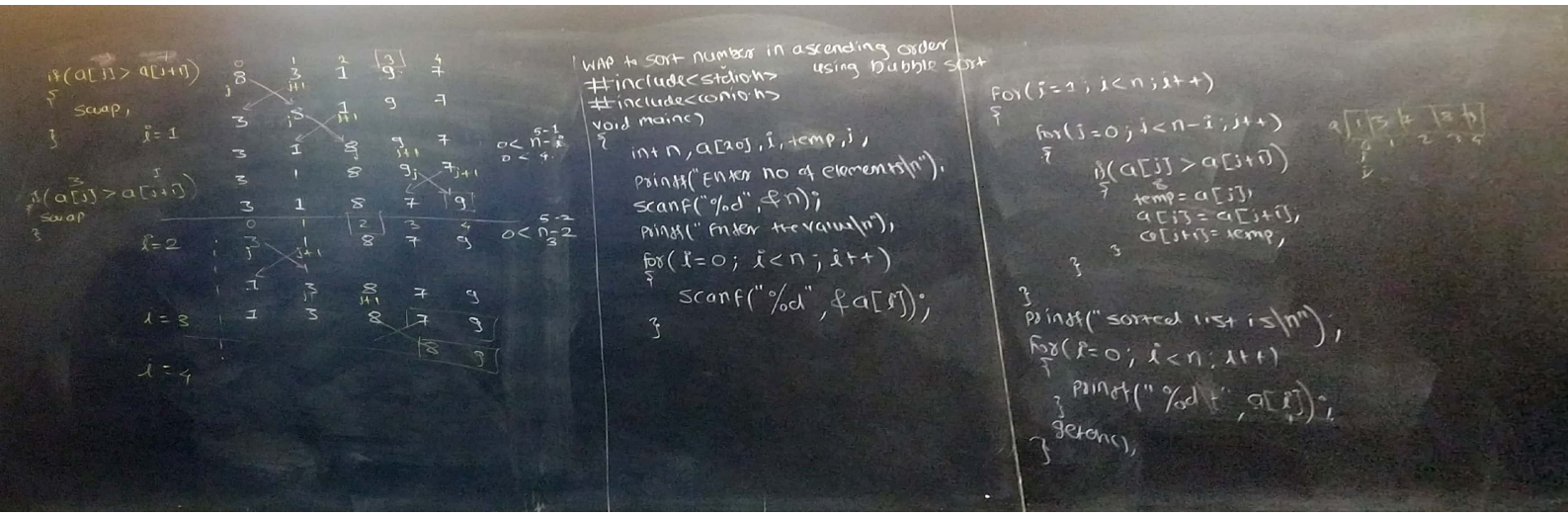
Initilize array

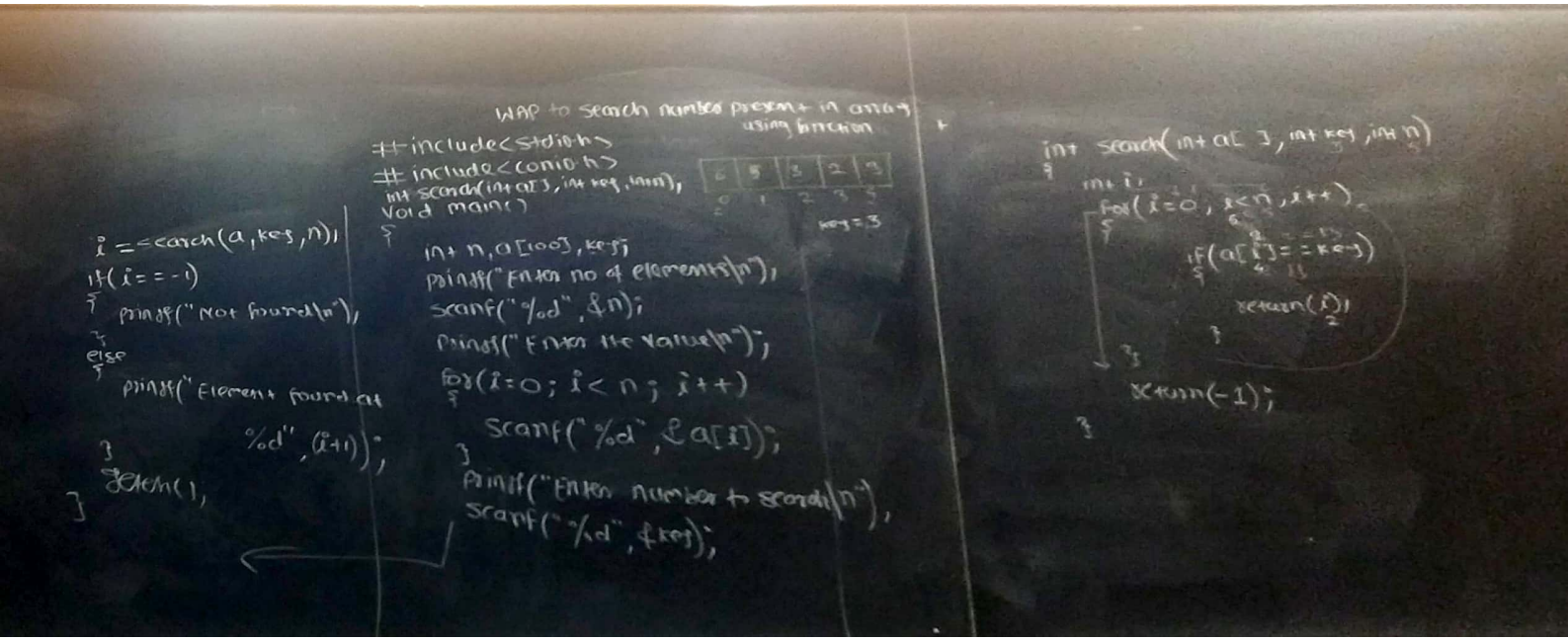
1) `int a[] = {10,20,30,40,50};`
2) `int a[5];`
   `a[0] = 10;`
   `a[1] = 20;`

WAP to find largest element present in array

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int n, a[20], i, max;
  printf("Enter no of elements\n");
  scanf("%d", &n);
  printf("Enter the value\n");
  for(i=0; i<n; i++)
    scanf("%d", &a[i]);
}
```


```
max = a[0]
if( a[4] > max)
  max = a[3];
max = a[0];
for(i = 1 ; i< n ; i++)
{
    if(a[i] > max)
      max = a[i];
}
printf("Largest number = %d", max);
getch();
}
```





Two Dimensional arrays

Syntax

data_type array_name[rowsize][colsize];

eg:

```
int a[3][3];
```

```
a[0][0] = 10;
a[0][1] = 20;
a[0][2] = 30;
```

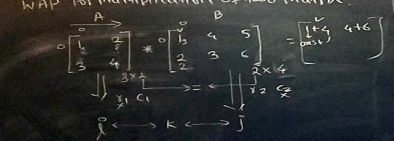
WAP for addition two matrix of size m x n

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int m,n,a[10][10],b[10][10],c[10][10],i,j;
  printf("Enter no of rows & columns");
  scanf("%d %d", &m, &n);
  printf("Enter the value of matrix-A");
  for(i=0; i<m; i++)
  {
    for(j=0; j<n; j++)
    {
      scanf("%d", &a[i][j]);
```

```
printf("Enter the value of matrix B\n");
for(i=0; i<m; i++)
{
  for(j=0; j<n; j++)
    scanf("%d", &b[i][j]);
}
for(i=0; i<m; i++)
{
  for(j=0; j<n; j++)
    c[i][j] = a[i][j] + b[i][j];
}
```

```
printf("Addition of two matrix\n");
for(i=0; i<m; i++)
{
  for(j=0; j<n; j++)
    printf("%d\t", c[i][j]);
  printf("\n");
}
getch();
}
```

WAP for multiplication of two matrix.



$c[i][j] = c[i][j] + a[i][k] * b[k][j]$

$= (1 * 1) +$

$ans = ans + 1;$

$c[i][j]$

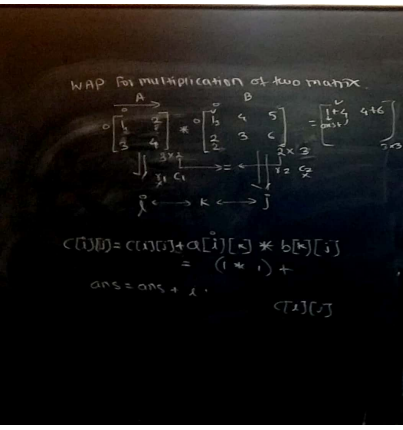
WAP for addition two matrix of size m x n

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int r1, c1, r2, c2, a[10][10], b[10][10], c[10][10], i, j, k;
    printf("Enter no of rows & columns of mat-a\n");
    scanf("%d %d", &r1, &c1);
    printf("Enter the value of matrix-a\n");
    for(i=0; i<r1; i++)
        for(j=0; j<c1; j++)
            scanf("%d", &a[i][j]);
    printf("no of rows & columns of matrix-b\n");
    scanf("%d %d", &r2, &c2);
```

```
printf("Enter the value of matrix b\n");
for(i=0; i<r2; i++)
    for(j=0; j<c2; j++)
        scanf("%d", &b[i][j]);
for(i=0; i<r1; i++)
    for(j=0; j<c2; j++)
    {
        c[i][j]=0;
        for(k=0; k<c1; k++)
            c[i][j]=c[i][j]+a[i][k]*b[k][j];
    }
```

```
printf("Addition of two matrix\n");
for(i=0; i<r1; i++)
{
    for(j=0; j<c1; j++)
        printf("%d\t", c[i][j]);
    printf("\n");
}
getch();
}
```

WAP for multiplication of two matrix.



$c[i][j] = c[i][j] + a[i][k] * b[k][j]$

$= (i * i) +$

ans = ans + i.

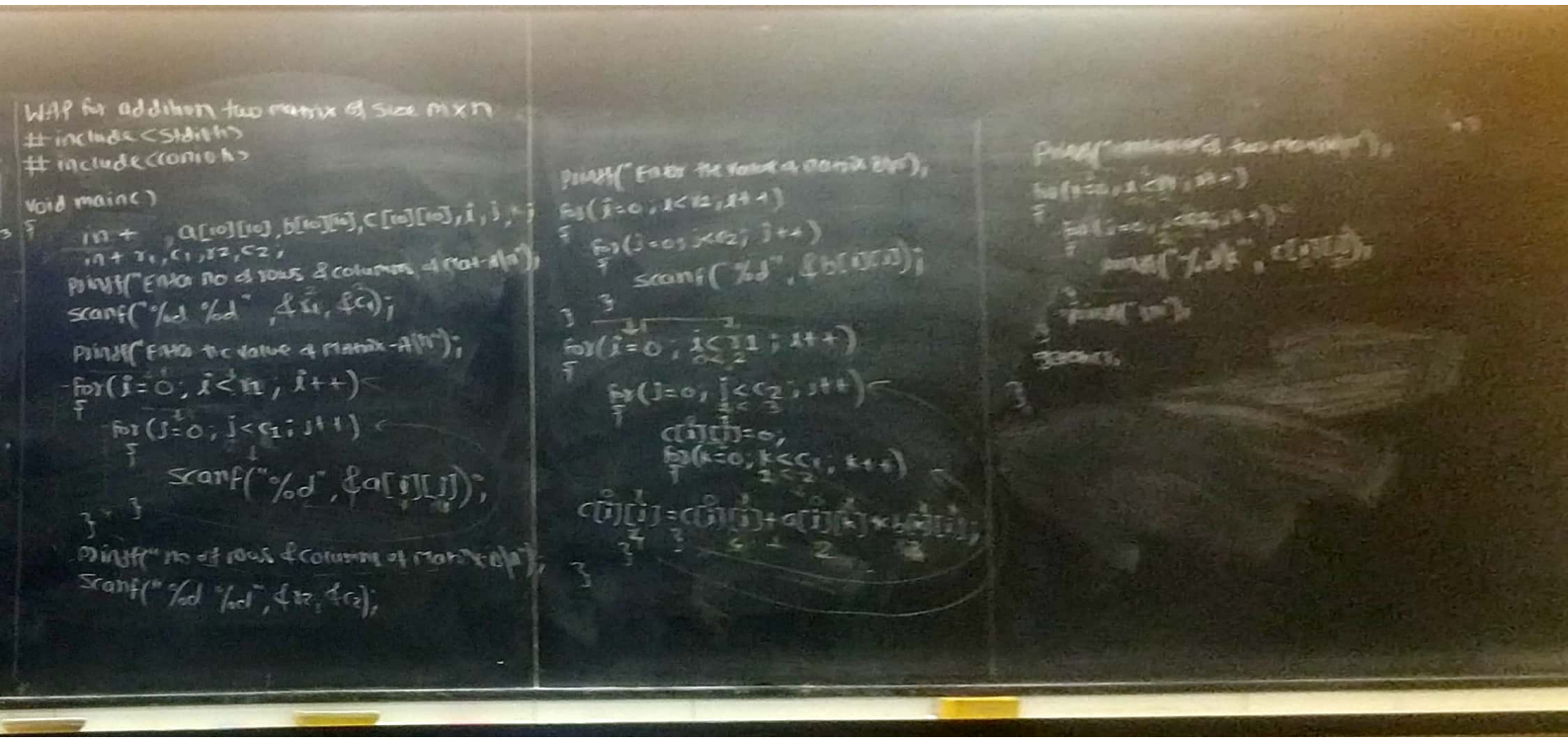
c[i][j]

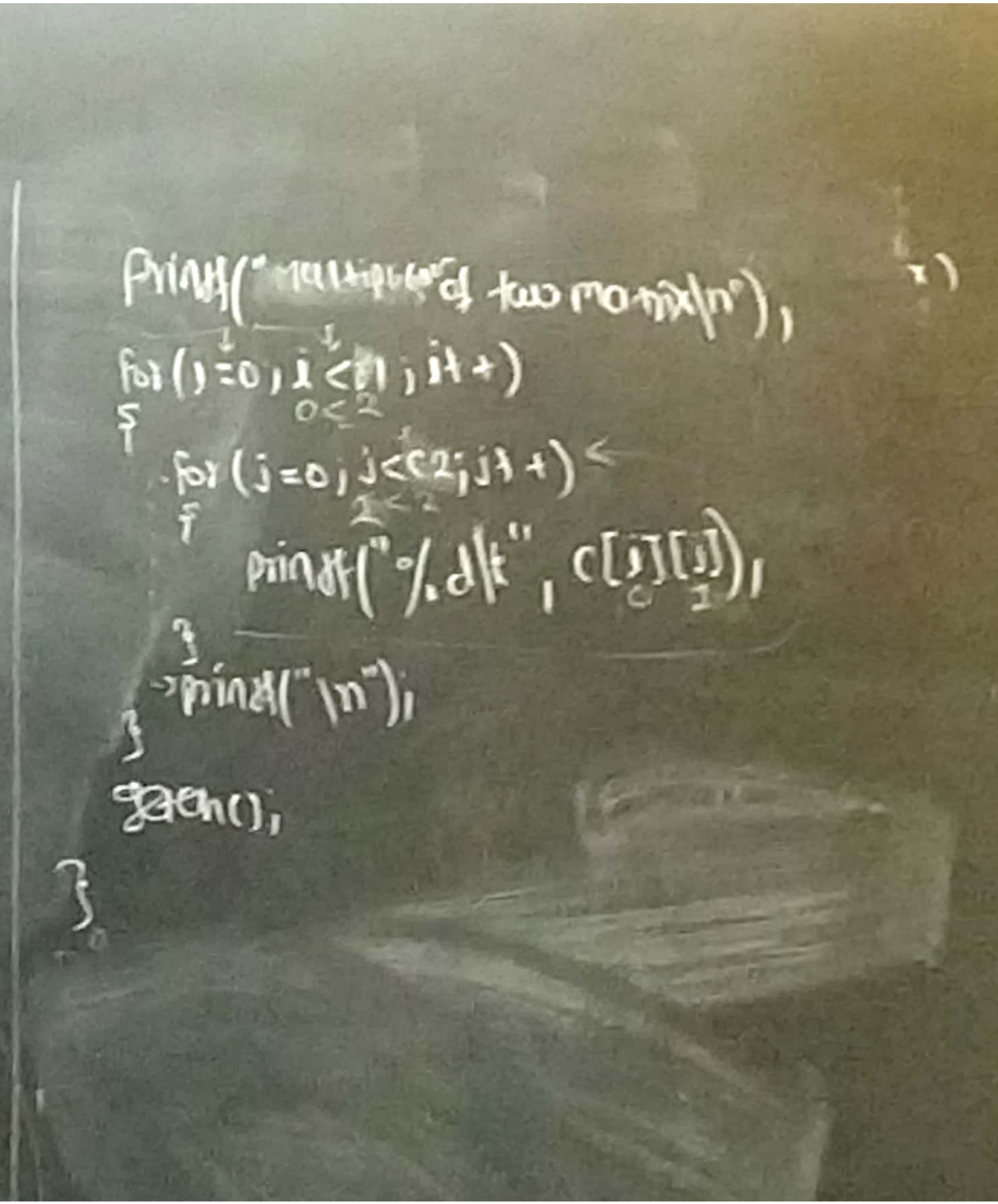
WAP for addition two matrix of size m x n

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int r1, c1, r2, c2,
printf("Enter no of rows & columns of matrix-A");
scanf("%d %d", &r1, &c1);
printf("Enter the value of matrix-A");
for(i=0; i<r1; i++)
{
  for(j=0; j<c1; j++)
  {
    scanf("%d", &a[i][j]);
  }
}
printf("no of rows & columns of matrix-B");
scanf("%d %d", &r2, &c2);
```

```
printf("Enter the value of matrix-B");
for(i=0; i<r2; i++)
{
  for(j=0; j<c2; j++)
    scanf("%d", &b[i][j]);
}
for(i=0; i<r1; i++)
{
  for(j=0; j<c2; j++)
  {
    c[i][j]=0;
    for(k=0; k<c1; k++)
    c[i][j]=c[i][j]+a[i][k]*b[k][j];
  }
}
```

```
printf("
for(i=0; i<r1; i++)
{
  for(j=0; j<c2; j++)
    printf("%d", c[i][j]);
  printf("\n");
}
getch();
}
```





Write a program to multiply two matrix by using following function
1) accept matrix 2) matmultiply 3) Display

```
if(c1 == r2)
{
    matmultiply(a, b, c, r1, c2, c1);
    Display(c, r1, c2);
}
else
{
    printf("Multiplication not possible\n");
}
getch();
}
```

WAP for addition two matrix of size m x n

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10][10], b[10][10], c[10][10], i, j, k;
    int r1, c1, r2, c2;
    printf("Enter no of rows & columns of mat-A\n");
    scanf("%d %d", &r1, &c1);
    printf("Enter the value of Matrix-A\n");
    accept(a, r1, c1);
    printf("Enter no of rows & col of mat-B\n");
    scanf("%d %d", &r2, &c2);
    printf("Enter the value of Matrix-B\n");
    accept(b, r2, c2);
```

```
void accept(int x[10][10], int m, int n)
{
    int i, j;
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
            scanf("%d", &x[i][j]);
    }
}
void matmultiply(int a[10][10], int b[10][10],
                 int c[10][10], int r1, int r2, int c1)
{
    int i, j, k;
    for(i=0; i<r1; i++)
    {
        for(j=0; j<c2; j++)
        {
            c[i][j]=0;
            for(k=0; k<c1; k++)
                c[i][j]= c[i][j] + a[i][k] * b[k][j];
        }
    }
```

```
void display(int x[10][10], int m, int n)
{
    int i, j;
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
            printf("%d\t", c[i][j]);
        printf("\n");
    }
```

String :-

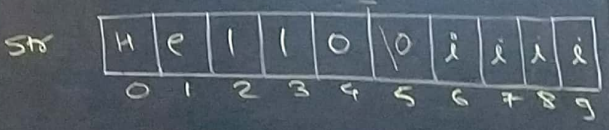
It is array of character.

Syntax

```
char string-name [size];
```

eg:-

```
char str[10];
```




```
char str[] = "Hello";
```

'\0' is used to terminate string.

String Handling function

1. Strlen() :- This fun^n is used to find length of a string

Syntax

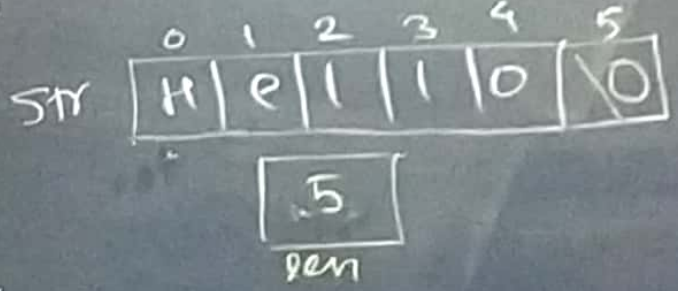
```
int strlen(string);
```

eg:-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char str[20];
    int len;
    printf("Enter a string\n");
    gets(str);
    len=strlen(str);
    printf("Length of a string %d", len);
    getch();
}
```

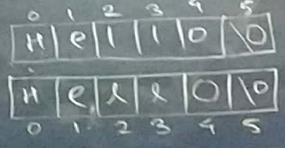
WAP to Find the length of String without using "string.h".

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char str[20];
    int len=0, i=0;
    printf("Enter a string\n");
    gets(str);
    while(str[i]!='\0')
    {
        len++;
        i++;
    }
    printf("length of a string=%d", len);
    getch();
}
```



WAP to Find to Copy String without using "string h".

```
#include<stdio.h>
#include<conio.h>
void main()
{
  char s[20], d[20];
  int len=0, i=0;
  printf("Enter a string\n");
  gets(s);
  while(s[i]!='\0')
  {
    d[i]=s[i];
    i++;
  }
  d[i]='\0';
  printf("String after copy=%s",d);
  getch();
}
```



String Handling Function

1. strcpy(); This function is used to copy one string into another string.

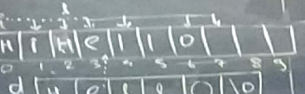
Syntax

strcpy(dest.string, source.string);

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
  char s[20], d[20];
  printf("Enter a string\n");
  gets(s);
  strcpy(d, s);
  printf("String after copy %s", d);
  getch();
}
```

WAP to Find to copy string without using "string h".

```
#include<stdio.h>
#include<conio.h>
void main()
{
  char s[20], d[20], d
  int i=0, j=0;
  printf("Enter a first string\n");
  gets(s);
  printf("Enter a second string\n");
  gets(d);
  while(s[i]!='\0')
  {
    i++;
  }
  while(d[j]!='\0')
  {
    s[i]=d[j];
    i++;
    j++;
  }
  s[i]='\0';
  printf("string after concat %s", s);
  getch();
}
```



String Handling function

3. Strcat(): This fun^n is used to concat (join) two string together.

Syntax

Strcat(String1, String2);

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
  char s[20], d[20];
  printf("Enter first string\n");
  gets(s);
  printf("Enter second string\n");
  gets(d);
  strcat(s, d);
  printf("String after concat = %s", s);
  getch();
}
```

WAP to check entered string is palindrome or not

```
#include<string.h>
#include<stdio.h>
#include<conio.h>
void main()
{
  char str[20];
  int len, i, flag=0;
  printf("Enter a string\n");
  gets(str);
  len = strlen(str);
  for(i=0; i<(len/2); i++)
  {
    if(str[i] != str[len-1-i])
    {
      flag=1;
      break;
    }
  }
  if(flag==0)
  {
    printf("string is palindrome\n");
  }
  else
  {
    printf("string is not palindrome\n");
  }
  getch();
}
```

str: N I T I N \0 (indices 0 1 2 3 4 5)

```
if(i==0)
{
  printf("strings are equal\n");
}
else
{
  printf("strings are not equal\n");
}
getch();
}
```

* strcmp():- This funth is used to compare two string

Syntax:-

```
int strcmp(string1, string2);
```

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[20], d[20];
Printf("enter first string\n");
gets(s);
Printf("enter second string\n");
gets(d);
i= strcmp(s,d)
if(i==0)
{
Printf("strings are equal\n");
}
Else
{
Printf("strings r not equal\n");
}
getch();
}
```

Structure :-

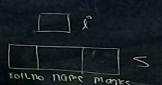
Structure is user defined datatype which can store mix type of data.

Syntax

```
struct structure_name
{
    Datatype Variable1;
    Datatype Variable2;
    .....
};
```

WAP to create Structure Student having data member roll_no, name & marks. accept the data for one student & display.

```
#include<stdio.h>
#include<conio.h>
struct student
{
    int roll_no;
    char name[20];
    float marks;
};
```




```
void main()
{
    int i;
struct student s;
printf("Enter roll number\n");
scanf("%d",&s.roll_no);
printf("Enter student name\n");
gets(s.name);
printf("Enter marks\n");
scanf("%f",&s.marks);
printf("Student details are\n");
printf("Roll No = %d\n Name = %s\n Marks=%f",s.roll_no,s.name,s.marks);
getch();
}
```

WAP to create nested structure to store details of n student with name, rollno & marks of phy, chem, maths. display the details of student in tabular form as RollNo Name Total marks

```
#include<stdio.h>
#include<conio.h>
struct student
{
  int rollno;
  char name[20];
  struct
  {
    int phy;
    int chem;
    int maths;
  }marks;
  int total;
};
```

	rollno	name	marks			total
			P	C	M	
s[0]	roll	name				
1						
2						
⋮						
99						

```
void main()
{
  struct student s[100];
  int n, i;
  printf("Enter no of student\n");
  scanf("%d", &n);
  for(i=0; i<n; i++)
  {
    printf("Enter roll no\n");
    scanf("%d", &s[i].rollno);
    printf("Enter name\n");
    gets(s[i].name);
    printf("Enter marks of phy, chem, maths\n");
    scanf("%d %d %d", &s[i].marks.phy, &s[i].marks.chem,
                      &s[i].marks.maths);

    s[i].total = s[i].marks.phy + s[i].marks.chem + s[i].marks.maths;
  }
```

```
  printf("Rollno\t Name\t Total marks\n");
  for(i=0; i<n; i++)
  {
    printf("%d\t %s\t %d\n", s[i].rollno, s[i].name, s[i].total);
  }
  getch();
}
```

WAP to create structure employee having data member empid, ename, salary accept details of 10 employee & display details of those employee whose salary greater than 50000 or equal to

Sat 8 2PM
Files & pointer

```
#include<stdio.h>
#include<conio.h>
struct employee
{
  int empid;
  char ename[10];
  float sal;
};
```

```
void main()
{
  struct employee e[10];
  int i;
  for(i=0; i<10; i++)
  {
    printf("Enter emp id\n");
    scanf("%d", &e[i].empid);
    printf("Enter name\n");
    gets(e[i].ename);
    printf("Enter salary\n");
    scanf("%f", &e[i].sal);
  }
```

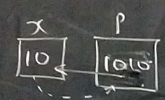
```
printf("EmpId\t EmpName\t Salary\n");
for(i=0; i<10; i++)
{
  if(e[i].sal >= 50000)
    printf("%d\t %s\t %f\n", e[i].empid, e[i].name, e[i].sal);
}
getch();
}
```

Pointer :- Pointer is variable which is used to store the address of another variable

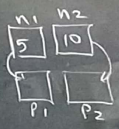
Syntax

data_type *pointer_name;

eg:- int *p;
char *c;
float *ptr;



printf("%d", x); // 10
printf("%d", *p); // 10

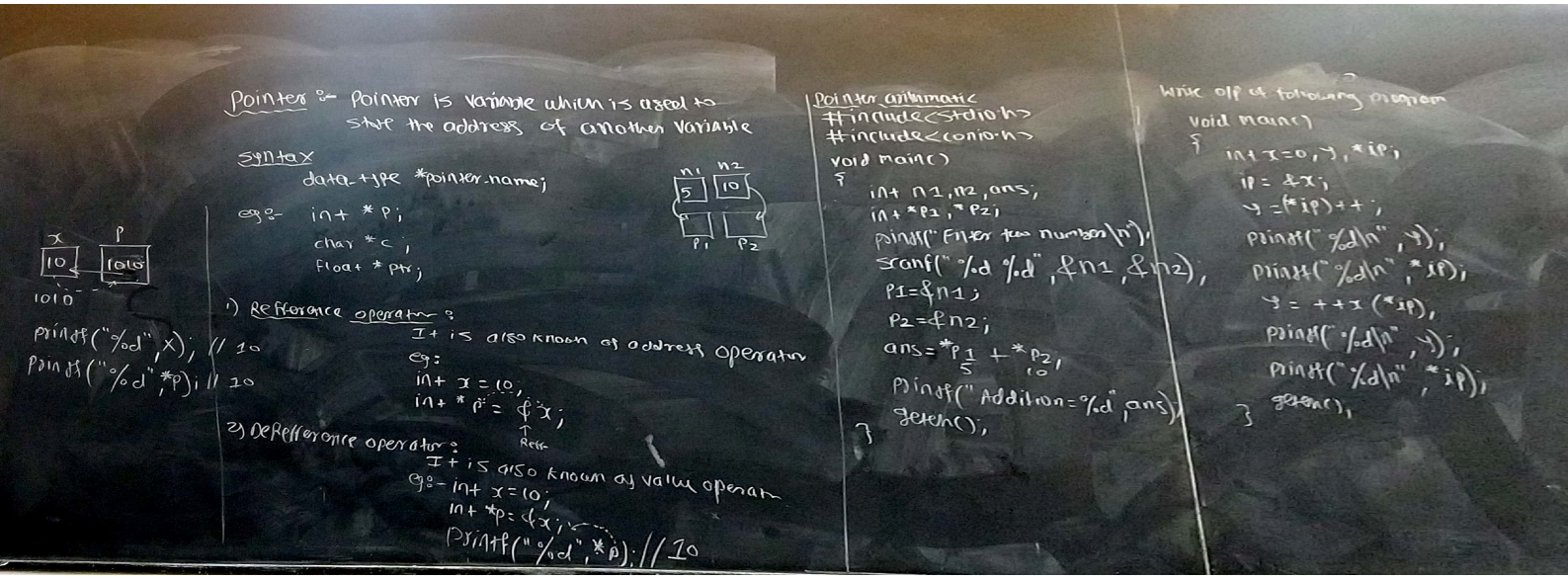


1) **Refferance operator :**
It is also known as address operator
eg:
int x = 10;
int *p = &x;
Reff

2) **Dereferrance operator :**
It is also known as value operator
eg:- int x = 10;
int *p = &x;
Printf("%d", *p); // 10

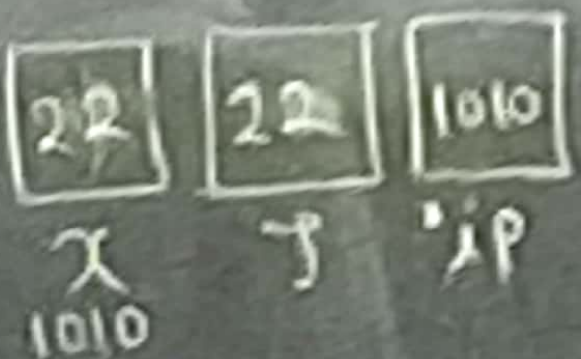
Pointer arithmatic

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n1,n2,ans;
   int *p1,*p2;
   printf("Enter two numbers\n");
   scanf("%d %d",&n1,&n2);
   p1=&n1;
   p2=&n2;
   ans=*p1 + *p2;
   printf("Addition=%d",ans);
   getch();
}
```



Write o/p of following program

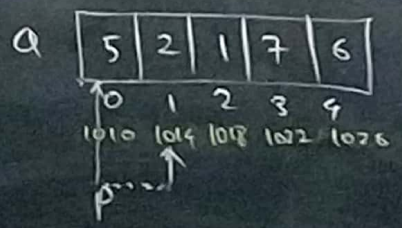
```
void main()
{ int x=20, y, *ip;
  ip = &x;
  y = (*ip)++;
  printf("%d\n", y);      // 20
  printf("%d\n", *ip);    // 21
  y = ++x (*ip);
  printf("%d\n", y);      // 22
  printf("%d\n", *ip);    // 22
} getch();
```



Pointer to array:-

Pointer stores the address of array.

WAP to display sum of array elements using pointer



Pointer

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10], n, i, sum=0, *p;
    printf("Enter no of elements\n");
    scanf("%d", &n);
    printf("Enter the value\n");
    for(i=0; i<n; i++)
    {
      scanf("%d", &a[i]);
    }
    p=a;
    for(i=0, i<n, i++)
    {
      sum = sum + *(p+i)
    }
    printf("Sum = %d", sum);
} getch();
```

Dynamic memory allocation

Memory is allocated dynamically during execution of program these is available in "stdlib.h"

1) malloc() : It is used to allocate memory dynamically with initial value as garbage values.

Syntax :

```
ptr = (typecast*)malloc((size in bytes));
```

eg :-

```
int *ptr;
ptr = (int*)malloc(10 * sizeof(int));
                   4
```

2) calloc() : It is used to allocate memory dynamically with initial value as "zero"

Syntax :

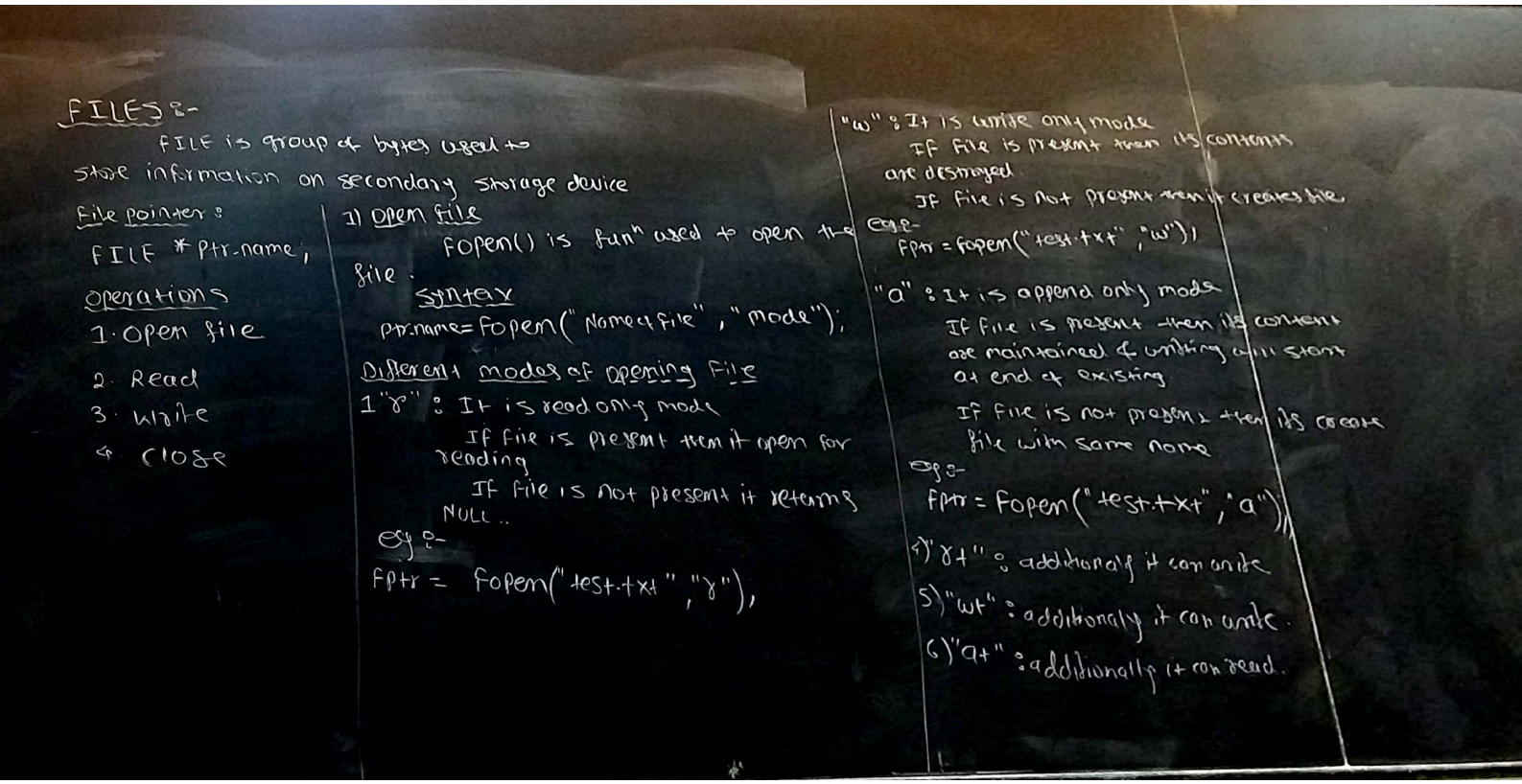
```
ptr = (typecast*)calloc((size in bytes));
```

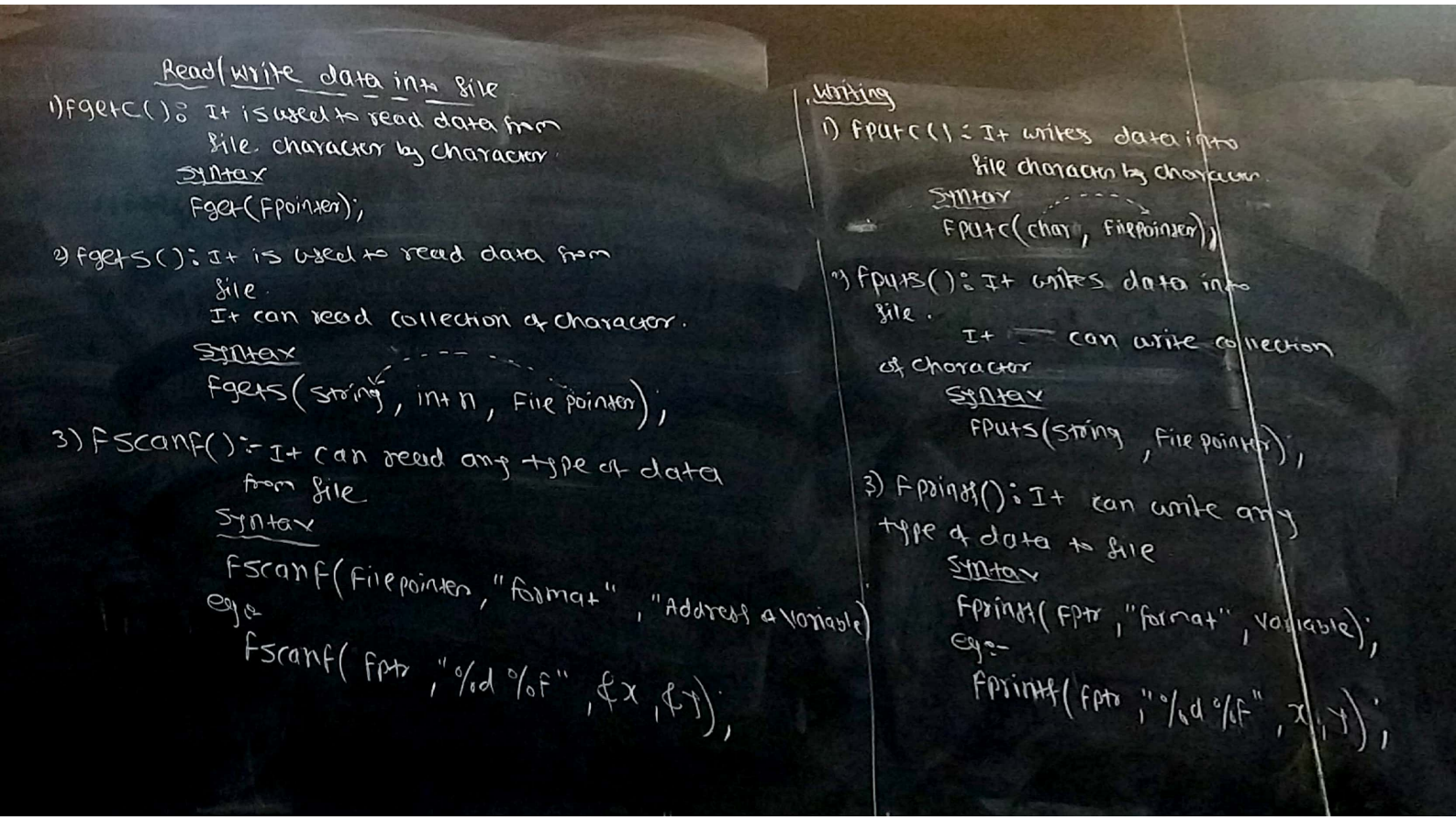
3) free() : It is used to free or deallocate memory allocated dynamically

Syntax

```
free(pointer);
```

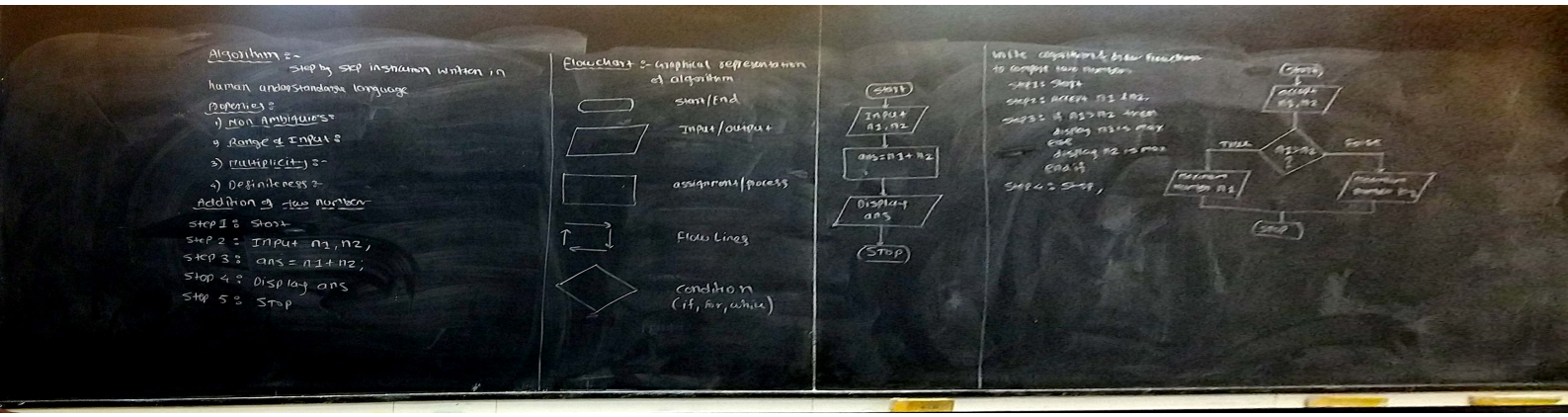
```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, *p, i;
    printf("Enter no of elements\n");
    scanf("%d", &n);
    p=(int*)malloc(n*sizeof(int));
    for(i=0; i<n; i++)
    {
      scanf("%d", (p+i));
    }

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

}getch();
```



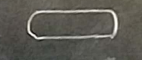


Write a program to read & write data into file.

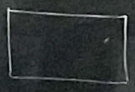
```
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *fptr;
    char ch;
    fptr = fopen("test.txt","w");
    printf("Enter data to write into file once completed
            enter full stop(.)\n");
    while(ch!='.')
    {
        scanf("%c",&ch);
        fputc(ch,fptr);
    }
    fclose(fptr);
    fptr = fopen("test.txt","r");
    while(!feof(fptr))
    {
        ch = fgetc(fptr);
        printf("%c",ch);
    }
    fclose(fptr);
    getch();
}
```

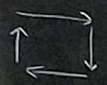


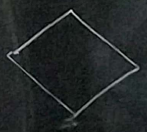
Flowchart :- Graphical representation of algorithm

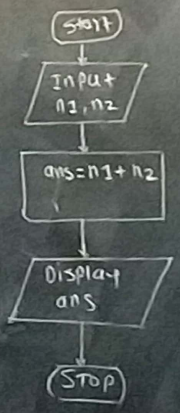

start/End


Input/output


assignment/process


Flow Lines


condition (if, for, while)



Write algorithm & draw flowchart to compare two numbers

Step1 : Start

Step2 : Accept n1 & n2

Step3 : if n1>n2 then
display n1 is max
else
display n2 is max
endif

Step4 : Stop,