Size of intesen, float, double, character datatype.

```
int main()
{
    int i;
    float f;
    printf("size of int = %d \n", sizeof(i));
    printf("size of float = %d \n", sizeof(f));
    return 0;
}
```

এখানে sizeof() operator use করতে হবে printf() এর ভিতর, % %d use করতে হবে integer type এর output terminal এ দেখতে। %d এর পরিবর্তে %f বা %lf use করলে output আসবেনা।

ASCII value input নিয়ে তার character output দেয়। [065 = A, 097 = a]

i) int input নিবে। A ASCII value

ii) scanf() - এ %d (int এর জন্য) use করবে।

iii) printf() - এ %c (character এ convert করতে) use করতে হবে output এর জন্য।

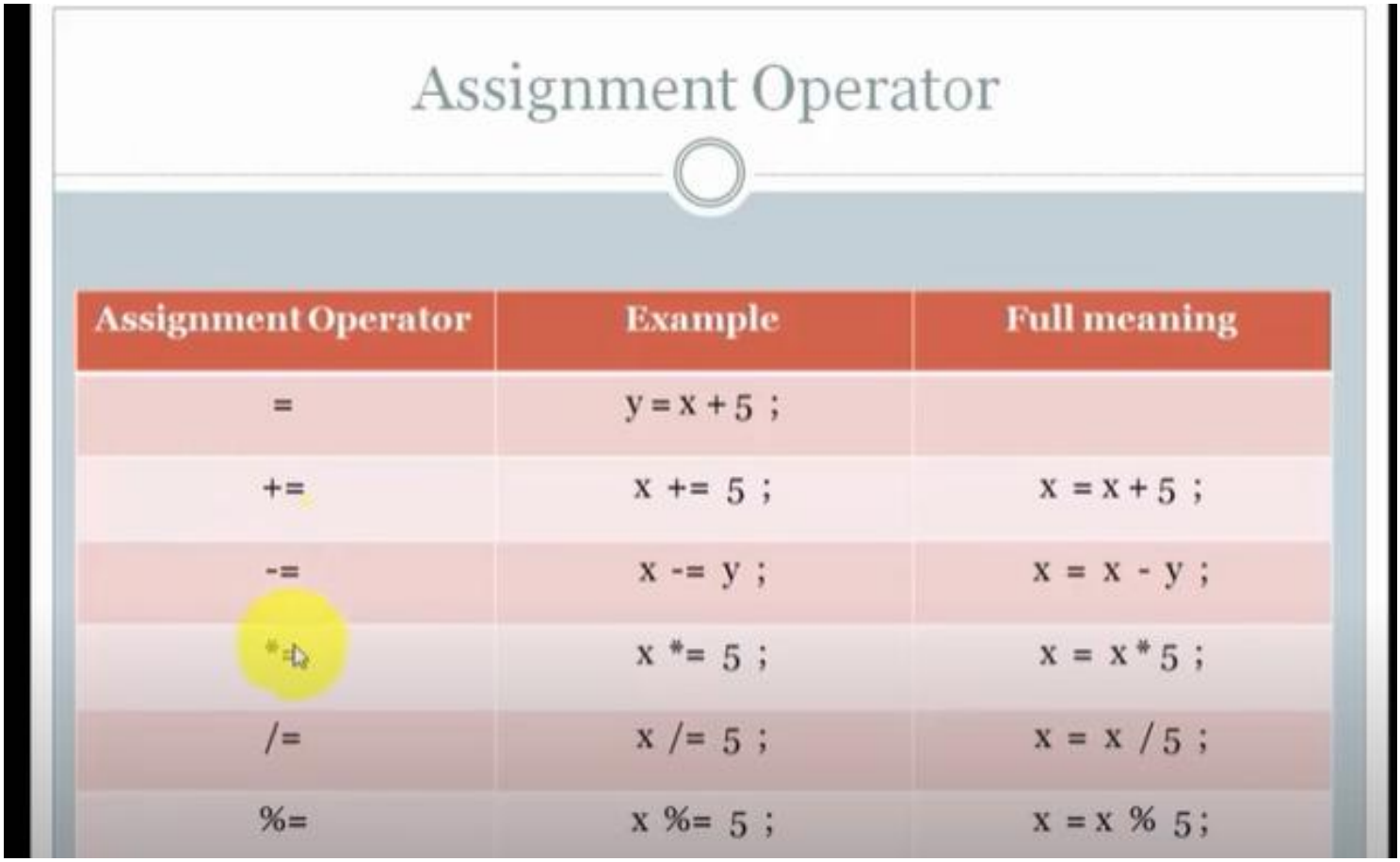
```
int main()
{
    int n;
    printf("Enter ASCII value: ");
    scanf("%d", &n);
    printf("The ASCII character is: %c", n);
    return 0;
}
```

- Character থেকে ASCII code এ convert করতে vice-versa.
- Lowercase to Uppercase convert without library function.

rr

dfsdfs

Assignment Operator



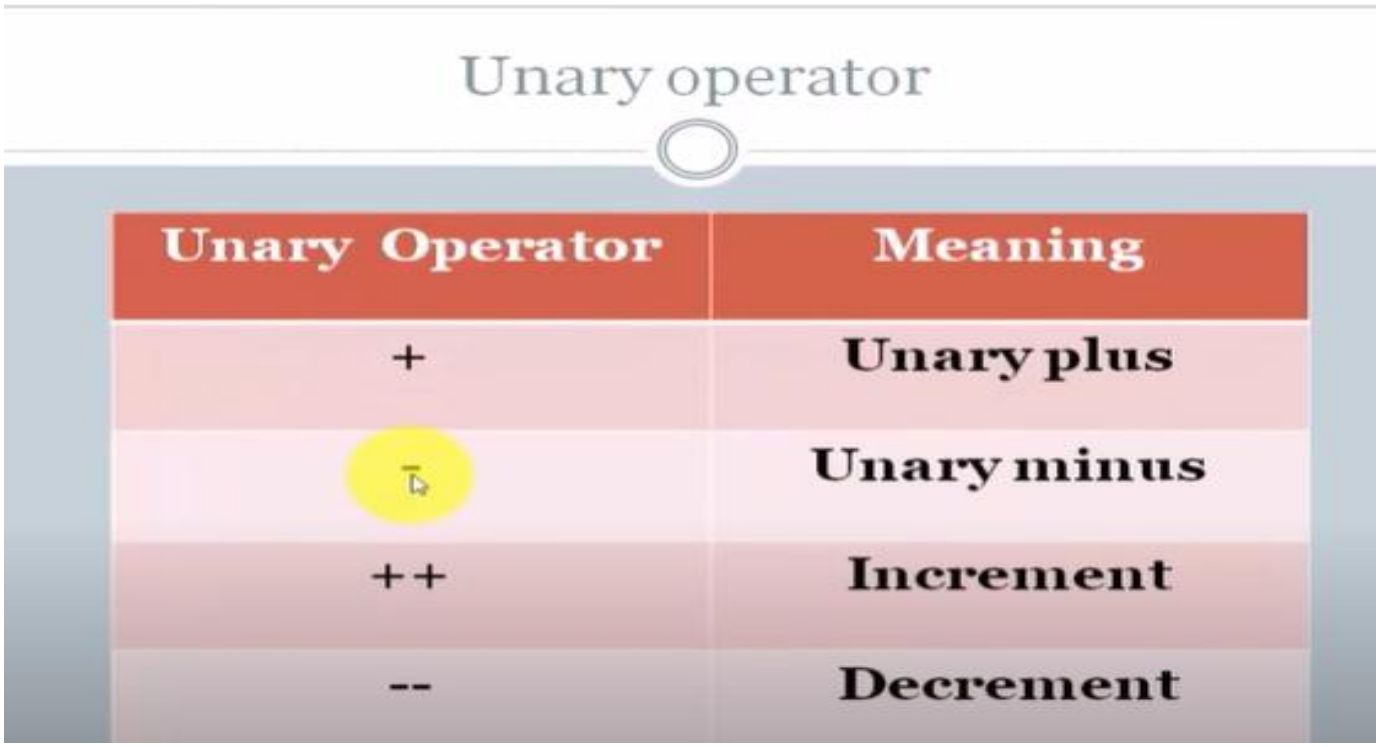
Code:

```c
#include<stdio.h>

int main()

{
    int a=5;

    a+=3; //a=a+3

    printf("the sum is :%d\n\n",a);

    a*=5;

    printf("the multiplication is:%d",a); //a=a*5

    return 0;
}
```

Result:

```
the sum is :8

the multiplication is:40
PS C:\Users\ranga\Documents\anisulcprog.c>
```

Unary Operator

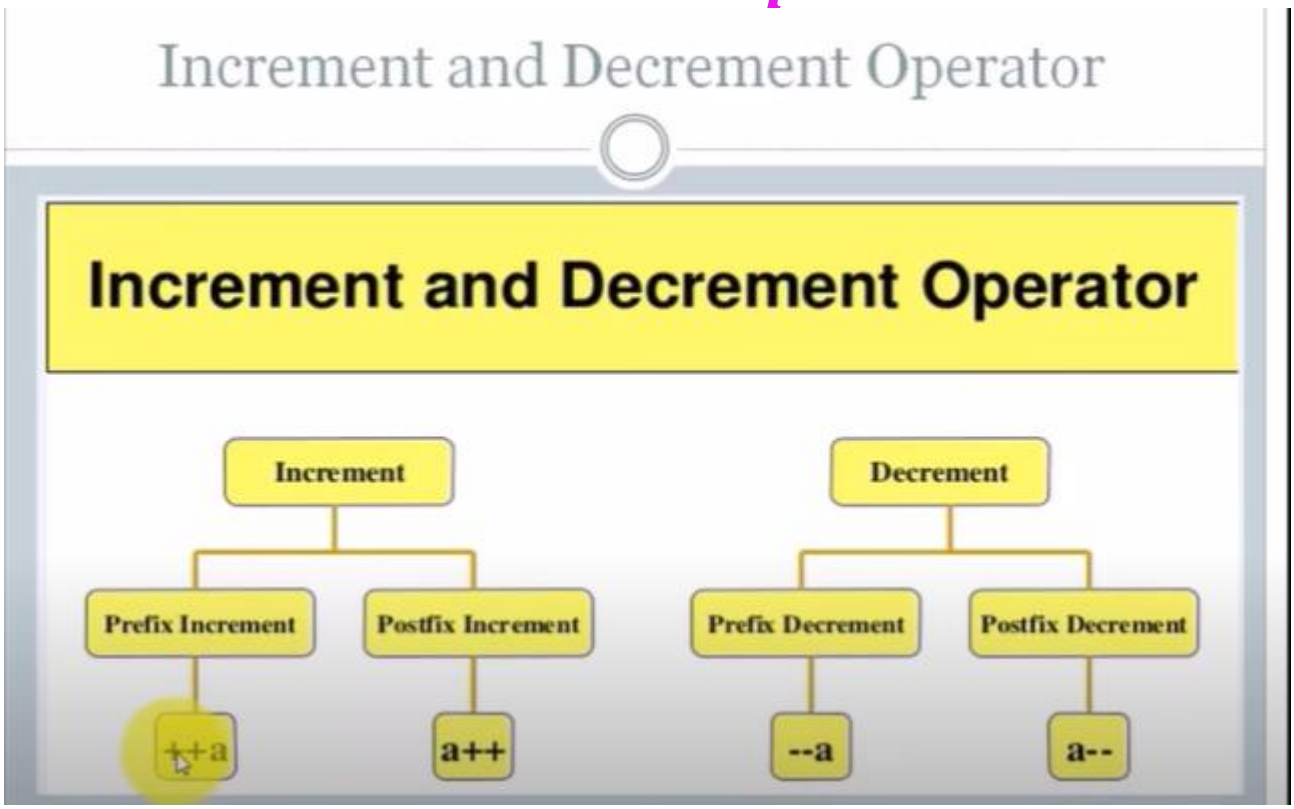


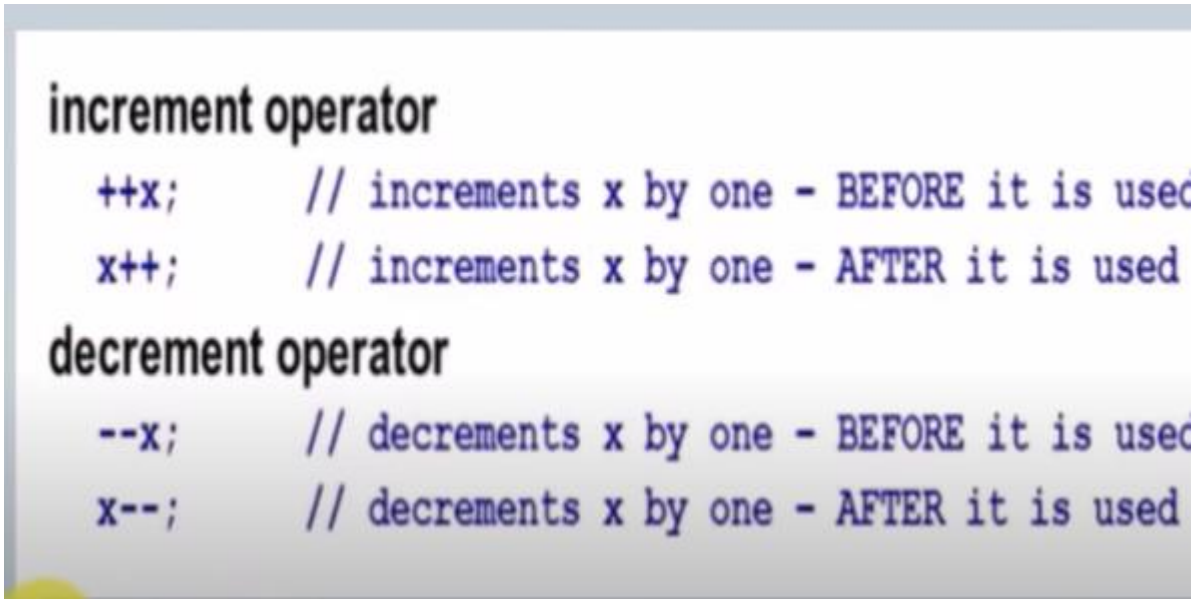
Code:

```c
#include<stdio.h>
int main()
{
    int x=10;
    int result = -x;
    printf("%d\n",result);


    return 0;
}
```

Result:

Increment and Decrement operators





Increment Code:

```c
#include<stdio.h>

int main()

{
    int x=10;

    int y=x++; //y=10

    printf("\n\nx=%d\n",x); //x=11

    printf("y=%d\n\n",y);

    return 0;
}
```

Return:

```
x=11
y=10
```

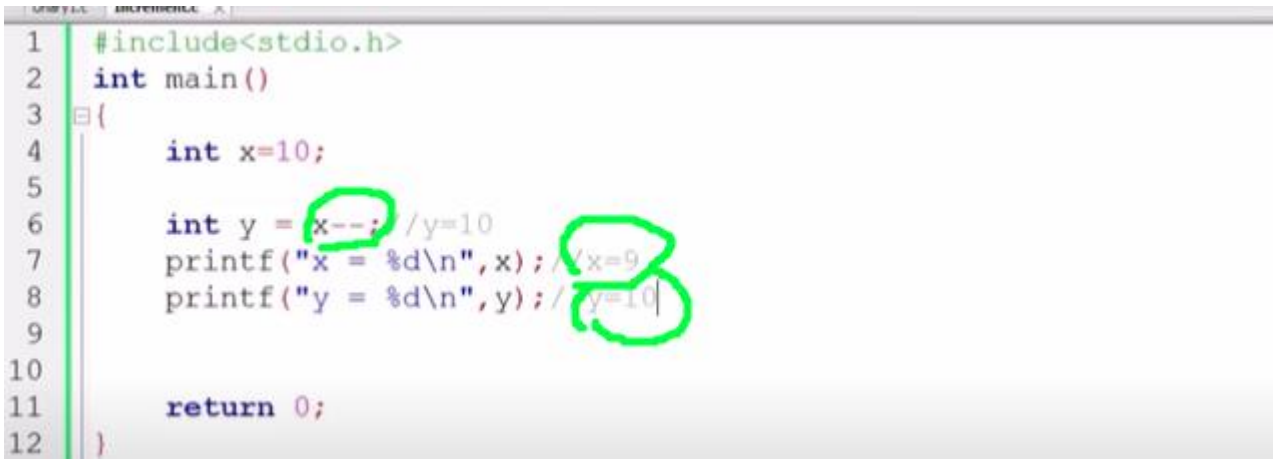
Increment Code:

```c
#include<stdio.h>

int main()

{
    int x=10;

    int y=++x; //y=11

    printf("\n\nx=%d\n",x); //x=11

    printf("y=%d\n\n",y);

    return 0;
}
```

Result:

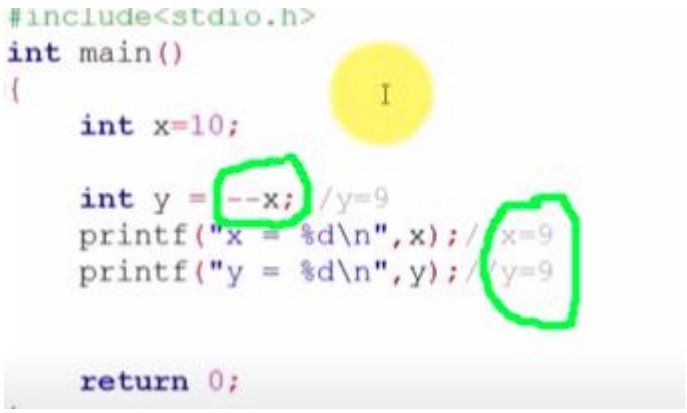
```
x=11
y=11
```

Decrement code:



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

    int y = x--; /y=10
    printf("x = %d\n",x);//x=9
    printf("y = %d\n",y);//y=10


    return 0;
}
```

Decrement code:



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

    int y = --x; /y=9
    printf("x = %d\n",x);//x=9
    printf("y = %d\n",y);//y=9


    return 0;
```

Multiple increment decrement code:

```
#include<stdio.h>

int main()

{
    int x=10;

    printf("%d\n",x++); //x=10
    printf("%d\n",x); //x=11
    printf("%d\n",++x); //x=12
    printf("%d\n",x); //x=12
```

```c
    printf("%d\n",x--); //x=12
    printf("%d\n",--x); //x=10

    return 0;
}
```

Result:

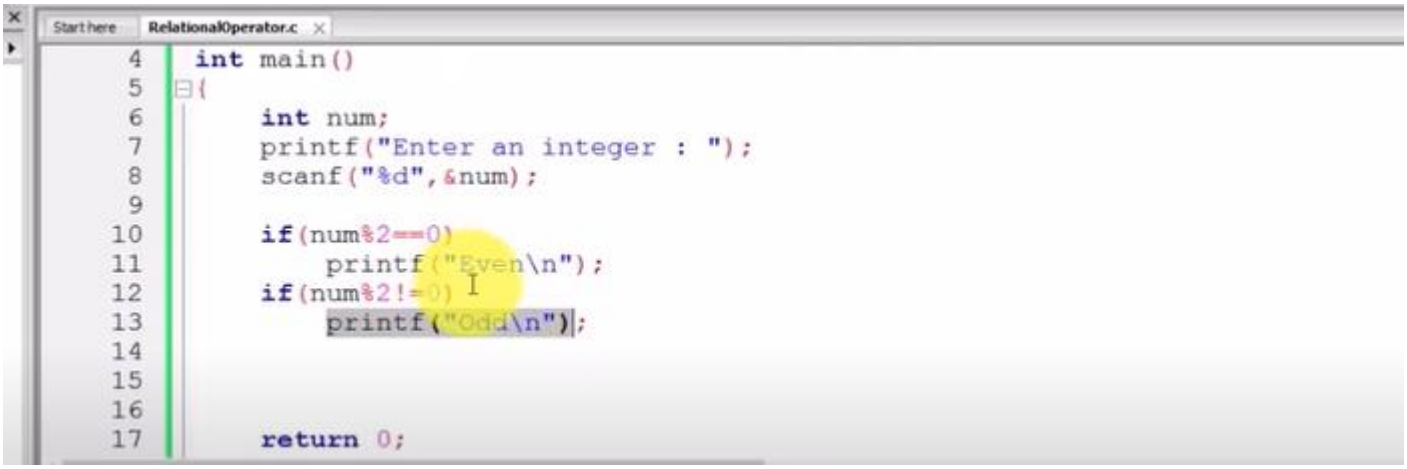
```
10
11
12
12
12
10
PS C:\Users\ranga\Documents\anisulcprog.c>
```

Relational Operators

If else statement

*** If and else if এর মধ্যে condition দেওয়া হই কিন্তু else এর মধ্যে condition দেওয়া হইনা

If statement(for single statement) code:



Result:

```
Enter an integer : 10
Even

Process returned 0 (0x0)   execution time : 3.943 s
Press any key to continue.
```

If else (for single statement) code:

```
tarthere   *RelationalOperator.c
 4  int main()
 5  {
 6      int num;
 7      printf("Enter an integer : ");
 8      scanf("%d",&num);
 9
10      if(num%2==0)
11          printf("Even\n");
12      else
13          printf("Odd\n");
14
15
16
17      return 0;
```

Result:

```
Enter an integer : 10
Even

Process returned 0 (0x0)   execution time : 3.943 s
Press any key to continue.
```

If else (for multiple statement) code:

```c
#include<stdio.h>
int main()
{
    int x=11;

   if(x==10)
   {
   printf("good morning\n");
   printf("its morning");
   }
   else
   {
   printf("not morning\n");
   printf("not 10");
   }
    return 0;
}
```

Result:

```
not morning
not 10
```

If ….. else if ……else (else if ladder) statement code:

```c
#include <stdio.h>

int main() {
    int marks;

    printf("Enter a student's mark: ");
    scanf("%d", &marks);

    if (marks >= 80) {
        printf("Your Grade: Distinction\n");
    } else if (marks >= 70) {
        printf("Your Grade: Very Good\n");
    } else if (marks >= 60) {
        printf("Your Grade: Pass\n");
    } else {
        printf("Your Grade: Fail\n");
    }

    return 0;
}
```

Result:

```
Enter a student's mark: 78
Your Grade: Very Good
```

,

```
Enter a student's mark: 86
Your Grade: Distinction
```

,

```
Enter a student's mark: 35
Your Grade: Fail
```

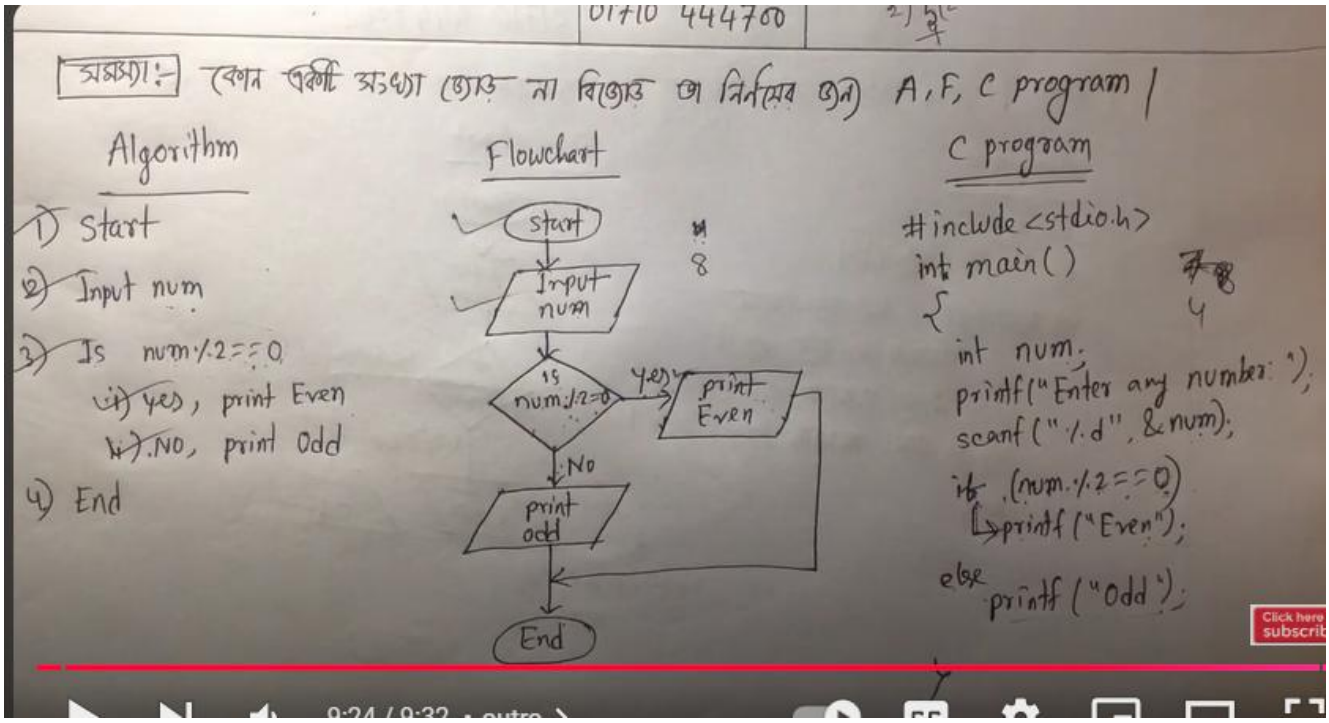
If ….else if….else code:

```c
#include<stdio.h>
int main()
{
    int num1,num2;

    printf("enter number 1 and number 2:");
    scanf("%d %d",&num1,&num2);

if (num1 > num2) {
    printf("Large = %d\n", num1);
} else if (num1 < num2) {
    printf("Large = %d\n", num2);
} else {
    printf("Numbers are equal\n");
}
return 0;
}
```

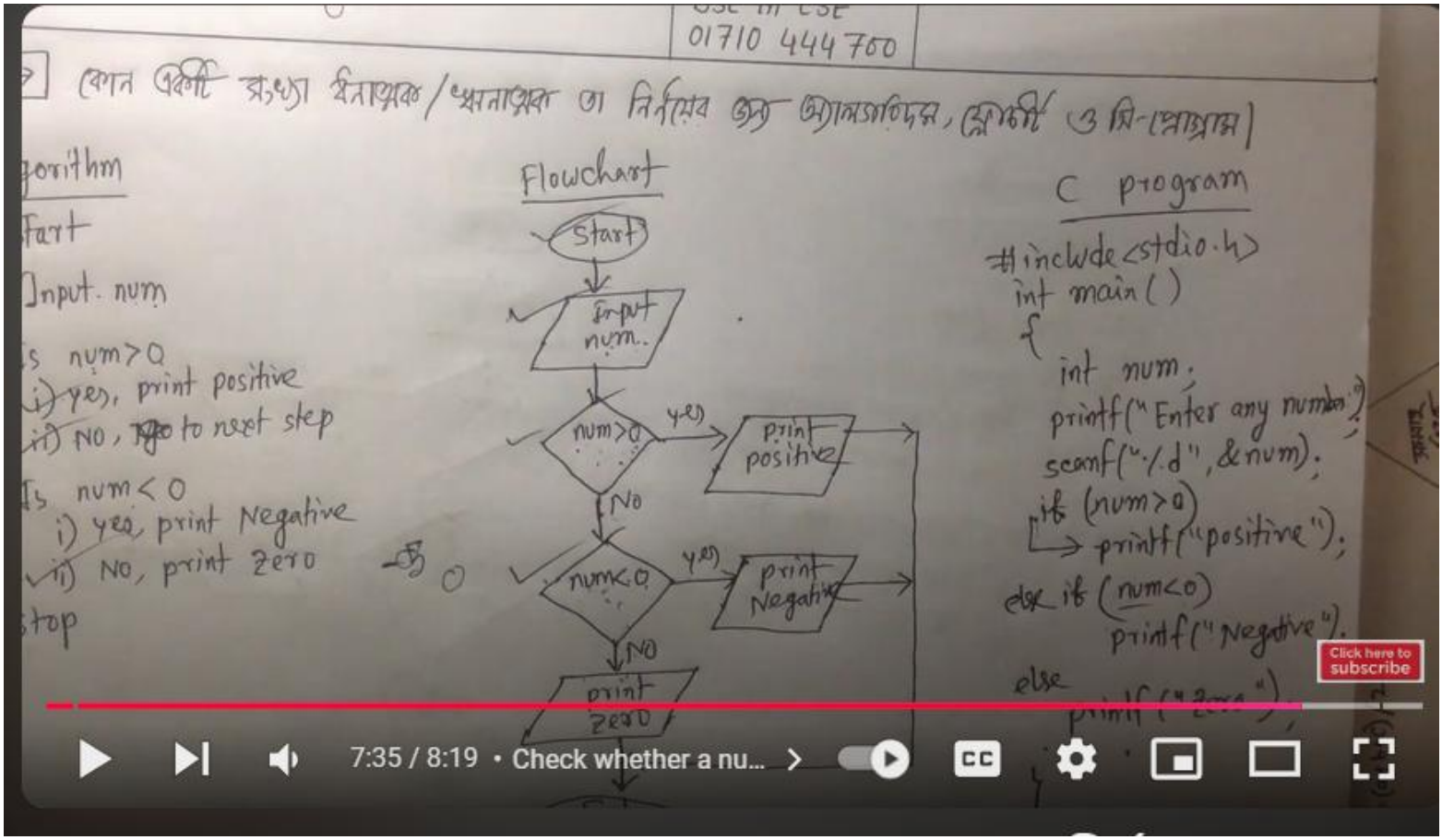
Result:

```
enter number 1 and number 2:4 9
Large = 9
```

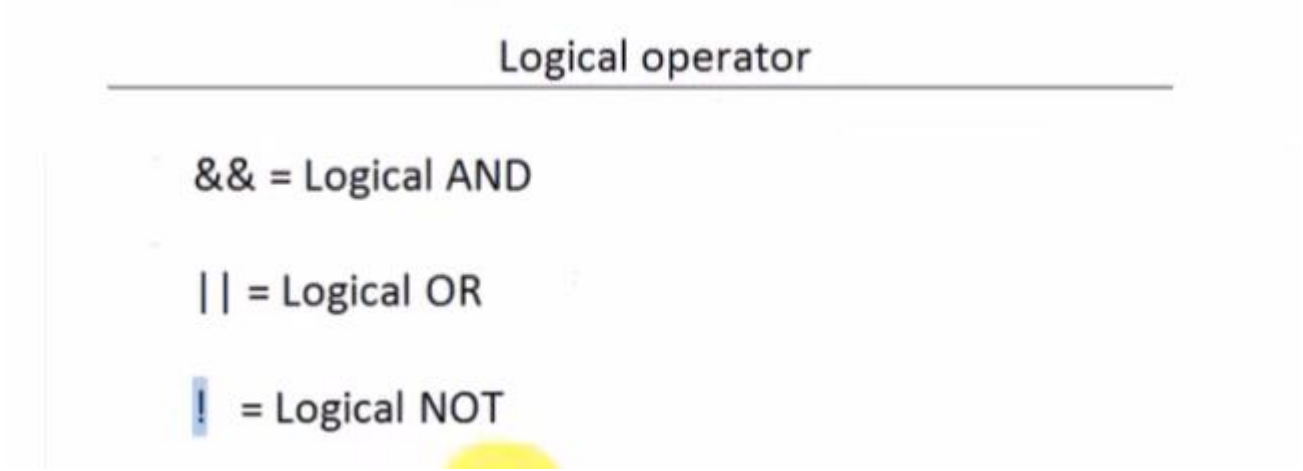
Odd even algorithm ,flowchart ,code:



Negative positive algorithm, flowchart, code:



Logical operators



Logical operators code:

```
//Enter a letter to check vowel or consonant
#include<stdio.h>
int main()
{

    char ch;
    printf("Enter a letter : ");
    scanf("%c",&ch);

    if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
        printf("Vowel");

    else
        printf("Consonant")
```

Result:

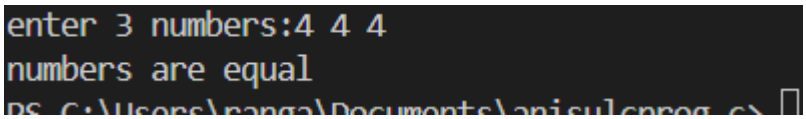
```
Enter a letter : x
Consonant
Process returned 9 (0x9)   execution time : 3.548 s
Press any key to continue.
```

Logical operators code:

```
#include<stdio.h>
int main()
{
    int num1,num2,num3;

    printf("enter 3 numbers:");
    scanf("%d %d %d",&num1,&num2,&num3);

    if (num1>num2 && num1>num3)
    printf("large number is=%d\n",num1);

    else if (num2>num1 && num2>num3)
    printf("large number is=%d\n",num2);

    else if (num3>num1 && num3>num2)
    printf("large number is=%d\n",num3);

    else
    printf("numbers are equal\n");

    return 0;

}
```

Result:

,

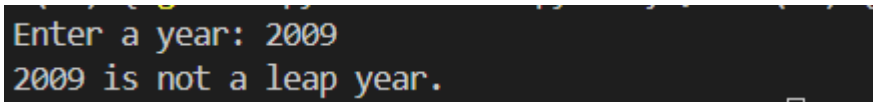


Video 77 to 79 due

Leap year code:

```c
#include <stdio.h>

int main()
{
    int year;

    printf("Enter a year: ");
    scanf("%d", &year);

    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
    {
        printf("%d is a leap year.\n", year);
    } else {
        printf("%d is not a leap year.\n", year);
    }

    return 0;

}
```

Result:



Switch statement

```c
//switch keyword : switch, case, break, default
```

Switch code:

```c
#include <stdio.h>

int main()
{
    int digit;

    printf("Enter a digit: ");
    scanf("%d", &digit);

    switch (digit)
    {
```

```c
    case 0:
    printf("zero\n");
    break;
    case 1:
    printf("one\n");
    break;
    case 2:
    printf("two\n");
    break;
    case 3:
    printf("three\n");
    break;
    case 4:
    printf("four\n");
    break;
    case 5:
    printf("five\n");
    break;
    case 6:
    printf("six\n");
    break;
    case 7:
    printf("seven\n");
    break;
    case 8:
    printf("eight\n");
    break;

    default:
    printf("not a valid digit");


    return 0;
    }
}
```

Result:

```
Enter a digit: 5
five
```

,

```
Enter a digit: 67
not a valid digit
```

Interesting facts about switch:

1.

Following are some interesting facts about switch statement.

1) The expression used in switch must be integral type (int, char and enum). Any other type of expression is not allowed.

```
// float is not allowed in switch
#include <stdio.h>
int main()
{
    float x = 1.1;
    switch (x)
    {
        case 1.1: printf("Choice is 1");
                  break;
        default: printf("Choice other than 1, 2 and 3");
                 break;
    }
    return 0;
}
```

Run on IDE

Output:

```
Compiler Error: switch quantity not an integer
```

2.

2) All the statements following a matching case execute until a break statement is reached.

```
// There is no break in all cases
#include <stdio.h>
int main()
{
   int x = 2;
   switch (x)
   {
       case 1: printf("Choice is 1\n");
       case 2: printf("Choice is 2\n");
       case 3: printf("Choice is 3\n");
       default: printf("Choice other than 1, 2 and 3\n");
   }
   return 0;
}
```

Run on IDE

Output:

```
Choice is 2
Choice is 3
Choice other than 1, 2 and 3
```

3.

3) The default block can be placed anywhere. The position of default doesn't matter, it is still executed if no match found.

```
// The default block is placed above other cases.
#include <stdio.h>
int main()
{
   int x = 4;
   switch (x)
   {
       default: printf("Choice other than 1 and 2");
                break;
       case 1: printf("Choice is 1");
               break;
       case 2: printf("Choice is 2");
                break;
   }
   return 0;
}
```

Output:

```
Choice other than 1 and 2
```

4.

5) The statements written above cases are never executed After the switch statement, the control transfers to the matching case, the statements written before case are not executed.

```
// Statements before all cases are never executed
#include <stdio.h>
int main()
{
   int x = 1;
   switch (x)
   {
      x = x + 1; // This statement is not executed
      case 1: printf("Choice is 1");
              break;
      case 2: printf("Choice is 2");
              break;
      default: printf("Choice other than 1 and 2");
               break;
   }
   return 0;
}
```

Output:

```
Choice is 1
```

5.

6) Two case labels cannot have same value

```
// Program where two case labels have same value
#include <stdio.h>
int main()
{
   int x = 1;
   switch (x)
   {
      case 2: printf("Choice is 1");
              break;
      case 1+1: printf("Choice is 2");
               break;
   }
   return 0;
}
```

Run on IDE

Output:

```
Compiler Error: duplicate case value
```

Switch (vowel consonant) code:

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

    printf("enter a letter:");
    scanf("%c",&ch);
```

```
    switch (ch)
    {
    case 'a':
    case 'e':
    case 'i':
    case 'o':
    case 'u':
    case 'A':
    case 'E':
    case 'I':
    case 'O':
    case 'U':

    printf("Vowel\n");
        break;

    default:
    printf("Consonant\n");
        break; //no need to write break after default

}
return 0;
}
```

Result:

```
enter a letter:a
Vowel
```

,

```
enter a letter:g
Consonant
```

Switch function menu-based temp conversion code:

```
#include<stdio.h>
int main()
{
    int choice;
    float temp,convertedtemp;
```

```c
    printf("\nTemperature convertion menu\n");
    printf("1. fahrenheit to celsius\n");
    printf("2. celsius to fahrenheit\n");
    printf("enter your choice:");
    scanf("%d",&choice);

    switch (choice)
    {
    case 1:
    {
        printf("enter the fahrenheit temperature:");
        scanf("%f",&temp);
        convertedtemp=(temp-32)/1.8;
        printf("the temperature in celcius is:%f\n",convertedtemp);
        break;
    }
    case 2:
    {
        printf("enter the celcius temperature:");
        scanf("%f",&temp);
        convertedtemp=(temp*1.8)+32;
        printf("the temperature in fahrenheir is:%f\n",convertedtemp);
        break;
    }
    default:
    printf("not a correct option");

    }
    return 0;
}
```

Result:

```
Temperature convertion menu
1. fahrenheit to celsius
2. celsius to fahrenheit
enter your choice:1
enter the fahrenheit temperature:35.58
the temperature in celcius is:1.988890
```

,

```
Temperature convertion menu
1. fahrenheit to celsius
2. celsius to fahrenheit
enter your choice:3
not a correct option
```

Switch calculator code:

```c
#include<stdio.h>
int main()
{
    double num1,num2;
    char op;

    printf("enter an operator(+,-,*,/) :");
    scanf("%c",&op); //we have to take operator before number

    printf("enter two numbers:");
    scanf("%lf %lf",&num1,&num2);

    switch (op)
    {
        case '+':
    {
        printf("%lf+%lf=%lf\n",num1,num2,num1+num2);
        break;
    }
    case '-':
    {
        printf("%lf-%lf=%lf\n",num1,num2,num1-num2);
        break;
    }
    case '*':
    {
        printf("%lf*%lf=%lf\n",num1,num2,num1*num2);
        break;
    }
    case '/':
    {
        printf("%lf/%lf=%lf\n",num1,num2,num1/num2);
        break;
    }
    default:
    printf("not a valid operator\n");
        break;
    }
    return 0;
}
```

Result:

```
enter an operator(+,-,*,/) :*
enter two numbers:3 9
3.000000*9.000000=27.000000
```

Conditional operator

Code:

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

    int num1,num2,large;

    printf("Enter two numbers : ");
    scanf("%d %d",&num1,&num2);


    large =   (num1>num2) ? num1 : num2;
    printf("Large number = %d\n",large);


    return 0;
}
```

Result:

```
Enter two numbers : 5 3
Large number = 5
```

Code:

```c
#include<stdio.h>
int main()
{
    int marks;
```

```
    char result;

    printf("enter marks :");
    scanf("%d",&marks);

    result=(marks>75)? printf("result:First class"):
    ((marks>65)?printf("result:second class"):
    (marks>55)?printf("result:Third class"):
    printf("result:Forth class"));

    return 0;
}
```

Result:

```
enter marks :64
result:Third class
PS C:\Users\ranga\Documents\ancslides\anc5thslide>
```

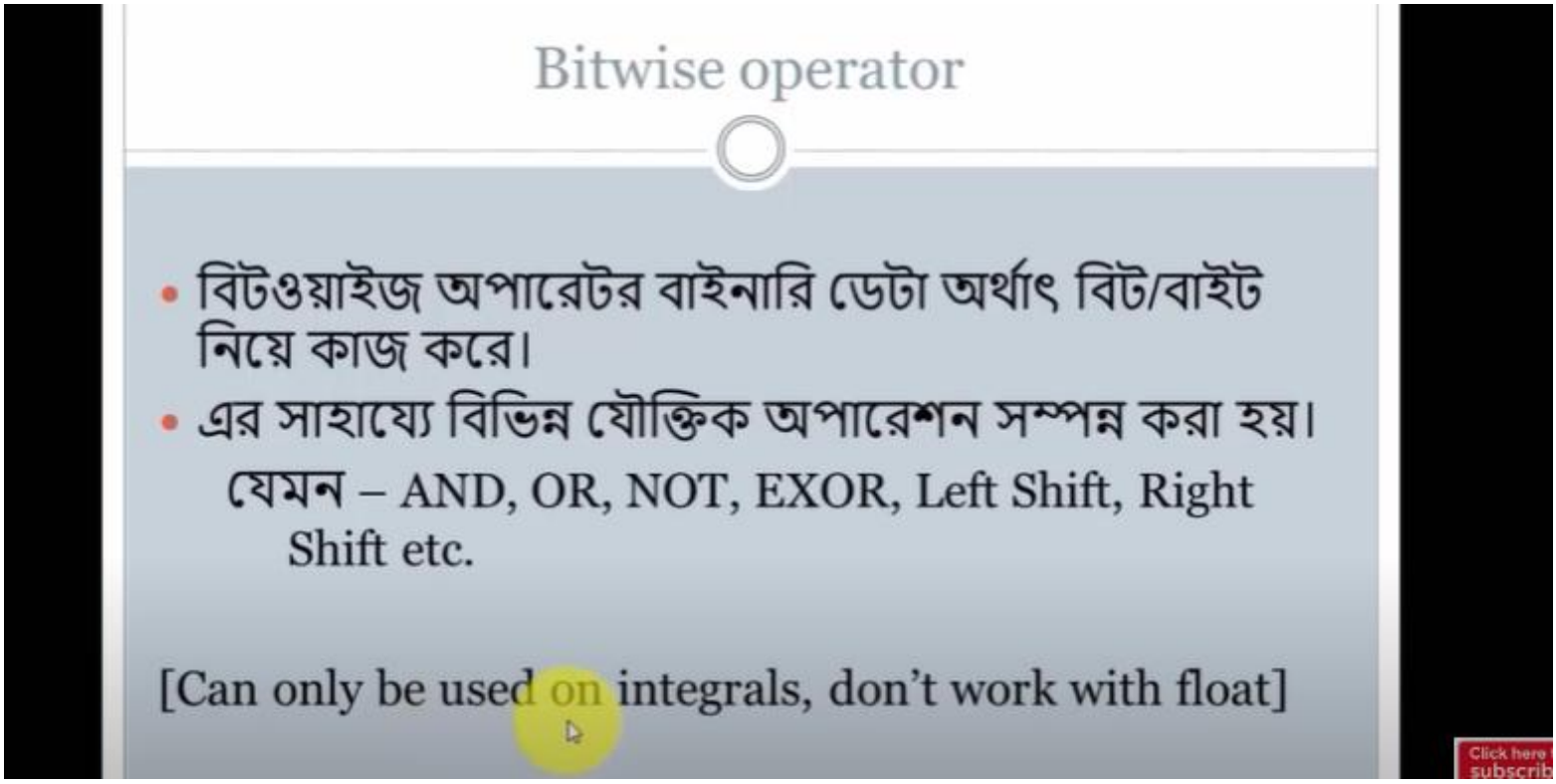
Code:

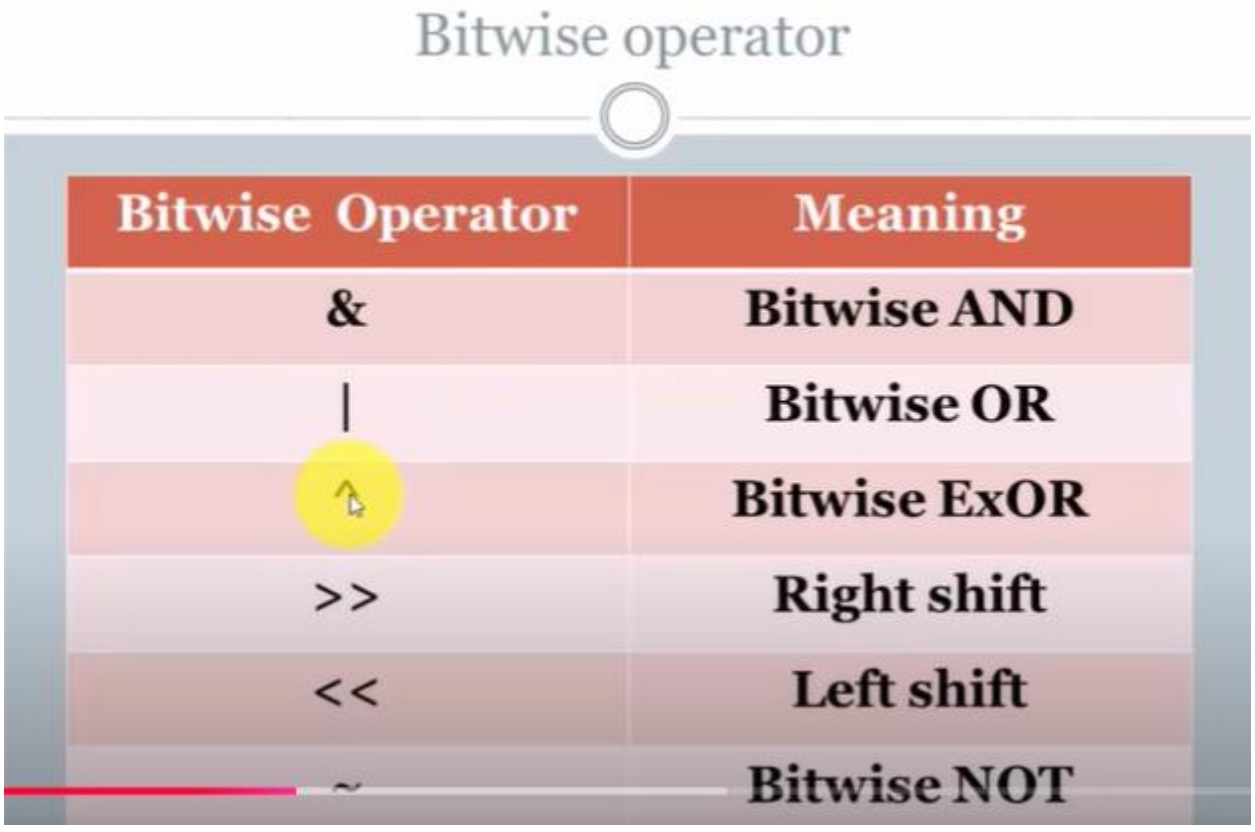
```
#include<stdio.h>
int main()
{
    int n1,n2,n3;
    char smallest_number;

    printf("enter three numbers:");
    scanf("%d%d%d",&n1,&n2,&n3);

    smallest_number=(n1<n2 && n1<n3)?printf("smallest number is :%d\n",n1):
    ((n2<n1 && n2<n3)?printf("smallest number is :%d\n",n2):
    printf("smallest number is :%d\n",n3));

    return 0;
}
```

Result:

```
enter three numbers:4 6 9
smallest number is :4
```

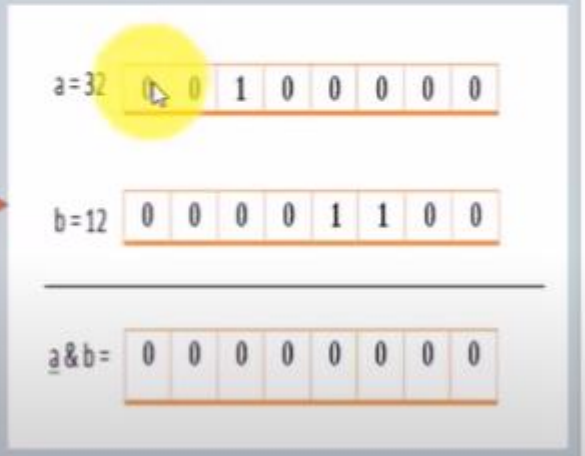
Bitwise operator





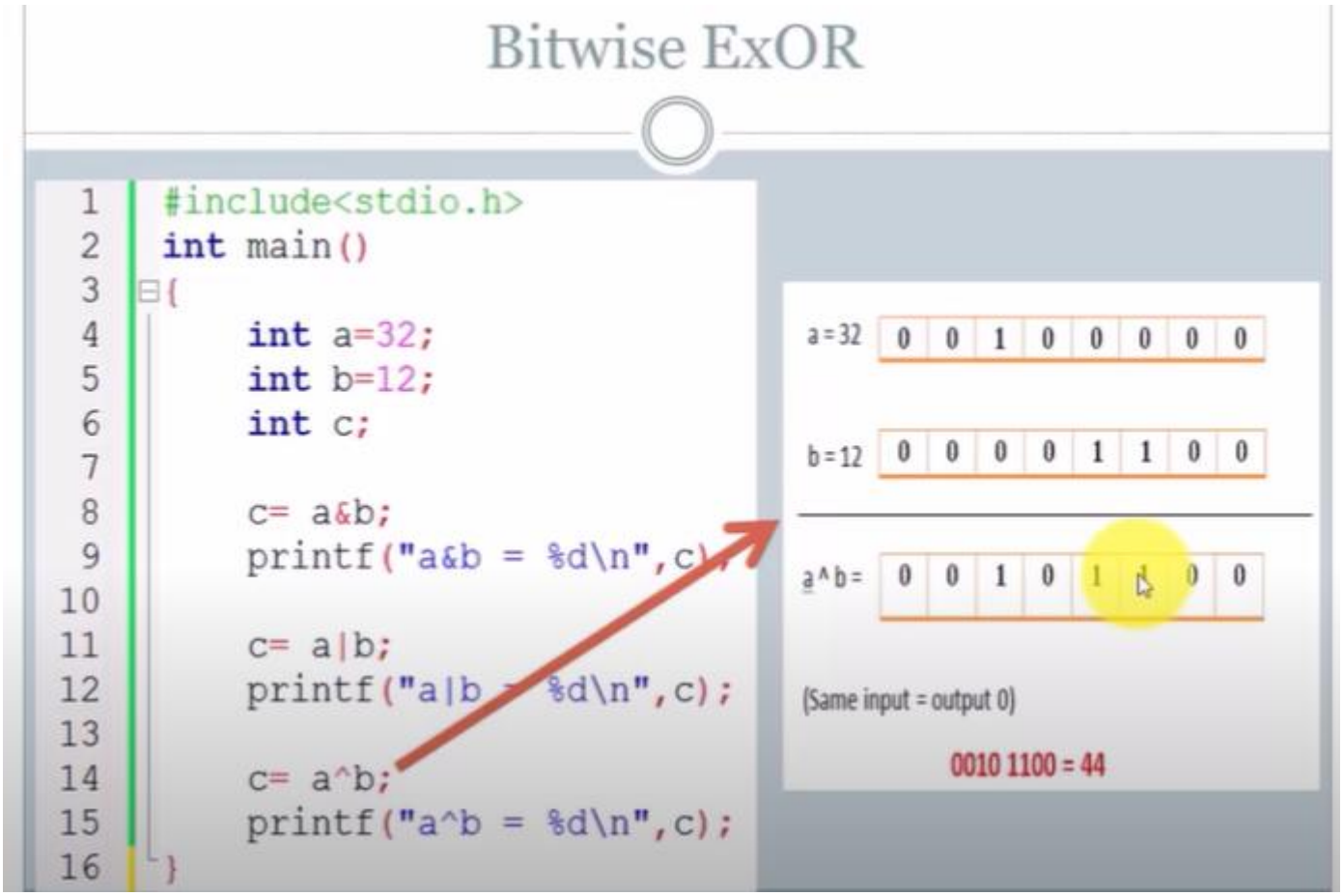
Bitwise AND

```c
#include<stdio.h>
int main()
{
    int a=32;
    int b=12;
    int c;

    c= a&b;
    printf("a&b = %d\n",c);

    c= a|b;
    printf("a|b = %d\n",c);

    c= a^b;
    printf("a^b = %d\n",c);
}
```

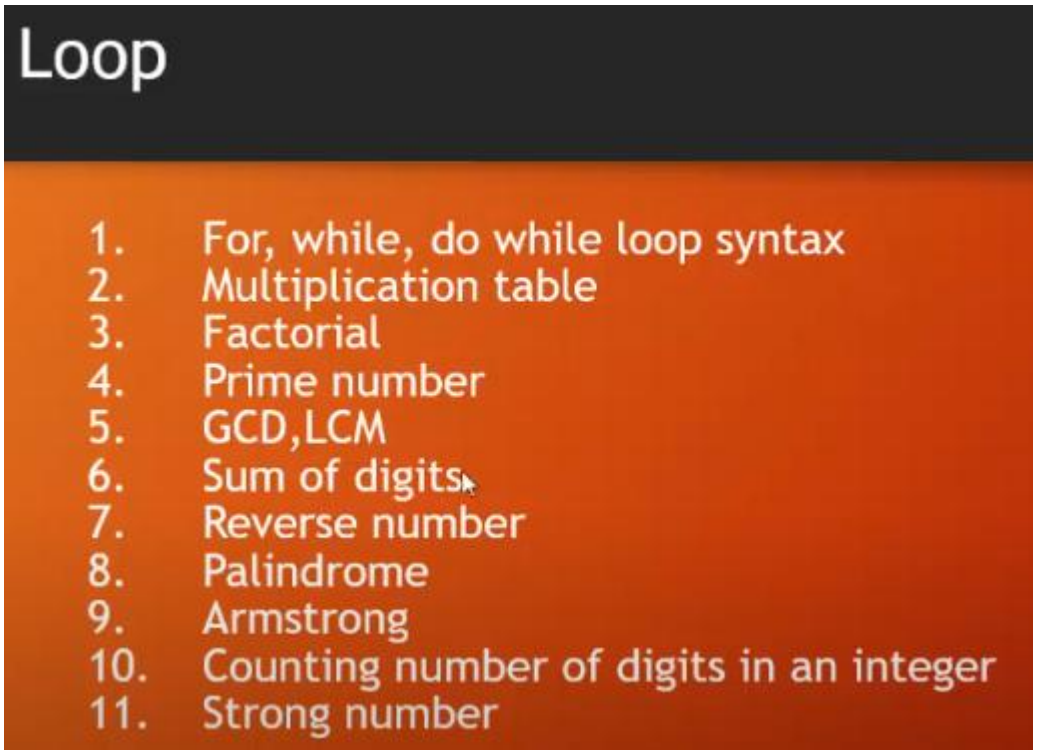


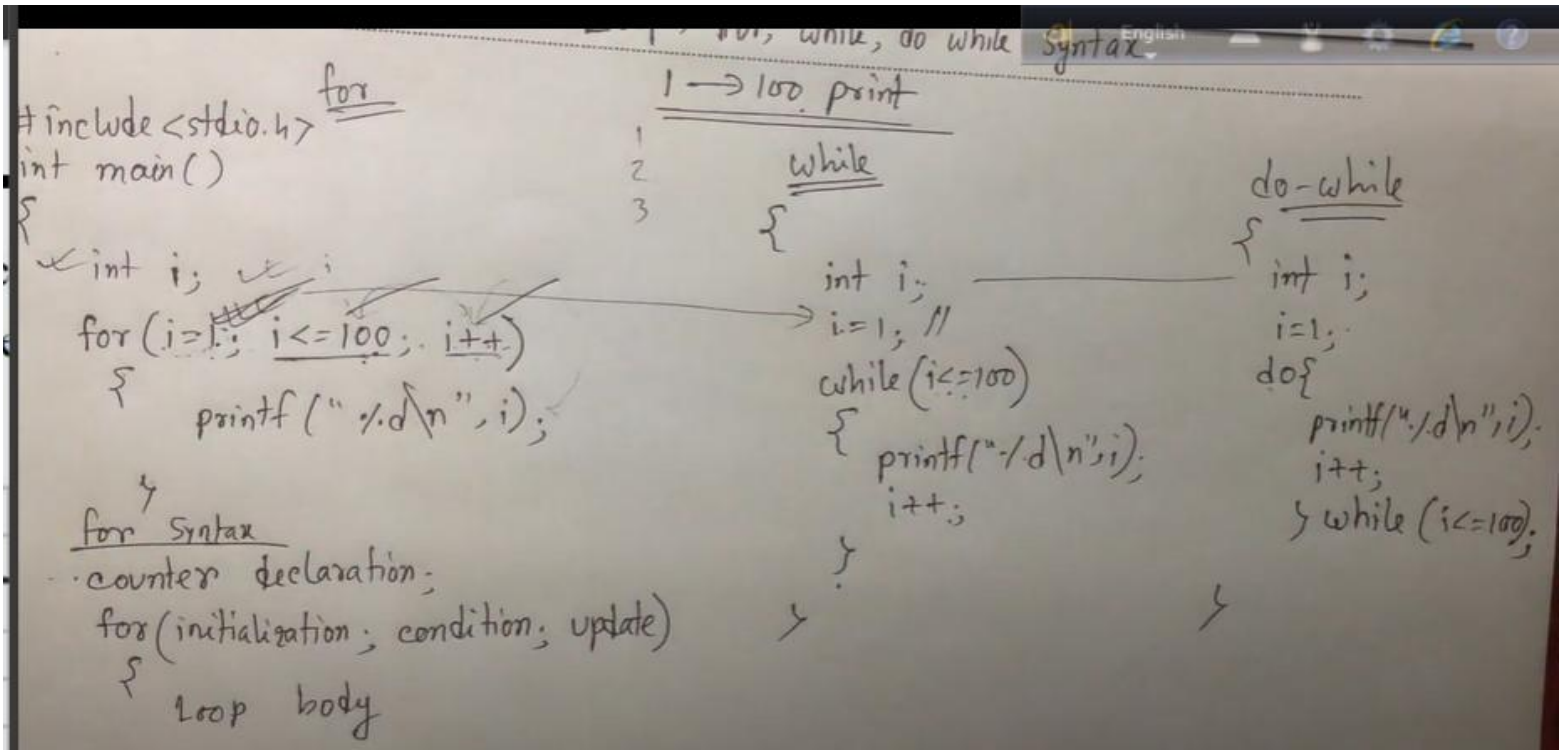
Bitwise OR

```c
#include<stdio.h>
int main()
{
    int a=32;
    int b=12;
    int c;

    c= a&b;
    printf("a&b = %d\n",c);

    c= a|b;
    printf("a|b = %d\n",c);

    c= a^b;
    printf("a^b = %d\n",c);
}
```


Bitwise ExOR
#include<stdio.h>
int main()
{
int a=32;
int b=12;
int c;
c= a&b;
printf("a&b = %d\n",c);
c= a|b;
printf("a|b = %d\n",c);
c= a^b;
printf("a^b = %d\n",c);
}
a=32 0 0 1 0 0 0 0 0
b=12 0 0 0 0 1 1 0 0
a^b= 0 0 1 0 1 1 0 0
(Same input = output 0)
0010 1100 = 44

Loop


Loop
1. For, while, do while loop syntax
2. Multiplication table
3. Factorial
4. Prime number
5. GCD,LCM
6. Sum of digits
7. Reverse number
8. Palindrome
9. Armstrong
10. Counting number of digits in an integer
11. Strong number



For loop

Code:

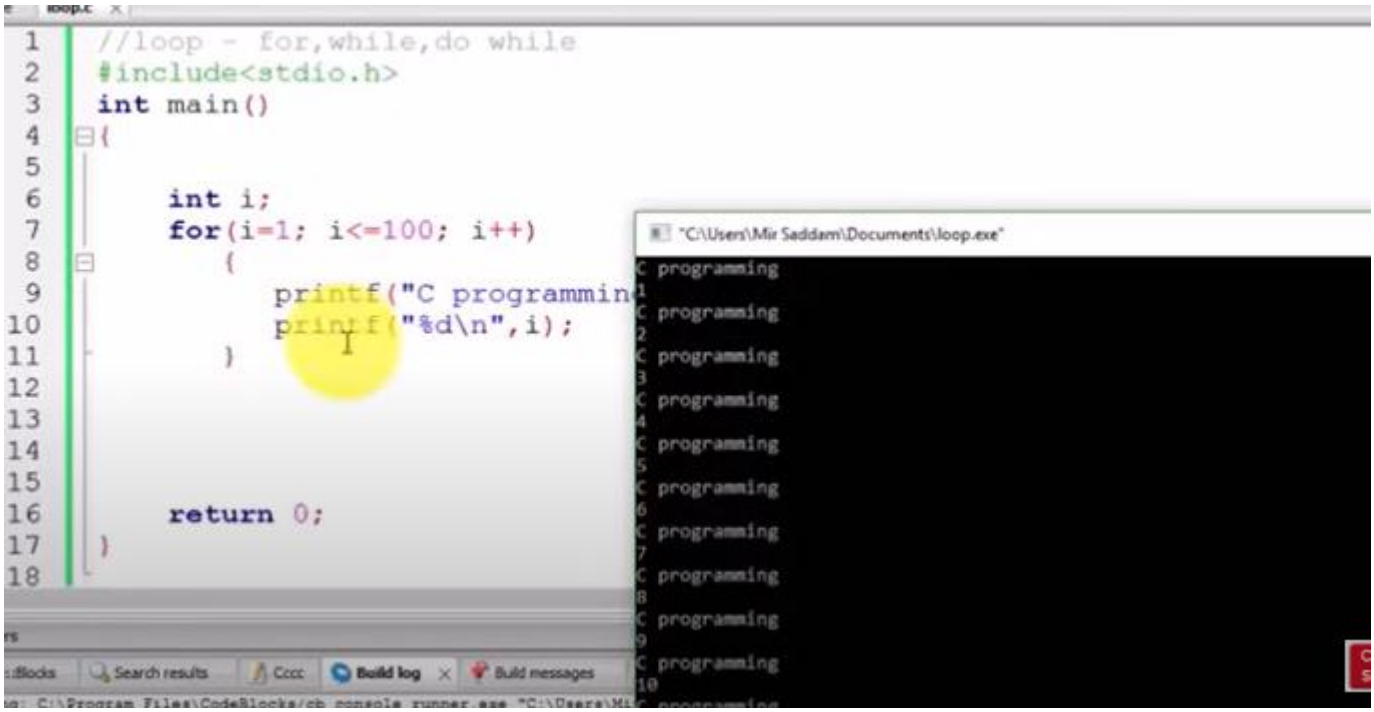
```c
//loop - for,while,do while
#include<stdio.h>
int main()
{

    int i;
    for(i=1; i<=5; i++) // for (inintialization; condition; counterUpdate)
                        // {

                        // }
        printf("C programming\n");



    return 0;
}
```

Result:

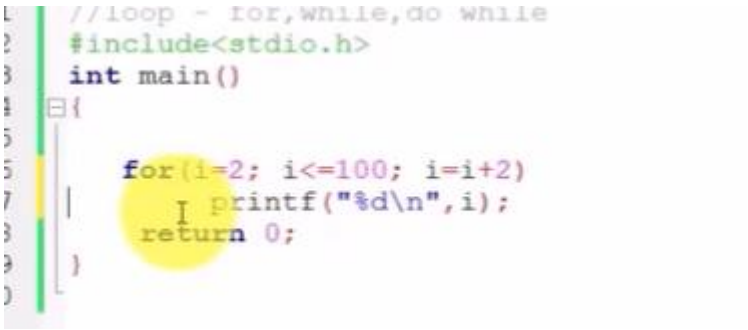
```
C programming
C programming
C programming
C programming
C programming
```

***For loop er moddhe more than one statement thakle 2nd bracket {} use korte hobe.

Code and result:



Print even numbers till 100 with for loop,code:



Result:

while loop

****there will be only condition in in while loop

Code:

```c
#include<stdio.h>
int main()
{
    int i=1;
    while(i<=10)
    {
        printf("%d\n",i);
        i++;
    }

    return 0;
```

Result:

Do while loop

Initialization will be in the int part;

Increment will be done in the do part;

And condition in the while part

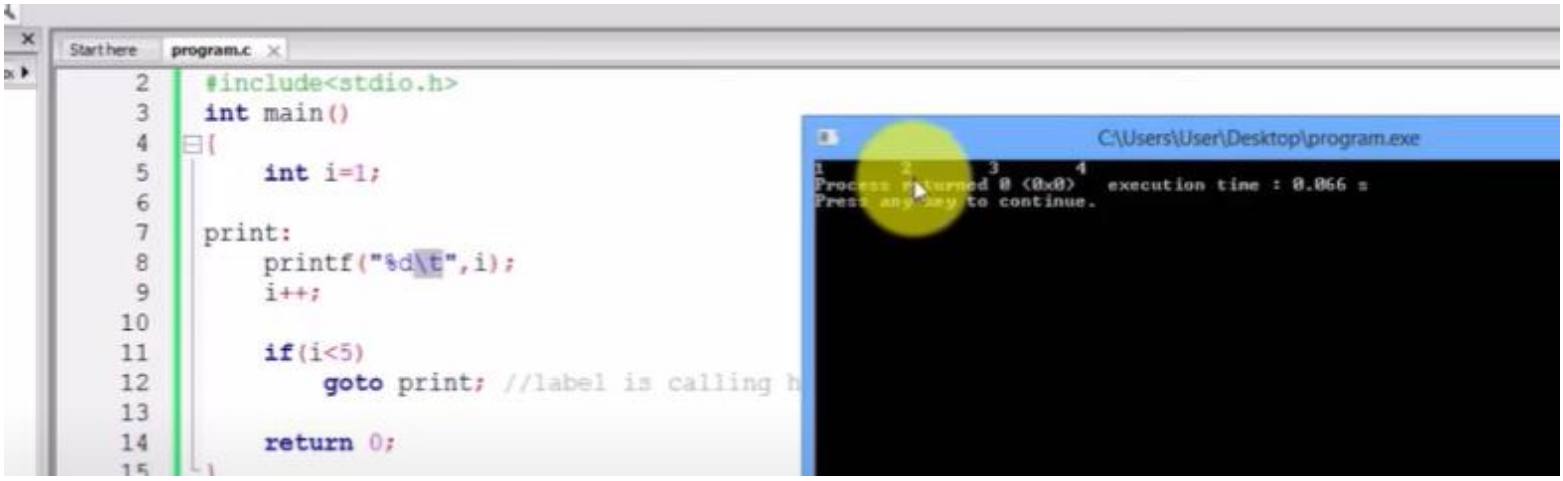
**do part er kaaj hobe at least 1 bar

Code:

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

    int i=1; //initialization
    do
       {

          printf("%d\n",i);
          i++; // increment
       }while(i<=10);//condition



    return 0;
```

Result:

Goto statement



Multiplication table code:

```
time table1.c
        ........
        3 x 10 = 30

*/
#include<stdio.h>
int main()
{
    int num,i;
    printf("Enter any number : ");
    scanf("%d",&num);
    for(i=1; i<=10;i++)
    {
        printf("%d X %d = %d\n",num,i,num*i);
    }
}
```

Result:

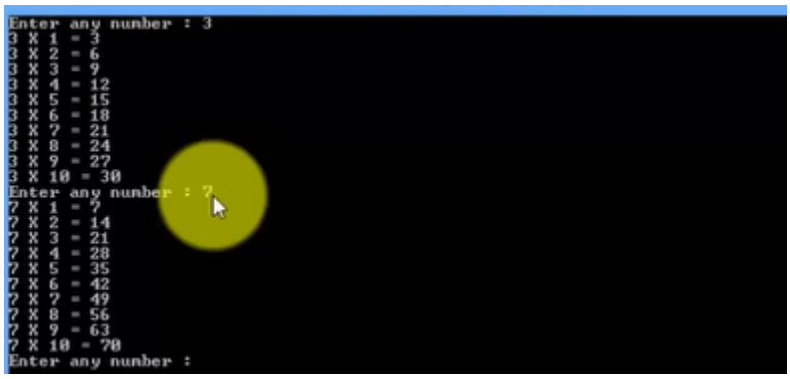
```
Enter any number : 3
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 12
3 X 5 = 15
3 X 6 = 18
3 X 7 = 21
3 X 8 = 24
3 X 9 = 27
3 X 10 = 30
```

***Jodi code bar bar terminal e run korte chai bar bar input diye, taile puro code body while loop er moddhe dite hobe (1) condition diye

Running the code multiple time and giving different input each time Code:

```
Start here   time table1.c
 9
10  */
11  #include<stdio.h>
12  int main()
13  {
14      while(1) {
15
16      int num,i;
17      printf("Enter any number : ");
18      scanf("%d",&num);
19      for(i=1; i<=10;i++)
20      {
21          printf("%d X %d = %d\n",num,i,num*i);
22      }
23
24      }
```

Result:

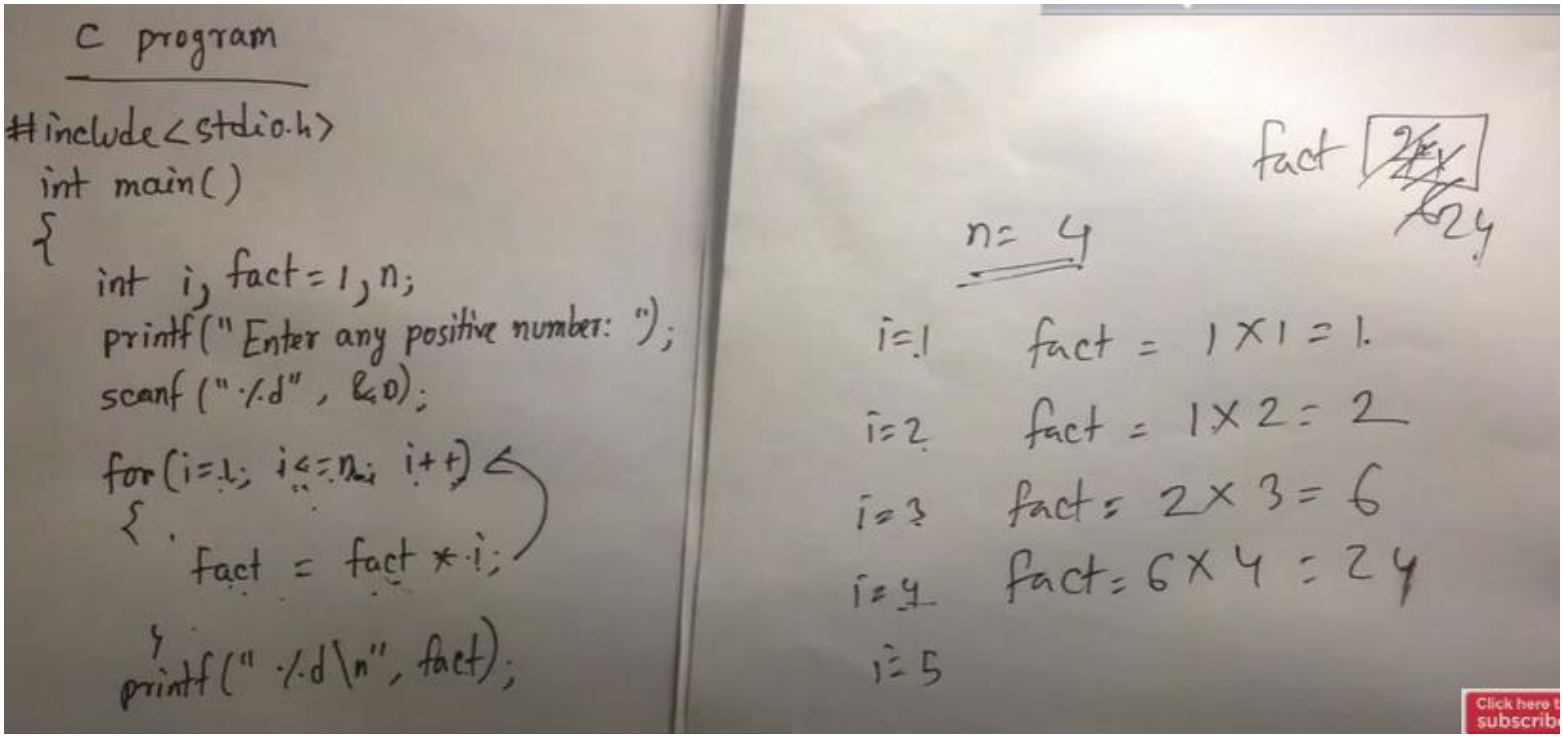


Factorial:

5!=1×2×3×4×5=120

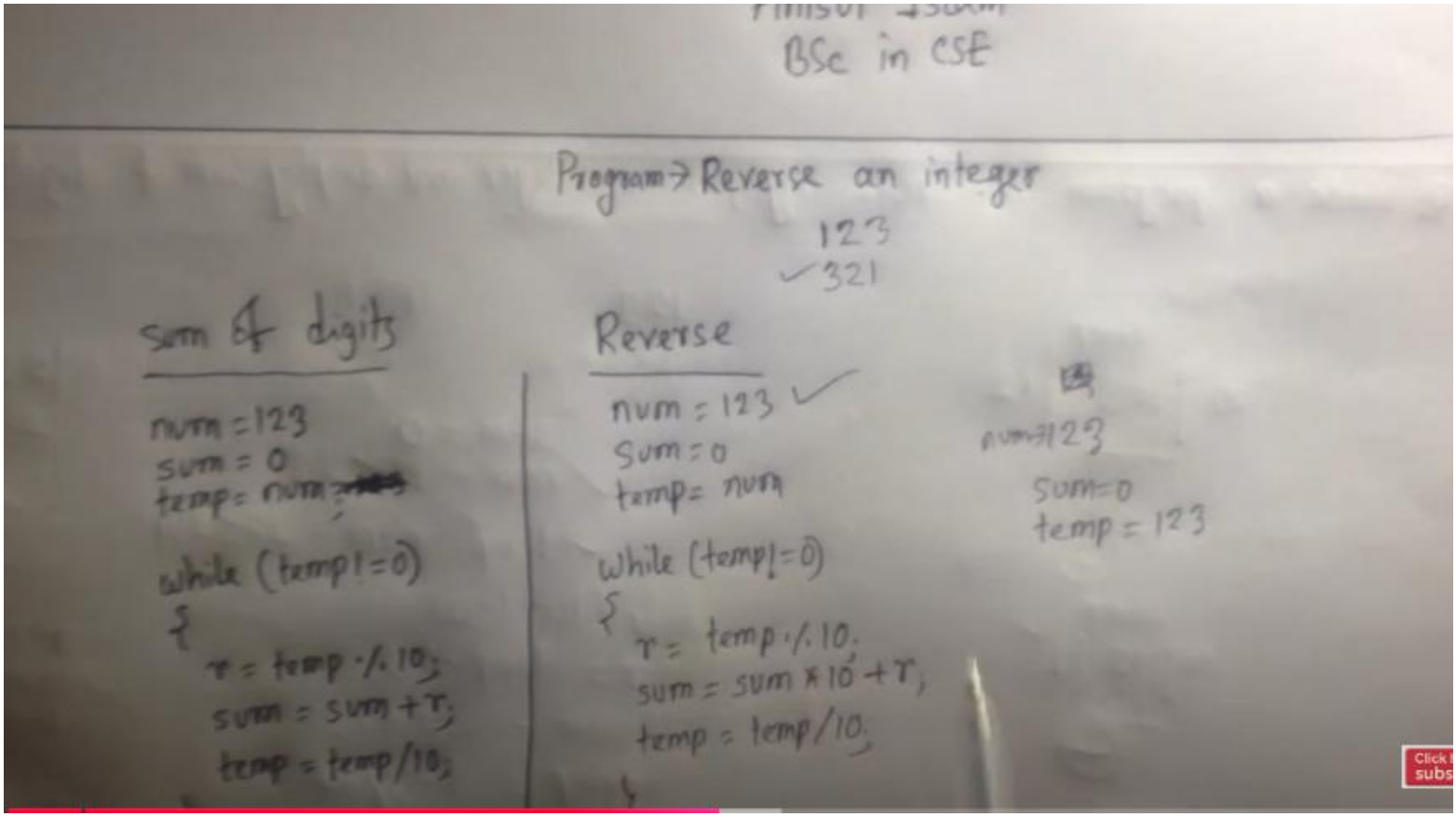
4!=1× 2×3×4 =24

0!=1



Lab e boshe korchi

Sum&reverse



Sum:

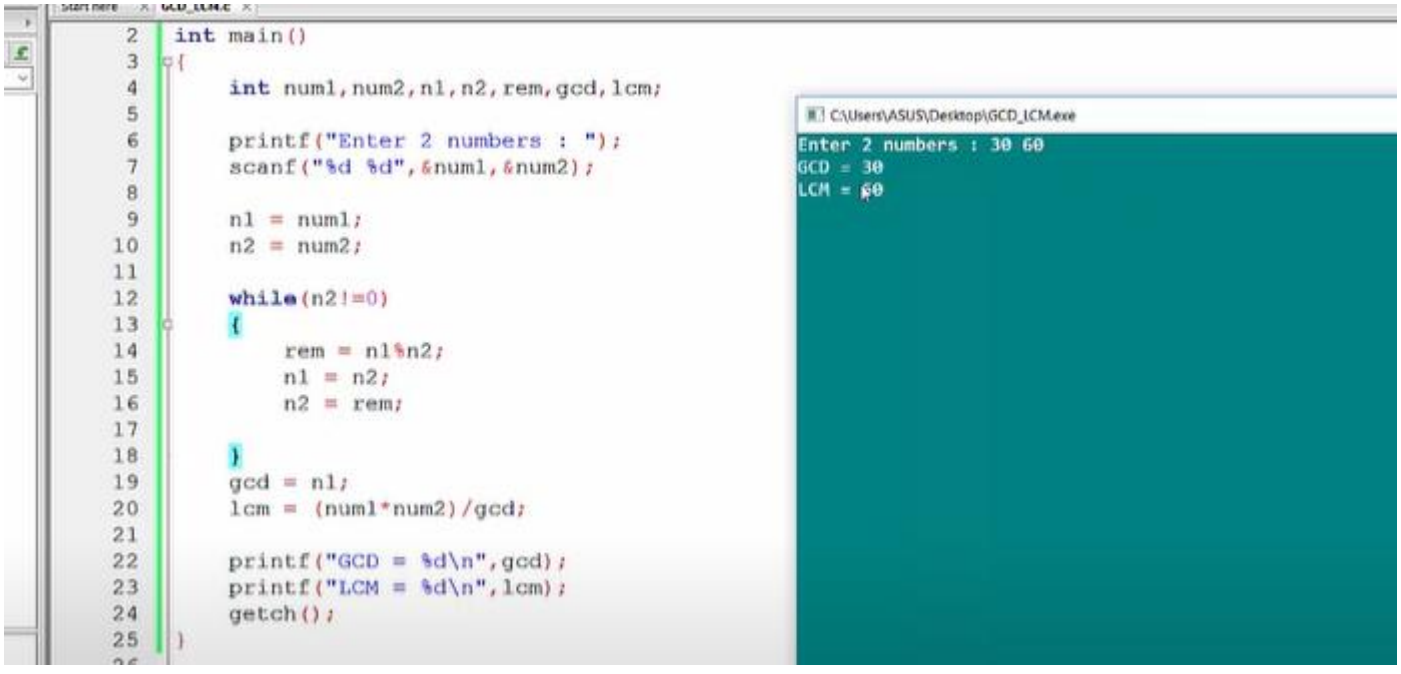
```c
#include<stdio.h>
int main()
{
    int num,temp,r,sum=0;

printf("enter number:");
scanf("%d",&num);

temp=num;

while(temp!=0)
{
    r=temp%10;
    sum=sum+r;
    temp=temp/10;
}
printf("sum of digits :%d",sum);
```

```
return 0;
}
```

Reverse:

```
#include<stdio.h>
int main()
{
    int num,temp,r,sum=0;

printf("enter number:");
scanf("%d",&num);

temp=num;

while(temp!=0)
{
    r=temp%10;
    sum=sum*10+r;
    temp=temp/10;
}
printf("sum of digits :%d",sum);

return 0;
}
```

Gcd,lcm



Moulik shonkha/prime number

```c
#include<stdio.h>
#include<math.h>
#include<ctype.h>
int main()
{
    int number,count=0;

    printf("enter any number: ");
    scanf("%d",&number);

    if(number<=1)
    {
        count=1;
    }
    else
    {
    for(int i=2; i<=sqrt(number); i++)   //i<number and i<=number is also
right but less efficient
    {
        if(number%i==0)
        {
            count=1;
        }
    }
    }
    if(count==0)
    printf("%d is a prime number",number);
    else
    printf("%d is not a prime number",number);

    return 0;
}
```

Result:

```
enter any number: 67
67 is a prime number
```

Print,count and sum of prime numbers from 1 to 100 code:

```c
//print prime numbers from 1 to 100
//print, count and sum of prime nummbers from 1 to 100

#include<stdio.h>
#include<math.h>
#include<ctype.h>
int main()
{
    int number, count=0, totalprimenummbers=0,sumofprimenummbers=0;

    for(number=1; number<=100; number++)
    {
    count=0;
    if(number<=1)
    {
        count=1;
    }
    else
    {
    for(int i=2; i<=sqrt(number); i++)   //i<number and i<=number is also
right but less efficient
    {
        if(number%i==0)
        {
            count=1;
        }
    }
    }

    if(count==0)
    {
    printf("%d ",number);
    totalprimenummbers++;
    sumofprimenummbers=sumofprimenummbers+number;
    }
    }
    printf("\n\ntotal prime numbers: %d\n",totalprimenummbers);
    printf("\nsum of prime numbers: %d\n",sumofprimenummbers);
    return 0;
```

```
}
```

Result:

```
 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

 total prime numbers: 25

 sum of prime numbers: 1060
```

Print, count and sum of prime numbers from m to n code:

```c
//print prime numbers from 1 to 100
//print, count and sum of prime nummbers from m to n

#include<stdio.h>
#include<math.h>
#include<ctype.h>
int main()
{
    int number, count=0, totalprimenummbers=0,sumofprimenummbers=0,m,n;

    printf("enter starting number and ending number:");
    scanf("%d%d",&m,&n);

    for(number=m; number<=n; number++)
    {
    count=0;
    if(number<=1)
    {
        count=1;
    }
    else
    {
    for(int i=2; i<=sqrt(number); i++)   //i<number and i<=number is also
right but less efficient
    {
        if(number%i==0)
        {
            count=1;
        }
    }
    }
```

```
    if(count==0)
    {
        printf("%d ",number);
        totalprimenummbers++;
        sumofprimenummbers=sumofprimenummbers+number;
    }
    }
        printf("\n\ntotal prime numbers: %d\n",totalprimenummbers);
        printf("\nsum of prime numbers: %d\n",sumofprimenummbers);
    return 0;

}
```

Result:

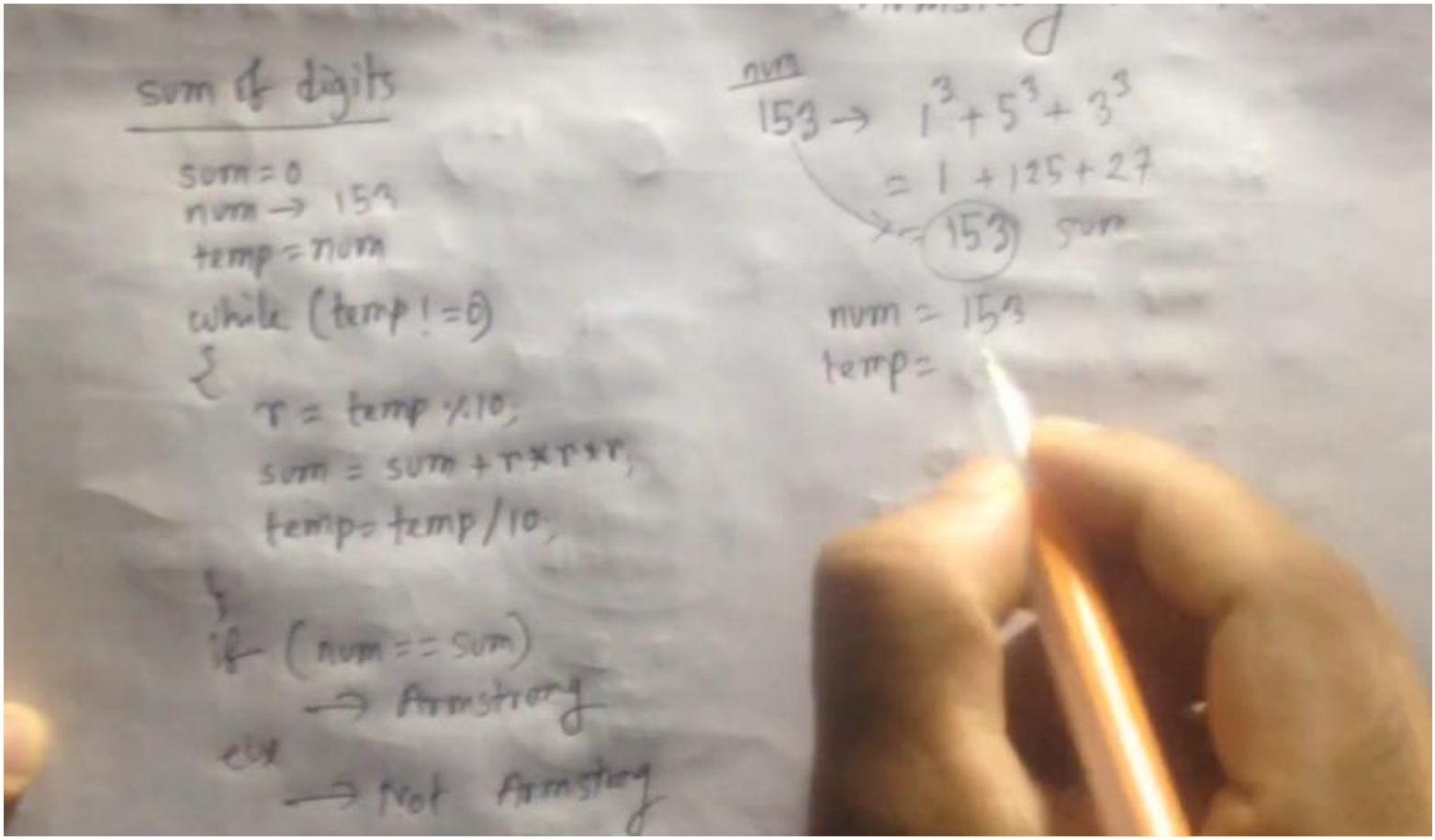
```
enter starting number and ending number:34 98
37 41 43 47 53 59 61 67 71 73 79 83 89 97

total prime numbers: 14

sum of prime numbers: 900
```

Palindrome number:

```
#include<stdio.h>
int main()
{
    int num,temp,r,sum=0;

printf("enter number:");
scanf("%d",&num);

temp=num;

while(temp!=0)
{
    r=temp%10;
    sum=sum*10+r;
    temp=temp/10;
```

```
}
if(sum==num)
{
    printf("palindrome");
}
else
printf("not palindrome");


return 0;
}
```

Armstrong number:



****An **Armstrong number** is a number that is equal to the sum of its own digits raised to the power of the number of digits.

For an n-digit number, N, it is an Armstrong number if:

$$N = d_1^n + d_2^n + \cdots + d_n^n$$

- $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$
- Hence, 153 is an Armstrong number.

Armstrong number code:

```c
#include<stdio.h>
int main()
{
    int num,temp,r,sum=0;

printf("enter number:");
scanf("%d",&num);

temp=num;

while(temp!=0)
{
    r=temp%10;
    sum=sum+r*r*r;
    temp=temp/10;
}
if(sum==num)
    printf("%d is an armstrong number",num);
else
    printf("%d is not an armstrong number",num);

return 0;
}
```

Result:

```
enter number:153
153 is armstrong number
```

Armstrong numbers in a range:

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

```
    int initialnum,finalnum,temp,r,sum=0,i;

printf("enter initial number:");
scanf("%d",&initialnum);
printf("enter final number:");
scanf("%d",&finalnum);

for(i=initialnum;i<=finalnum;i++)
{
    temp=i;

    while(temp!=0)
    {
        r=temp%10;
        sum=sum+r*r*r;
        temp=temp/10;
    }
    if(sum==i)
        printf("%d  ",i);
    sum=0;
}

return 0;
}
```

Result:

```
enter initial number:1
enter final number:1000
1  153  370  371  407
```

Counting numbers of a digit in an integer code:

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

{
    int num, count=0;

    printf("enter any integer:");
    scanf("%d",&num);

    while(num!=0)
```

```
    {
        num=num/10;
        ++count;
    }
    printf("total number of digits:%d\n",count);

    return 0;
}
```

Result:

```
enter any integer:34567
total number of digits:5
```

*****A **Strong number** is a number for which the sum of the factorials of its digits equals the number itself.

A number N is a Strong number if:

$$N = d_1! + d_2! + \cdots + d_k!$$

- Digits: $1, 4, 5$
- $1! + 4! + 5! = 1 + 24 + 120 = 145$

Strong number code:

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

{
    int num, sum=0,rem,temp,fact,i;

    printf("enter an integer:");
    scanf("%d",&num);

    temp=num;

    while(temp!=0)
    {
```

```
        rem = temp % 10;

        fact=1;
    for(i=1;i<=rem;i++)
    {
        fact=fact*i;
    }
        sum =sum+fact;
        temp=temp/10;
    }
    if(sum==num)
        printf("%d is a STRONG NUMBER",num);
    else
        printf("%d is not a STRONG NUMBER",num);

    return 0;
}
```

Result:

```
enter an integer:145
145 is a STRONG NUMBER
```

```
enter an integer:143
143 is not a STRONG NUMBER
```

Series

1+2+3+……+n(for loop)

Code:

```
#include<stdio.h>
int main()
{
    int n, sum=0, i;
```

```
    printf("enter the last number of the series:");
    scanf("%d",&n);

    printf("1+2+3.....+%d",n);

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

    return 0;
}
```

Result:

```
enter the last number of the series:10
1+2+3.....+10=55
```

1+2+3+……..+n(while loop)

Code:

```
#include<stdio.h>
int main()
{
    int n, sum=0, a=1;
    printf("enter the last nummber:");
    scanf("%d",&n);

    printf("1+2+3....+%d",n);

    while (a<=n)
    {
        sum=sum+a;
        a=a+1;
    }

    printf("= %d",sum);
```

```
    return 0;

}
```

Result:

```
enter the last nummber:10
1+2+3....+10= 55
```

1.2+2.3+3.4+……..+n1.n2

Code:

```
#include<stdio.h>
int main()
{
    int n1,n2,sum=0,a=1,b=2;

    printf("enter n1 and n2:");
    scanf("%d %d",&n1,&n2);

    printf("1.2+2.3+3.4+.....+n1.n2= ");

    while (a<=n1 && b<=n2)
    {
        sum=sum+ a*b;
        a=a+1;
        b=b+1;
    }
    printf("%d\n",sum);

    return 0;

}
```

Result:

```
enter n1 and n2:4 5
1.2+2.3+3.4+.....+n1.n2= 40
```

Printing 1 to n with for loop code:

```c
#include<stdio.h>
int main()
{
    int n,i;

    printf("enter n:");
    scanf("%d",&n);

    for(i=1;i<=n;i++)
    printf("%d  ",i);

    return 0;
}
```

Result:

```
enter n:100
1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38
 39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56
  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  7
4  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91
92  93  94  95  96  97  98  99  100
```


1.5+2.5+3.5+……..+n code:

```c
#include<stdio.h>
int main()
{
    float n,i,sum=0;

    printf("enter n:");
    scanf("%f",&n);

    printf("1.5+2.5+3.5+……..+n");

    for(i=1.5;i<=n;i++)
    {
        sum=sum+i;
    }
    printf("=%f\n",sum);

    return 0;
```

```
}
```

Result:

```
enter n:4
1.5+2.5+3.5+....+n=7.500000
```

1^2+3^2+5^2.....+n^2 code:

```c
#include<stdio.h>
int main()
{
    int n, sum=0, i;

    printf("enter n:");
    scanf("%d",&n);

    printf("1^2+3^2+5^2.....+%d^2",n);

    for(i=1; i<=n; i=i+2)
    {
        sum=sum+(i*i);
    }
    printf("=%d\n",sum);

    return 0;
}
```

Result:

```
enter n:9
1^2+3^2+5^2.....+9^2=165
```

1+1/2+1/3+…..+1/n code:

```c
#include<stdio.h>
int main()
{
    double n, sum=0, i;
```

```
    printf("enter n:");
    scanf("%lf",&n);

    for(i=1; i<=n; i++)
    {
        sum=sum+(1/i);

        if(i==1)
        printf("1 + ");

        else if(i==n)
        printf("(1/%lf)",i);

        else
        printf("(1/%lf) + ",i);
    }
    printf("=%.2lf\n",sum);

    return 0;
}
```

Result:

```
enter n:4
1 + (1/2.000000) + (1/3.000000) + (1/4.000000)=2.08
```

1x2x3x…….xn code:

```
#include<stdio.h>
int main()
{
    int n, result=1, i;

    printf("enter n:");
    scanf("%d",&n);

    printf("1x2x3x.....x%d",n);

    for(i=1; i<=n; i=i+1)
    {
        result=result*i;
```

```
    }
    printf("=%d\n",result);

    return 0;
}
```

Result:

```
enter n:4
1x2x3x.....x4=24
```

1^2x2^2x3^2x__xn^2 code:

```
#include<stdio.h>
int main()
{
    int n, result=1, i;

    printf("enter n:");
    scanf("%d",&n);

    printf("1^2x2^2x3^2x.....x%d^2",n);

    for(i=1; i<=n; i=i+1)
    {
        result=result*i*i;  //cube er jonno 3 bar i multiply korte hobe
    }
    printf("=%d\n",result);

    return 0;
}
```

Result:

```
enter n:6
1^2x2^2x3^2x.....x6^2=518400
```

1-2+3-4+5-6+……+n code:

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

```
    int n,i,even=0,odd=0;

    printf("Enter the last term:");
    scanf("%d",&n);

    for(i=1;i<=n;i++)
    {
        if(i%2==0)
            even=even+i;
        else
            odd=odd+i;
    }
    printf("sum=%d\n",odd-even);

    return 0;
}
```

Result:

```
Enter the last term:6
sum=-3
```

Fibonacci numbers: The Fibonacci numbers are a sequence of numbers where each number is the sum of the two preceding ones, usually starting with 0 and 1. The sequence goes as follows:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

```
#include<stdio.h>
int main()
{
    int first=0,second=1,count=0,fibo,n;

    printf("enter range :");   //range=n
    scanf("%d",&n);

    while(count<n)
    {
        if(count<=1)
```

```
            fibo=count;
        else
        {
            fibo=first+second;
            first=second;
            second=fibo;
        }
        printf("%d ",fibo);
        count++;
    }
    return 0;
}
```

Result:

```
enter range :8
0 1 1 2 3 5 8 13
```

Lucas series: 2,1,3,4,7,11,18,29,47,76,…

Code:

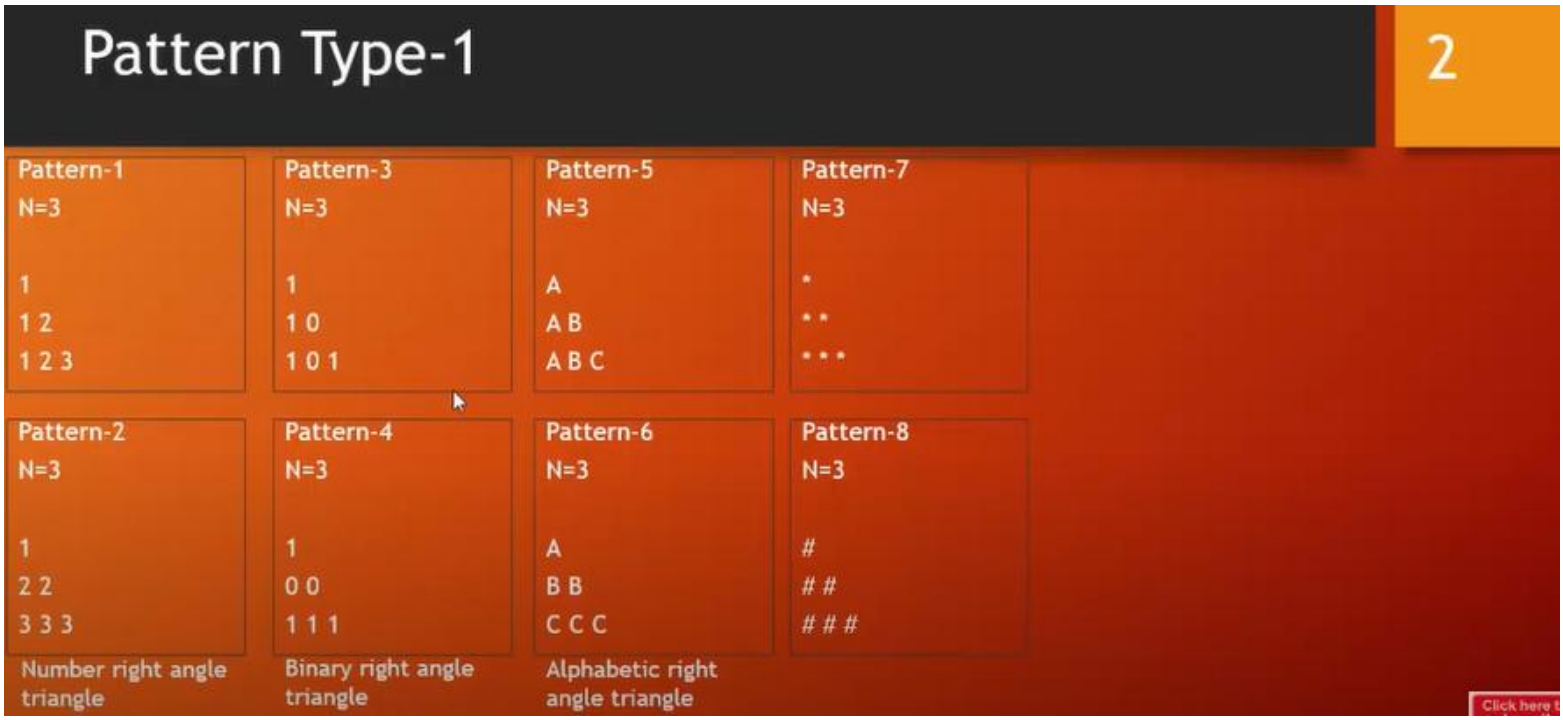
```
#include<stdio.h>
int main()
{
    int numTerms,first=2,second=1,lucas,i;

    printf("enter the number of terms in lucas series:");   //range=n
    scanf("%d",&numTerms);

    printf("%d %d ",first,second);

    for(i=3; i <= numTerms; i++)
    {
        lucas=first+second;

        printf("%d ",lucas);

        first=second;
        second=lucas;
    }
```

```
    return 0;
}
```

Result:

```
enter the number of terms in lucas series:5
2 1 3 4 7
```

Pattern



Type-1_Right angle Triangle

Pattern 1_type 1;n=3:

1

1 2

1 2 3

Code:

```c
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=1; row<=n; row++)
    {
        for(col=1; col<=row; col++)
        {
            printf("%d ",col);
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
1
1 2
1 2 3
```

Pattern 2_type1; n=3:

1

2 2

3 3 3

Code:

```c
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
```

```
    scanf("%d",&n);

    for(row=1; row<=n; row++)
    {
        for(col=1; col<=row; col++)
        {
            printf("%d ",row);
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
1
2 2
3 3 3
```

Pattern 3_type1; n=3:

1

1 0

1 0 1

Code:

```
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=1; row<=n; row++)
    {
```

```
        for(col=1; col<=row; col++)
        {
            printf("%d ",col%2);
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
1
1 0
1 0 1
```

Or,

```
enter n=5
1
1 0
1 0 1
1 0 1 0
1 0 1 0 1
```

Pattern 4_type1; n=3:

1

0 0

1 1 1

Code:

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

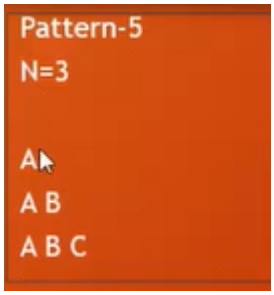
```
    int n,row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=1; row<=n; row++)
    {
        for(col=1; col<=row; col++)
        {
            printf("%d ",row%2);
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
1
0 0
1 1 1
```

Or,

```
enter n=5
1
0 0
1 1 1
0 0 0 0
1 1 1 1 1
```



_type1

*****A=65(asci value); a=97

Code:

```c
#include<stdio.h>
int main()
{
    int n, row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=1; row<=n; row++)
    {
        for(col=1; col<=row; col++)
        {
            printf("%c ",col+64);  //A=65;a=97
        }
        printf("\n");
    }

    return 0;
}
```

Return:

```
enter n=3
A
A B
A B C
```

_type1

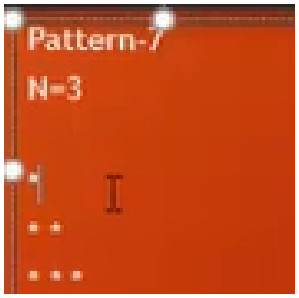
Code:

```c
#include<stdio.h>
int main()
{
    int n, row, col;
```

```
    printf("enter n=");
    scanf("%d",&n);

    for(row=1; row<=n; row++)
    {
        for(col=1; col<=row; col++)
        {
            printf("%c ",row+64);  //A=65;a=97
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
A
B B
C C C
```



_type1

Code:

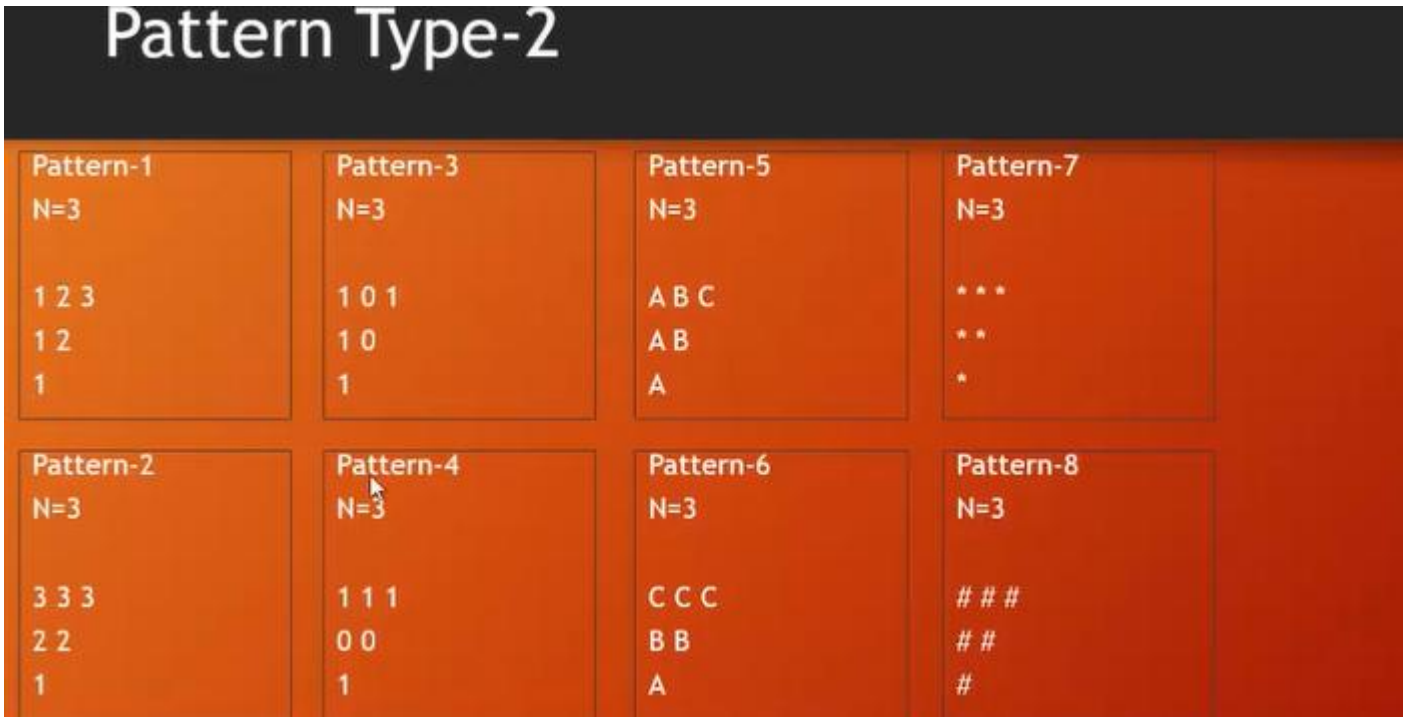
```
#include<stdio.h>
int main()
{
    int n, row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=1; row<=n; row++)
```

```
    {
        for(col=1; col<=row; col++)
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
*
* *
* * *
```

Type-2

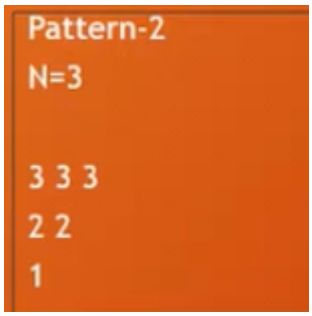


Code:

```c
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=n; row>=1; row--)
    {
        for(col=1; col<=row; col++)
        {
            printf("%d ",col);
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
1 2 3
1 2
1
```



Code:

```c
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=n; row>=1; row--)
    {
        for(col=1; col<=row; col++)
        {
            printf("%d ",row);
        }
        printf("\n");
    }

    return 0;
}
```

Result:

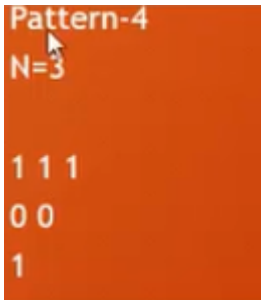
```
enter n=3
3 3 3
2 2
1
```

Code:

```c
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=n; row>=1; row--)
    {
        for(col=1; col<=row; col++)
        {
            printf("%d ",col%2);
        }
        printf("\n");
    }

    return 0;
}
```

Result:

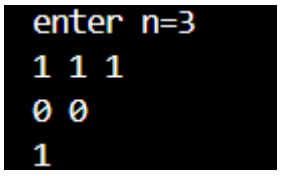
```
enter n=3
1 0 1
1 0
1
```

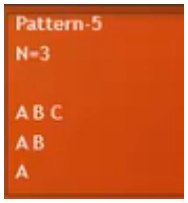


Code:

```c
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=n; row>=1; row--)
    {
        for(col=1; col<=row; col++)
        {
            printf("%d ",row%2);
        }
        printf("\n");
    }

    return 0;
}
```

Result:





Code:

```c
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
    scanf("%d",&n);

    for(row=n; row>=1; row--)
    {
        for(col=1; col<=row; col++)
        {
            printf("%c ",col+64);
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
A B C
A B
A
```

Code:

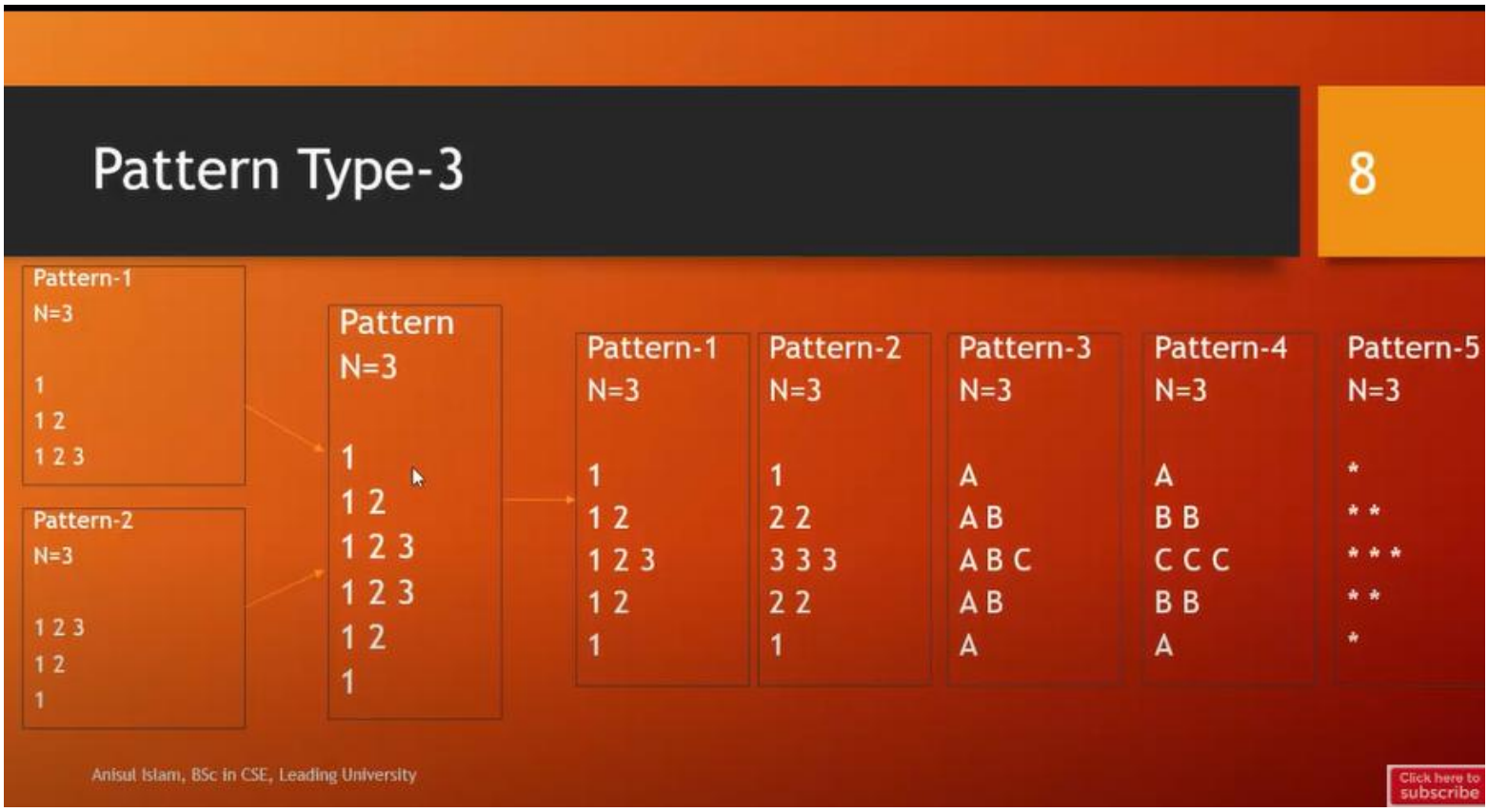
```c
#include<stdio.h>
int main()
{
    int n,row, col;

    printf("enter n=");
```

```
    scanf("%d",&n);

    for(row=n; row>=1; row--)
    {
        for(col=1; col<=row; col++)
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

Result:

```
enter n=3
* * *
* *
*
```

Due from vid155 to vid



Pattern-1
N=3

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

*pattern3.c

```
    int n,row,col;

    printf("Enter N= ");
    scanf("%d",&n);

    for(row=1;row<=n;row++)
    {
        for(col=1;col<=row;col++)
        {
            printf("%d ",col);
        }
        printf("\n");

    }

    for(row=n-1;row>=1;row--)
```

*pattern3.c

```
    }

    for(row=n-1;row>=1;row--)
    {
        for(col=1;col<=row;col++)
        {
            printf("%d ",col);
        }
        printf("\n");

    }

}
```

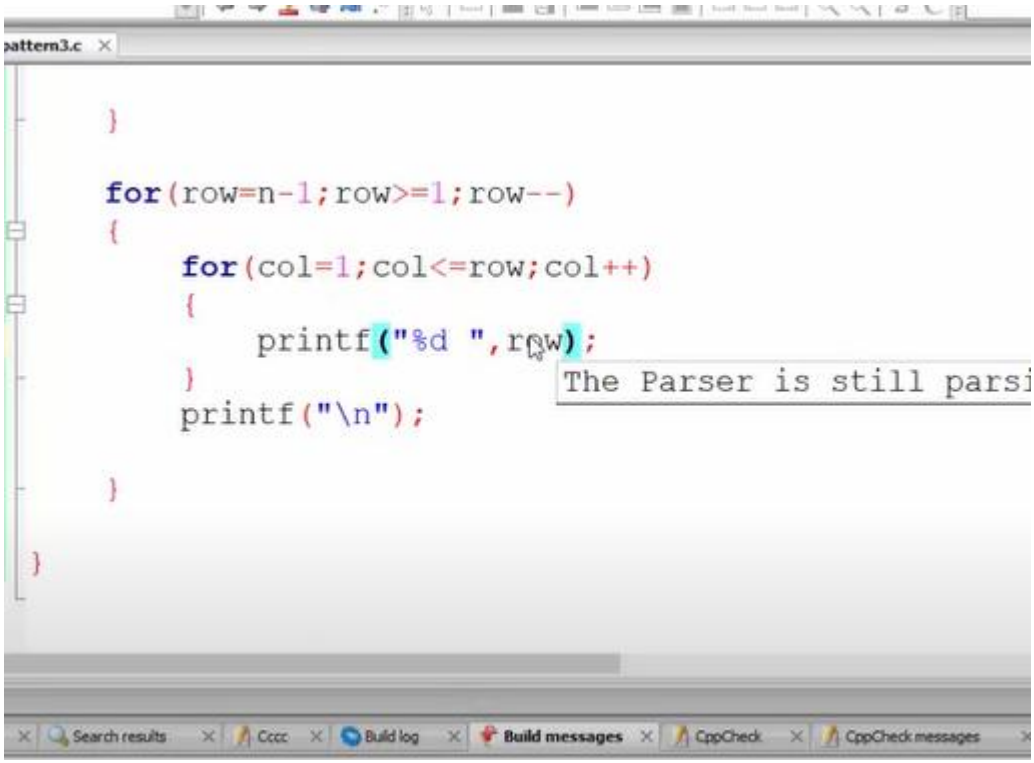
Search results | Cccc | Build log | Build messages | CppCheck

Pattern-2
N=3

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

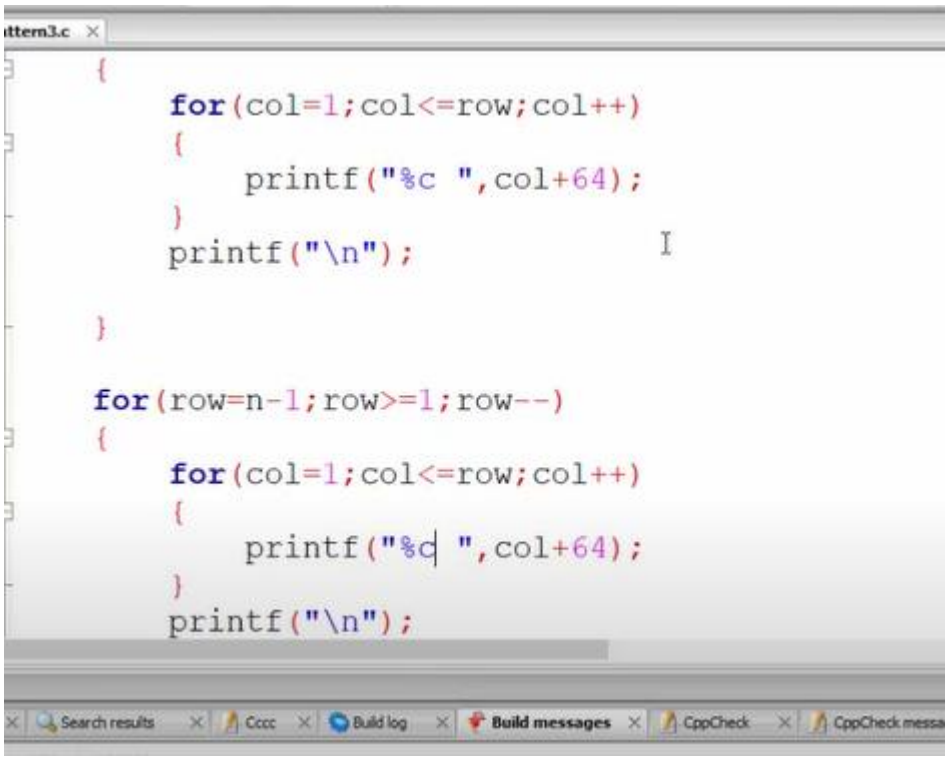
pattern3.c

```c
int n,row,col;

printf("Enter N= ");
scanf("%d",&n);

for(row=1;row<=n;row++)
{
    for(col=1;col<=row;col++)
    {
        printf("%d ",row);
    }
    printf("\n");

}

for(row=n-1;row>=1;row--)
```



Pattern-3
N=3

A
A B
A B C
A B
A



```
ttern3.c
    {
        for(col=1;col<=row;col++)
        {
            printf("%c ",col+64);
        }
        printf("\n");

    }

    for(row=n-1;row>=1;row--)
    {
        for(col=1;col<=row;col++)
        {
            printf("%c ",col+64);
        }
        printf("\n");
```

Search results | Cccc | Build log | Build messages | CppCheck | CppCheck messa

Pattern-5
N=3

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

```c
{

    int n,row,col;

    printf("Enter N= ");
    scanf("%d",&n);

    for(row=1;row<=n;row++)
    {
        for(col=1;col<=row;col++)
        {
            printf("* ");
        }
        printf("\n");

    }

    for(row=n-1;row>=1;row--)
```

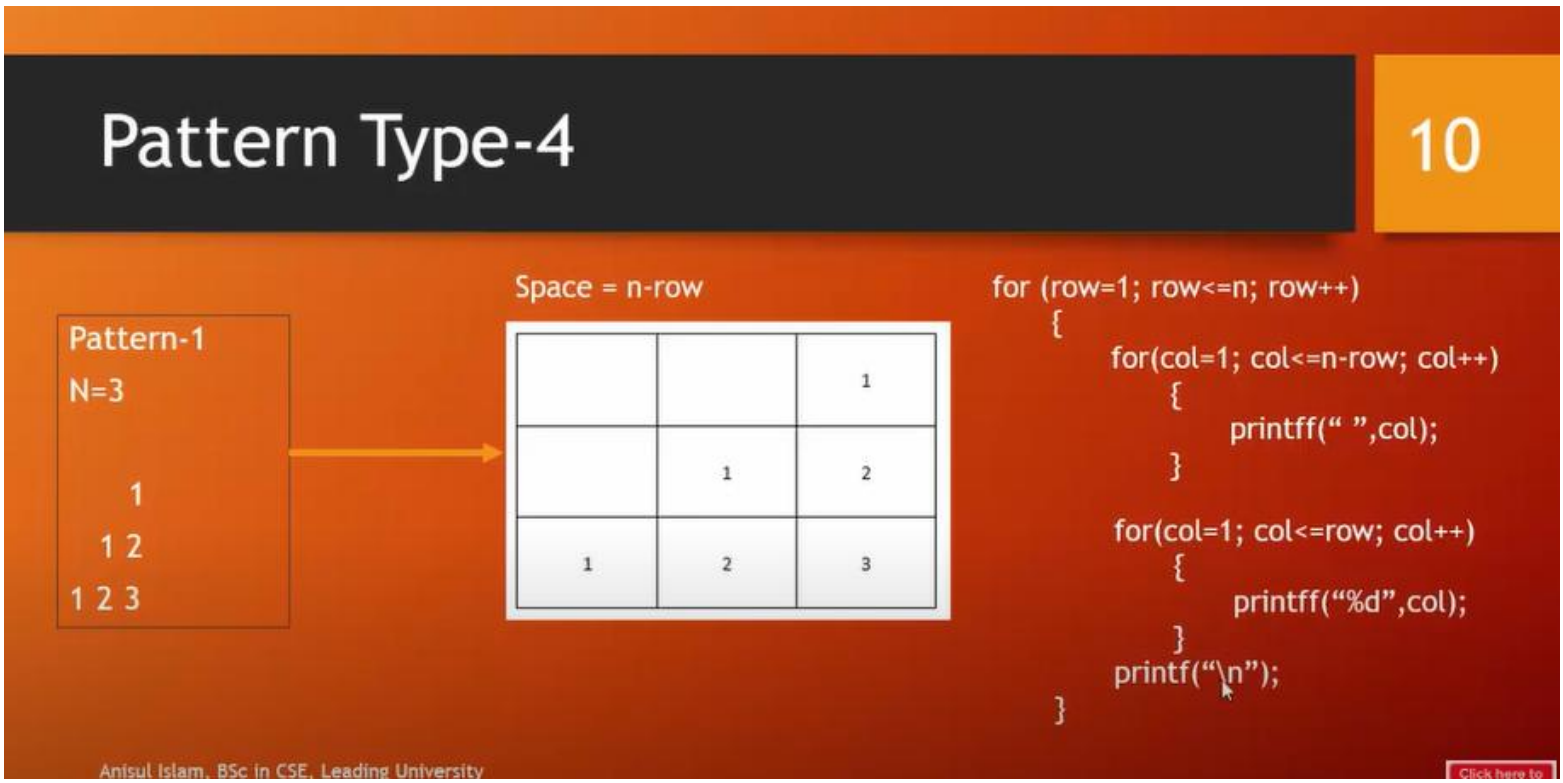
```
C:\Users\Dell\Desktop\pattern3.exe
Enter N= 5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

Process returned 10 (0xA)   execution time : 2.641 s
Press any key to continue.
```

```c
ttern3.c ×
        }
        printf("\n");

    }

    for(row=n-1;row>=1;row--)
    {
        for(col=1;col<=row;col++)
        {
            printf("* ");
        }
        printf("\n");

    }

}
```

```
C:\Users\Dell\Desktop\pattern3.exe
Enter N= 5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

Process returned 10 (0xA)   execution time
Press any key to continue.
```

Search results | Cccc | Build log | Build messages | CppC

Line Message



Array

Code:

```c
#include<stdio.h>
int main()
{
    int numbers[5]; //array initialization

    numbers[0]=10;
    numbers[1]=20;
    numbers[2]=30;
    numbers[3]=40;
    numbers[4]=50;

    printf("%d\n", numbers[0]);
```

```
    printf("%d\n", numbers[1]);
    printf("%d\n", numbers[2]);
    printf("%d\n", numbers[3]);
    printf("%d\n", numbers[4]);

    return 0;

}
```

Result:

****array declare korar shomoy e array er value assign korar jonno curly braces er moddhe value assign kora jabe, value assign korle koita value ta age declare korar dorkar nei

Code:

```
#include<stdio.h>
int main()
{
    //array initialization and declaration
    int numbers[] = {10,20,30,40,50};

    for(int i=0; i<=4; i++)
    {
    printf("%d\n", numbers[i]);
    }

    return 0;

}
```

Result:

```
10
20
30
40
50
```


****user er kach theke input newar jonno scanf() use kora jabe

Code:

```c
#include<stdio.h>
int main()
{
    //array initialization and declaration
    int numbers[5];

    for(int i=0; i<=4; i++)
    {
        printf("enter number %d : ",i);
        scanf("%d", &numbers[i]);
    }

    for(int i=0; i<=4; i++)
    {
        printf("%d\n", numbers[i]);
    }

    return 0;

}
```

Result:

```
enter number 0 : 10
enter number 1 : 20
enter number 2 : 30
enter number 3 : 40
enter number 4 : 50
10
20
30
40
50
```

*****sum and average of numbers in array

Code:

```c
#include<stdio.h>
int main()
{
    //array initialization and declaration
    int numbers[5], sum=0;
    float average;

    for(int i=0; i<=4; i++)
    {
        printf("enter number %d : ",i);
        scanf("%d", &numbers[i]);
    }

    for(int i=0; i<=4; i++)
    {
        sum = sum + numbers[i];
    }

    printf("the sum is %d\n", sum);

    average = (float)sum/5;
    printf("the average is : %.4f\n", average);

    return 0;

}
```

Result:

```
enter number 0 : 20
enter number 1 : 24
enter number 2 : 7
enter number 3 : 54
enter number 4 : 37
the sum is 142
the average is : 28.4000
```

*****maximum and minimum number in array

Code:

```c
#include<stdio.h>
int main()
{
    int numbers[] = {20, 40, 1, 100, 98, -4};

    int max = numbers[0];
    int min = numbers[0];

    for(int index=1; index<6; index++)
    {
    if(max<numbers[index])
    {
        max = numbers[index];
    }
    if(min>numbers[index])
    {
        min = numbers[index];
    }
    }

    printf("Maximum number is : %d\n", max);
    printf("Minimum number is : %d\n", min);

    return 0;

}
```

Result:

```
Maximum number is : 100
Minimum number is : -4
```

Linear search:

```c
#include<stdio.h>
int main()
{
    int numbers[] = {20, 40, 1, 100, 98, -4};
```

```
    int searchnumber = 100;
    int found = -1;

    for(int index=0; index<6; index++)
    {
        if(numbers[index] == searchnumber)
        {
        found = index;
        break;
        }
    }
    if(found == -1)
    {
        printf("%d is Not Found",searchnumber);
    }
    else
        printf("%d is Found in position %d ", searchnumber, found);

    return 0;
}
```

Result:

```
100 is Found in position 3
```

```
500 is Not Found
```

First second largest number(unsorted):

Code:

```
#include<stdio.h>
int main()
{
    int numbers[] = {15, 14, 18};

    int lengthOfArray = sizeof(numbers)/sizeof(numbers[0]);

    if(lengthOfArray<2)
    {
        printf("Array should have atleast 2 elements.\n");
    }
```

```
    int first, second;
    if(numbers[1]>numbers[0])
    {    first=numbers[1];
        second=numbers[0];
    }
    else
    {
        first=numbers[0];
        second=numbers[1];
    }

    for(int index=2; index<lengthOfArray; index++)
    {
        if(numbers[index]>first)
        {
            second=first;
            first=numbers[index];
        }
        else if(numbers>second && numbers[index]!=first)
        {
            second=numbers[index];
        }
    }

    printf("First largest : %d\n", first);
    printf("Second largest : %d\n", second);

    for(int index=0; index<lengthOfArray; index++)
    printf("%d ", numbers[index]);

    return 0;
}
```

Result:

```
First largest : 18
Second largest : 15
15 14 18
```

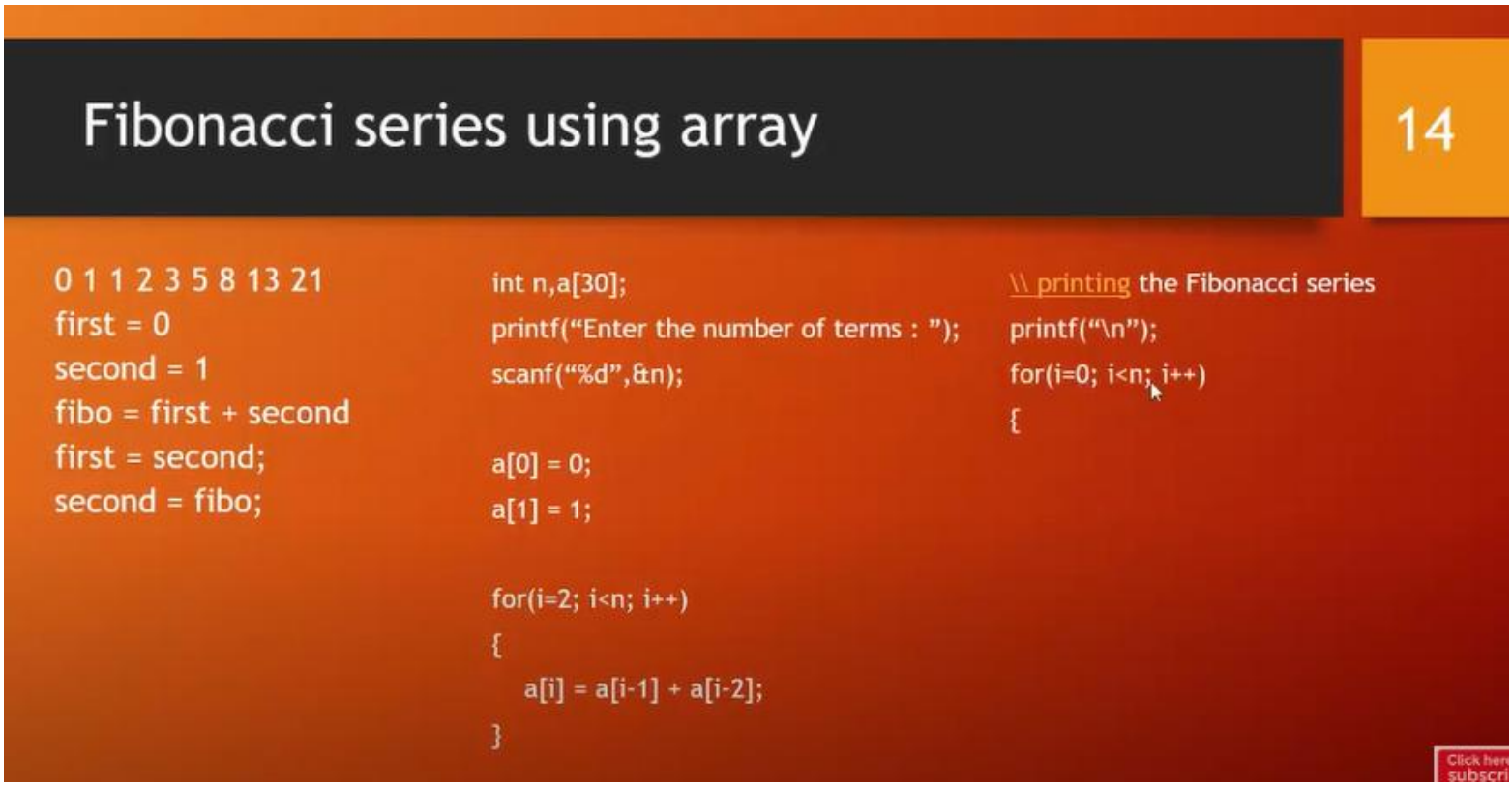
First second largest number(sorted):

Code:

```c
#include<stdio.h>
int main()  //sorted array
{
    int numbers[] = {10, 20, 30, 40};

    int lengthOfArray = sizeof(numbers)/sizeof(numbers[0]);

    if(lengthOfArray<2)
    {
        printf("Array should have atleast 2 elements.\n");
    }

    int first = numbers[lengthOfArray-1];
    int second = numbers[lengthOfArray-2];



    printf("First largest : %d\n", first);
    printf("Second largest : %d\n", second);

    for(int index=0; index<lengthOfArray; index++)
    printf("%d ", numbers[index]);

    return 0;
}
```

Result:

```
First largest : 40
Second largest : 30
10 20 30 40
```



Copy all elements of an array to another array

Code:

```c
#include<stdio.h>
int main()
{
    int array1[] = {10, 20, 30, 40, 50}, array2[5], i;

printf("Array1 : ");

for(i=0; i<5; i++)
    printf("%d ",array1[i]);

for(i=0; i<5; i++)
    array2[i] = array1[i];

printf("\nArray2 : ");

for(i=0; i<5; i++)
    printf("%d ",array2[i]);

return 0;
```

```
}
```

Result:

```
Array1 : 10 20 30 40 50
Array2 : 10 20 30 40 50
```

Copy paste array by taking input from user:

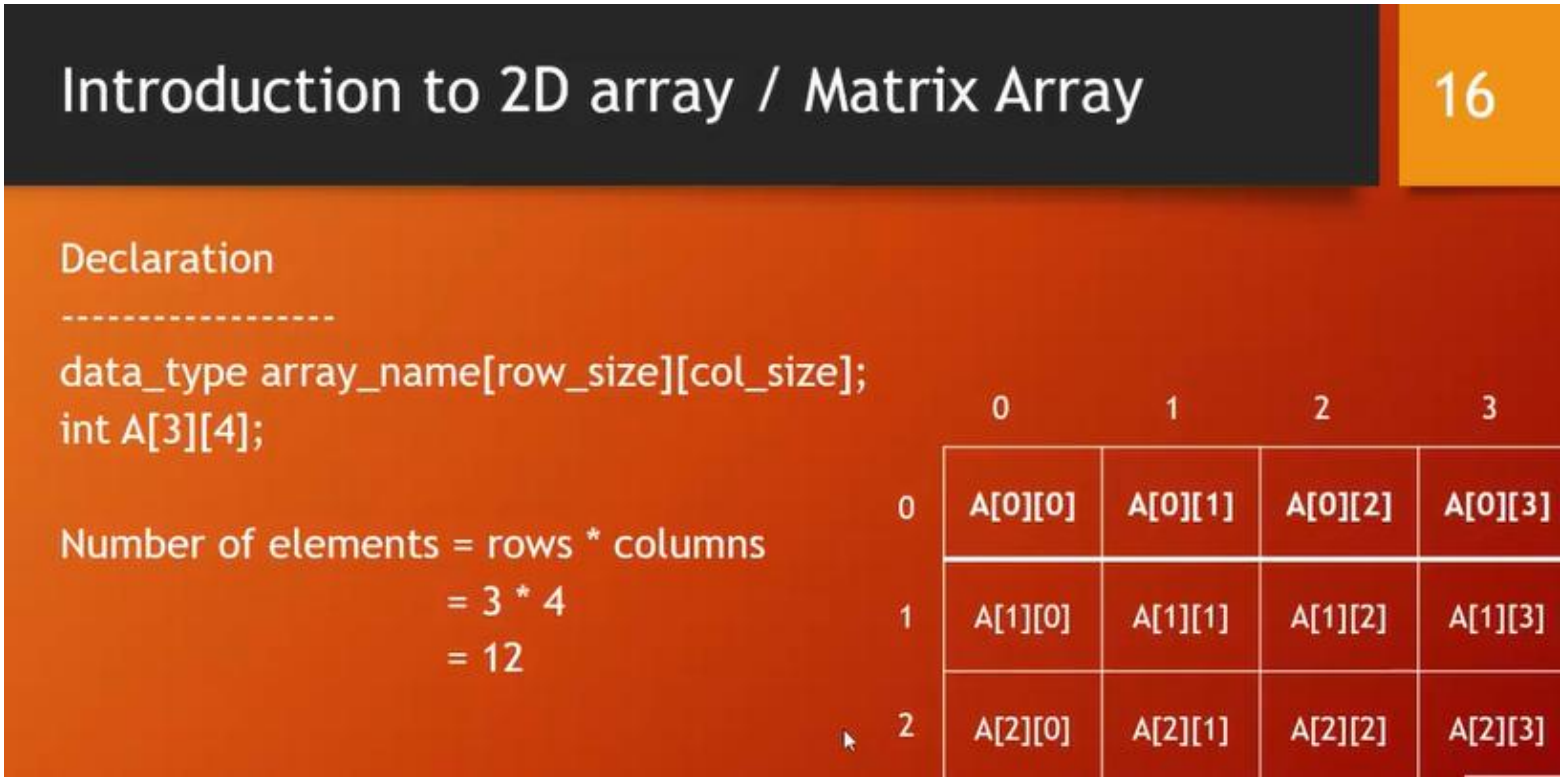
Code:

```
#include<stdio.h>
int main()
{
    int array1[30], array2[30], n, i;

printf("How many numbers : ");
scanf("%d",&n);

for(i=0; i<n; i++)
    scanf("%d",&array1[i]);

printf("Array1 : ");

for(i=0; i<n; i++)
    printf("%d ",array1[i]);

for(i=0; i<n; i++)
    array2[i] = array1[i];

printf("\nArray2 : ");

for(i=0; i<n; i++)
    printf("%d ",array2[i]);

return 0;

}
```

Result:

```
How many numbers : 6
10 20 30 40 50 60
Array1 : 10 20 30 40 50 60
Array2 : 10 20 30 40 50 60
```

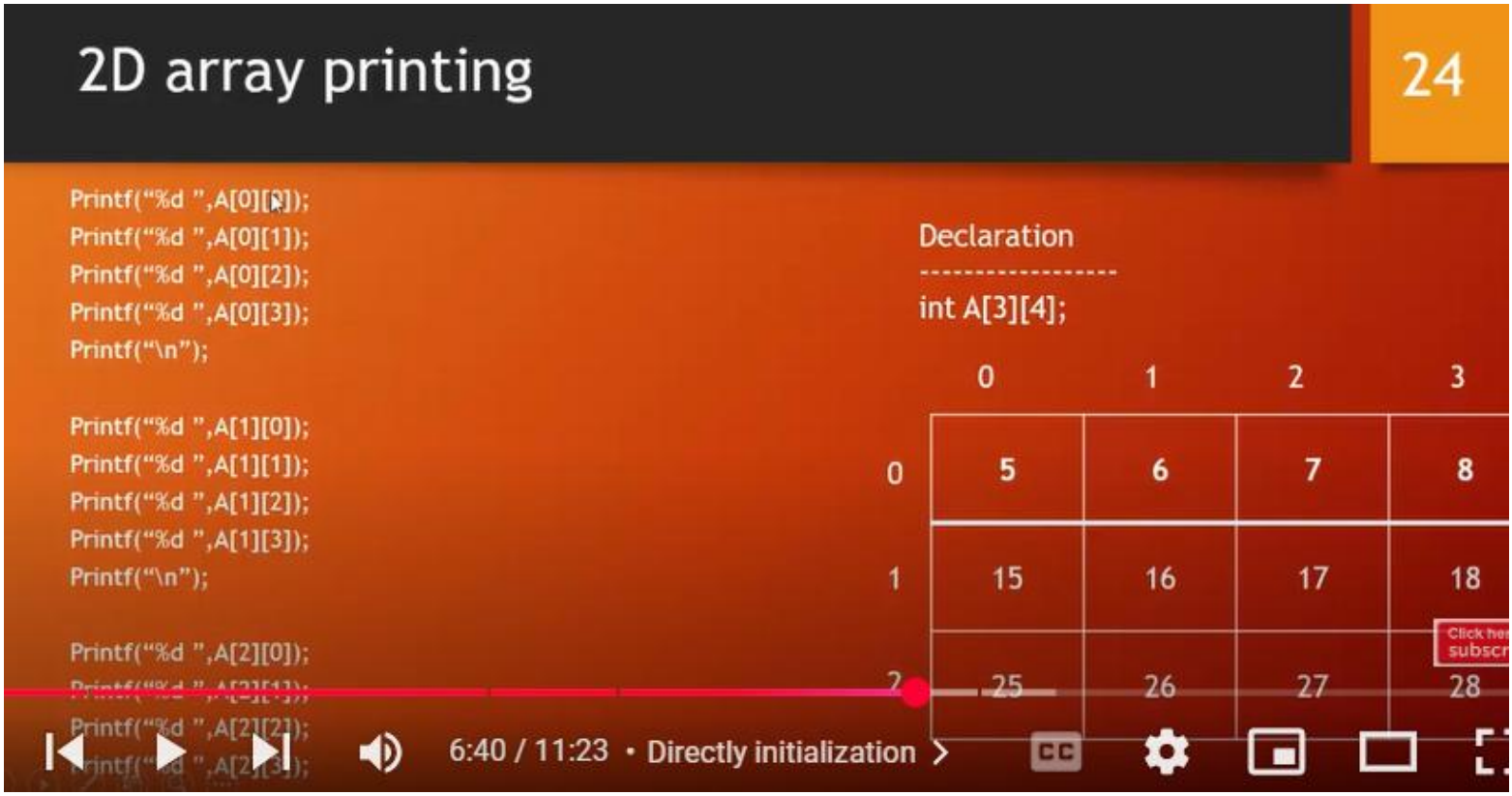
2d array:



Emon alada kore initialize kora jabe

Or directly sobgula initialize kora jabe

Print kora



Loop use kore print kora

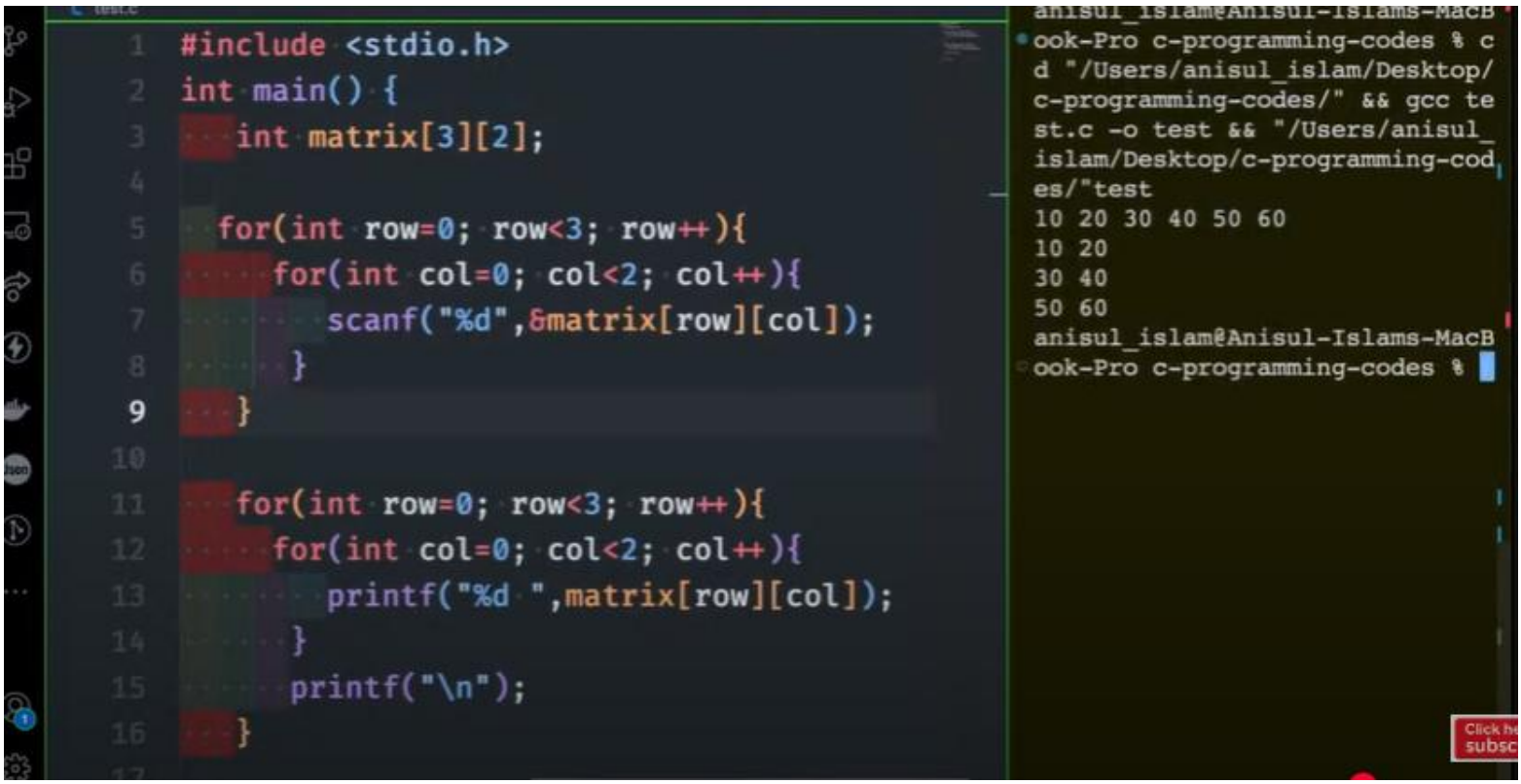
Code:

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

    printf("%d ",matrix[0][0]);
    printf("%d ",matrix[0][1]);
    printf("%d ",matrix[1][0]);
    printf("%d ",matrix[1][1]);
    printf("%d ",matrix[2][0]);
    printf("%d ",matrix[2][1]);

    return 0;
}
```

Loop diye output

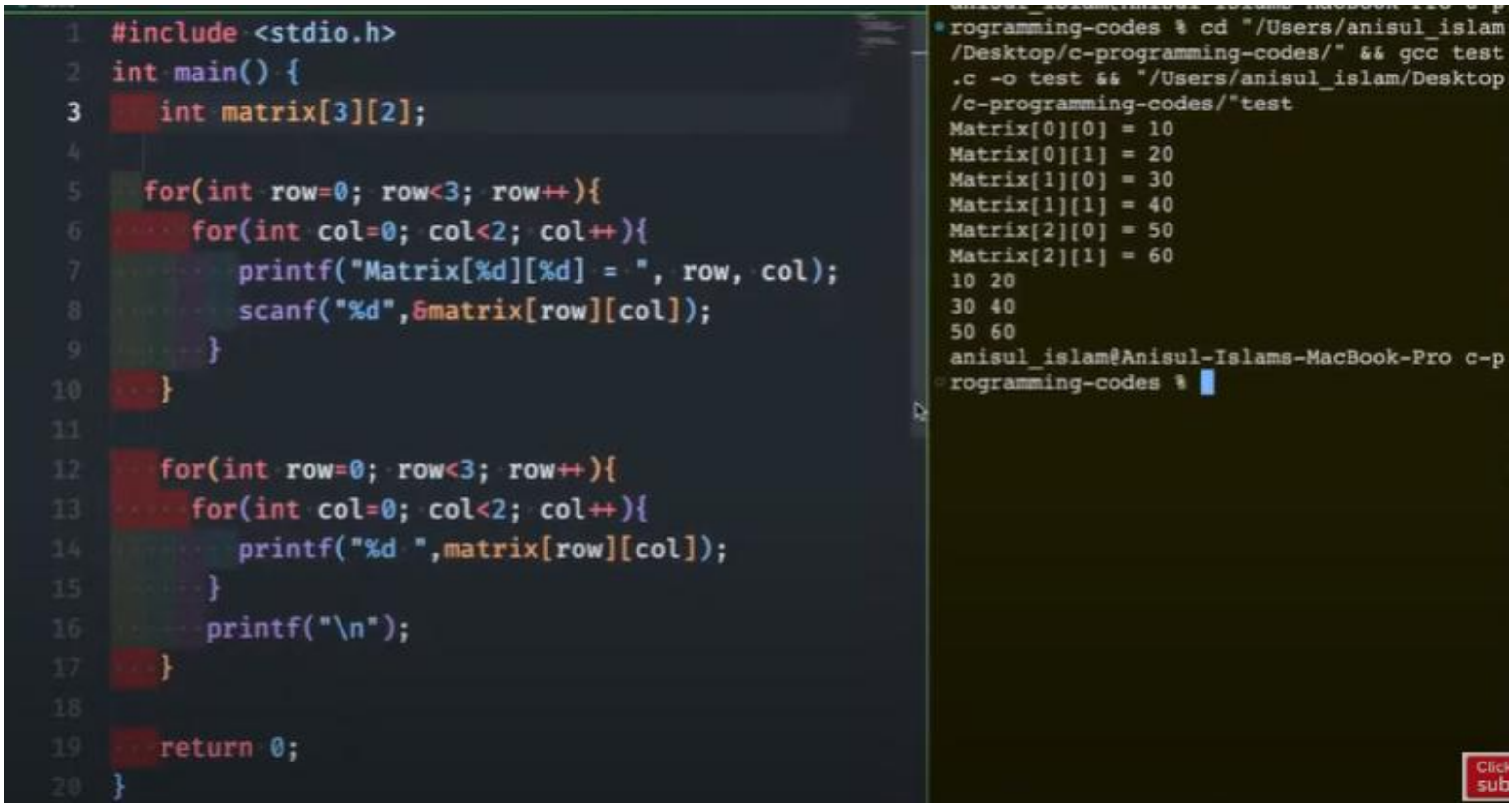
```c
#include <stdio.h>
int main() {
    int matrix[3][2] = {
     {10,20},
     {30,40},
     {50,60}
    };

    for(int row=0; row<3; row++){
        for(int col=0; col<2; col++){
            printf("%d ",matrix[row][col]);
        }
        printf("\n");
    }

    return 0;
}
```

User er kach theke input niye



```c
#include <stdio.h>
int main() {
    int matrix[3][2];

    for(int row=0; row<3; row++){
        for(int col=0; col<2; col++){
            scanf("%d",&matrix[row][col]);
        }
    }

    for(int row=0; row<3; row++){
        for(int col=0; col<2; col++){
            printf("%d ",matrix[row][col]);
        }
        printf("\n");
    }
```

```
ook-Pro c-programming-codes % c
d "/Users/anisul_islam/Desktop/
c-programming-codes/" && gcc te
st.c -o test && "/Users/anisul_
islam/Desktop/c-programming-cod
es/"test
10 20 30 40 50 60
10 20
30 40
50 60
anisul_islam@Anisul-Islams-MacB
ook-Pro c-programming-codes %
```



```c
#include <stdio.h>
int main() {
    int matrix[3][2];

    for(int row=0; row<3; row++){
        for(int col=0; col<2; col++){
            printf("Matrix[%d][%d] = ", row, col);
            scanf("%d",&matrix[row][col]);
        }
    }

    for(int row=0; row<3; row++){
        for(int col=0; col<2; col++){
            printf("%d ",matrix[row][col]);
        }
        printf("\n");
    }

    return 0;
}
```

```
rogramming-codes % cd "/Users/anisul_islam
/Desktop/c-programming-codes/" && gcc test
.c -o test && "/Users/anisul_islam/Desktop
/c-programming-codes/"test
Matrix[0][0] = 10
Matrix[0][1] = 20
Matrix[1][0] = 30
Matrix[1][1] = 40
Matrix[2][0] = 50
Matrix[2][1] = 60
10 20
30 40
50 60
anisul_islam@Anisul-Islams-MacBook-Pro c-p
rogramming-codes %
```

String

Code:

```c
#include<stdio.h>
int main()
{
    char s1[5]; //null character er jonno string er size ek beshi dite hobe

    s1[0]='A';
    s1[1]='n';
    s1[2]='i';
    s1[3]='s';
    s1[4]='\0'; // \0  holo null character

    printf("s1 = %s\n",s1);

    return 0;

}
```

Result:

```
s1 = Anis
```

Directly initialize kora code:

```c
//directly initialize korte
#include<stdio.h>
int main()
{
    char s1[]={'A', 'n', 'i', 's', '\0'};

    char s2[]="Anisul Islam"; //duibhabei kora jabe

    printf("s1 = %s\n",s1);
    printf("s2 = %s\n",s2);

    return 0;
```

```
}
```

Result:

```
s1 = Anis
s2 = Anisul Islam
```

String input and display code:

```c
#include<stdio.h>
int main()
{
    char s1[30];

    printf("Enter your full name : ");
    gets(s1);  //scanf() space er porer part tuku input nite parena tai gets() function use korte hobe

    printf("Full name = %s\n", s1);

    return 0;

}
```

Result:

```
Enter your full name : Adnan Abir Rangan
Full name = Adnan Abir Rangan
```

Display srtring characterwise code:

```c
#include<stdio.h>
int main()
{
    char s1[]="Suparna";

    int i=0;

    while(s1[i]!='\0')
    {
```

```
        printf("%c\n",s1[i]);
        i++;
    }
    return 0;
}
```

Result:

```
S
u
p
a
r
n
a
```

finding length of String using strlen() function

code:

```
#include<stdio.h>
int main() //using strlen() function
{
    char s1[]="Adnan Abir Rangan";

    int len = strlen(s1);

    printf("Length = %d\n",len);

    return 0;
}
```

Result:

```
Length = 17
```

finding length of String without strlen() function

Code:

```c
#include<stdio.h>
int main()
{
    char s1[]="Adnan Abir Rangan";

    int i=0, len=0;

    while(s1[i]!='\0')
    {
        i++;
        len++;
    }

    printf("Length = %d\n",len);

    return 0;
}
```

Result:

```
Length = 17
```

copy string using strcpy()

code:

```c
#include<stdio.h>
int main()
{
    char source[]="C Programming";
    char target[20];

    strcpy(target,source);

    printf("Source string = %s\n",source);
    printf("Target string = %s\n",target);

    return 0;
}
```

Result:

```
Source string = C Programming
Target string = C Programming
```

Duita string jog kora;

concatenation using strcat()

code:

```c
#include<stdio.h>
int main()
{
    char str1[]="my name is ";
    char str2[]="Adnan Abir Rangan";

    strcat(str1,str2);

    printf("str1 = %s\n",str1);
    printf("str2 = %s\n",str2);

    return 0;
}

/* OR---
#include<stdio.h>
int main()
{
    char str1[]="my name is ";

    strcat(str1,"Adnan Abir Rangan");

    printf("str1 = %s\n",str1);

    return 0;
}

*/
```

Result:

```
str1 = my name is Adnan Abir Rangan
str2 = Adnan Abir Rangan
```

String | concatenation without strcat()

Code:

```c
#include<stdio.h>
int main()
{
    char str1[50] = "my name is ";
    char str2[] = "Adnan Abir Rangan";

    int i=0, len=0, j=0;

    while(str1[i]!='\0')
    {
        i++;
        len++;
    }

    while(str2[j]!='\0')
    {
        str1[len+j] = str2[j];
        j++;
    }

    printf("str1 = %s\n",str1);

    return 0;
}
```

Result:

```
str1 = my name is Adnan Abir Rangan
```

String compare using strcmp() function

Code:

```c
#include<stdio.h>
int main()
{
    char str1[] = "Adnan Abir Rangan";
    char str2[] = "Adnan";
```

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

    if(d==0)
    {
        printf("strings are equal");
    }
    else
        printf("strings are not equal");

    return 0;

}
```

Result:

```
strings are not equal
```

String reverse using strrev()

Code:

```
#include<stdio.h>
int main()
{
    char str[] = "adnan abir rangan";
    printf("str = %s\n",str);
    strrev(str);

    printf("str = %s\n",str);

    return 0;
}
```

Result:

```
str = adnan abir rangan
str = nagnar riba nanda
```

String reverse without strrev()

Code:

```c
#include<stdio.h>
int main()
{
    char str1[30] = "adnan abir rangan";
    char str2[30];

    int i=0,len=0,j;
    while(str1[i]!='\0')
    {
        i++;
        len++;
    }
    for (j=0,i=len-1;i>=0;i--,j++)
    {
        str2[j] = str1[i];
    }
    str2[j] = '\0';

    printf("str1 = %s\n",str1);
    printf("str2 = %s\n",str2);

    return 0;
}
```

Result:

```
str1 = adnan abir rangan
str2 = nagnar riba nanda
```

string palindrome

code:

```c
#include<stdio.h>
int main()
{
    char str1[30] = "adnan abir rangan";
```

```
    char str2[30];

    int i=0,len=0,j;
    while(str1[i]!='\0')
    {
        i++;
        len++;
    }
    for (j=0,i=len-1;i>=0;i--,j++)
    {
        str2[j] = str1[i];
    }
    str2[j] = '\0';

    printf("str1 = %s\n",str1);
    printf("str2 = %s\n",str2);

    int d = strcmp(str1,str2);

    if(d==0)
        printf("string is palindrome");
    else
        printf("string is not palindrome");

    return 0;
}
```

Result:

```
str1 = adnan abir rangan
str2 = nagnar riba nanda
string is not palindrome
```

string swapping

code:

```c
#include<stdio.h>
int main()
{
    char str1[15]="japan";
    char str2[15]="bangladesh";
    char temp[15];

    printf("\nbefore swapping:\n");
    printf("str1 = %s\n",str1);
    printf("str2 = %s\n",str2);

    strcpy(temp,str1);
    strcpy(str1,str2);
    strcpy(str2,temp);

    printf("\n\nafter swapping:\n");
    printf("str1 = %s\n",str1);
    printf("str2 = %s\n",str2);

    return 0;

}
```

Result:

```
before swapping:
str1 = japan
str2 = bangladesh


after swapping:
str1 = bangladesh
str2 = japan
```

strupr() and strlwr()

code:

```c
#include<stdio.h>
int main()
{
    char str[]="Adnan Abir Rangan";
    printf("str = %s\n",str);

    strupr(str);
    printf("strupr = %s\n",str);

    strlwr(str);
    printf("strlwr = %s\n",str);

    return 0;
}
```

Result:

```
str = Adnan Abir Rangan
strupr = ADNAN ABIR RANGAN
strlwr = adnan abir rangan
```

Number of vowels, consonants, words, digits and other

Code:

```c
#include<stdio.h>
int main()
{
    char str[100],ch;
    int i,vowel,consonant,digit,word,other;

    printf("enter a string : ");
    gets(str);

    i=vowel=consonant=digit=word=other=0;

    while((ch=str[i])!='\0')
    {
```

```
        if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch==
'I'||ch=='O'||ch=='U')
        vowel++;

        else if((ch>='a' && ch<='z') ||(ch>='A' && ch<='Z'))
        consonant++;

        else if(ch>='0' && ch<='9')
        digit++;

        else if(ch==' ')
        word++;

        else
        other++;

        i++;

    }
    word++;

    return 0;
}
```

Result:

```
#include<stdio.h>
int main()
{
    char str[100],ch;
    int i,vowel,consonant,digit,word,other;

    printf("enter a string : ");
    gets(str);

    i=vowel=consonant=digit=word=other=0;

    while((ch=str[i])!='\0')
    {
        if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch==
'I'||ch=='O'||ch=='U')
        vowel++;
```

```
        else if((ch>='a' && ch<='z') ||(ch>='A' && ch<='Z'))
        consonant++;

        else if(ch>='0' && ch<='9')
        digit++;

        else if(ch==' ')
        word++;

        else
        other++;

        i++;

    }
    word++;

    printf("Number of vowels = %d\n",vowel);
    printf("Number of consonant = %d\n",consonant);
    printf("Number of digits = %d\n",digit);
    printf("Number of words = %d\n",word);
    printf("Number of others = %d\n",other);

    return 0;
}
```

Result:

```
enter a string : Adnan Abir Rangan 7&(90
Number of vowels = 6
Number of consonant = 9
Number of digits = 3
Number of words = 4
Number of others = 2
```

Number of capital-small letters, digits

Code:

```c
#include<stdio.h>
int main()
{
    char str[100];
    int i, capital, small, digit;

    i=capital=small=digit=0;

    printf("enter a string : ");
    gets(str);

    while(str[i]!='\0')
    {
        if(str[i]>=65 && str[i]<=90)
        capital++;

        if(str[i]>=97 && str[i]<=122)
        small++;

        if(str[i]>=48 && str[i]<=57)
        digit++;

        i++;

    }
    printf("number of capital letters:%d\n",capital);
    printf("number of small letters:%d\n",small);
    printf("number of digits:%d\n",digit);

    return 0;
}
```

Result:

```
enter a string : Adnan Abir Rangan 68685
number of capital letters:3
number of small letters:12
number of digits:5
```

Function

Sum of 2/3 numbers and subtraction

Code:

```c
#include<stdio.h>

void sum(int a, int b, int c)
{
    printf("the sum is : %d\n",a+b+c);
}
void sub(int a, int b)
{
    printf("the subtraction is : %d\n",a-b);
}

int main()
{
    sum(10,39,28);
    sum(29,37,56);
    sub(38,75);
    sub(93,57);

    return 0;
}
```

Result:

```
the sum is : 77
the sum is : 122
the subtraction is : -37
the subtraction is : 36
```

Taking input from user and sum;

Code:

```c
#include<stdio.h>
int sum(int a, int b)
```

```
{
    return a+b;
}

int main()
{
    int num1, num2;

    printf("enter two numbers : ");
    scanf("%d%d",&num1,&num2);

    int result = sum(num1,num2);

    printf("the sum is : %d\n",result);

    return 0;
}
```

Result:

```
enter two numbers : 65 87
the sum is : 152
```

Square of a number

Code:

```
#include<stdio.h>

int square(int a)
{
    return a*a;
}


int main()
{
    int num,result;

    printf("enter a number : ");
    scanf("%d",&num);
```

```
    result = square(num);

    printf("square of number is : %d\n",result);

    return 0;
}
```

Result:

```
enter a number : 7
square of number is : 49
```

Area of a Triangle

Code:

```
#include<stdio.h>

double area(double b, double h)
{
    return 0.5*b*h;
}
int main()
{
    double base, height, result;

    printf("enter the base and height : ");
    scanf("%lf%lf",&base,&height);

    result = area(base,height);

    printf("the area of triangle is : %.4lf\n",result);

    return 0;
}
```

Result:

```
enter the base and height : 4 7
the area of triangle is : 14.0000
```

X To The Power Y using User-defined Function

Code:

```c
#include<stdio.h>

void calculatepower(double base,double exponential)
{
    double result=1,i;
    for(i=1;i<=exponential;i++)
    {
        result = result * base;
    }
    printf("the result is : %.4lf\n",result);
}

int main()
{
    double base, exponential;

    printf("enter base and exponential : ");
    scanf("%lf%lf",&base,&exponential);

    calculatepower(base,exponential);

    return 0;
}
```

Result:

```
enter base and exponential : 2 5
the result is : 32.0000
```

Getting result multiple times using function:

Code:

```c
#include<stdio.h>

void calculatepower(double base,double exponential)
{
    double result=1,i;
    for(i=1;i<=exponential;i++)
```

```
    {
        result = result * base;
    }
    printf("the result is : %.4lf\n",result);
}

int main()
{
    calculatepower(2,4);
    calculatepower(2,9);
    calculatepower(6,4);

    return 0;
}
```

Result:

```
the result is : 16.0000
the result is : 512.0000
the result is : 1296.0000
```

x to the power y using Library function

code:

```
#include<stdio.h>
int main()
{
    double base, exponent, result;

    printf("enter the base and exponenet : ");
    scanf("%lf%lf",&base,&exponent);

    result = pow(base,exponent);

    printf("%lf",result);

    return 0;
}
```

Result:

```
enter the base and exponenet : 3 4
81.000000
```

x to the power y without using Library function

code:

```c
#include<stdio.h>
int main()
{
    double base, exponent, result=1, i;

    printf("enter the base and exponenet : ");
    scanf("%lf%lf",&base,&exponent);

    for(i=1; i<=exponent; i++)
    {
        result = result * base;
    }

    printf("the result is : %lf",result);

    return 0;
}
```

Result:

```
enter the base and exponenet : 3 4
the result is : 81.000000
```

Passing Array to function

Code:

```c
#include<stdio.h>

void display(int x[])
{
    printf("the array is : ");

    int i;
    for(i=0; i<5; i++)
```

```
    {
        printf("%d ",x[i]);
    }
}

int main()
{
    int num[] = {10, 20, 30, 40, 50};
    display(num);

    return 0;
}
```

Result:

```
the array is : 10 20 30 40 50
```

finding maximum value from an array using function

Code:

```
#include<stdio.h>

void maximum(int num[])
{
    printf("the maximum value is : ");

    int i;
    int max = num[0];
    for(i=1; i<5; i++)
    {
        if(max<num[i])
        max = num[i];
    }
    printf("%d ",max);
}

int main()
{
    int num[] = {10, 20, 30, 40, 50};
```

```
    maximum(num);

    return 0;
}
```

Result:

```
the maximum value is : 50
```

Passing String to function

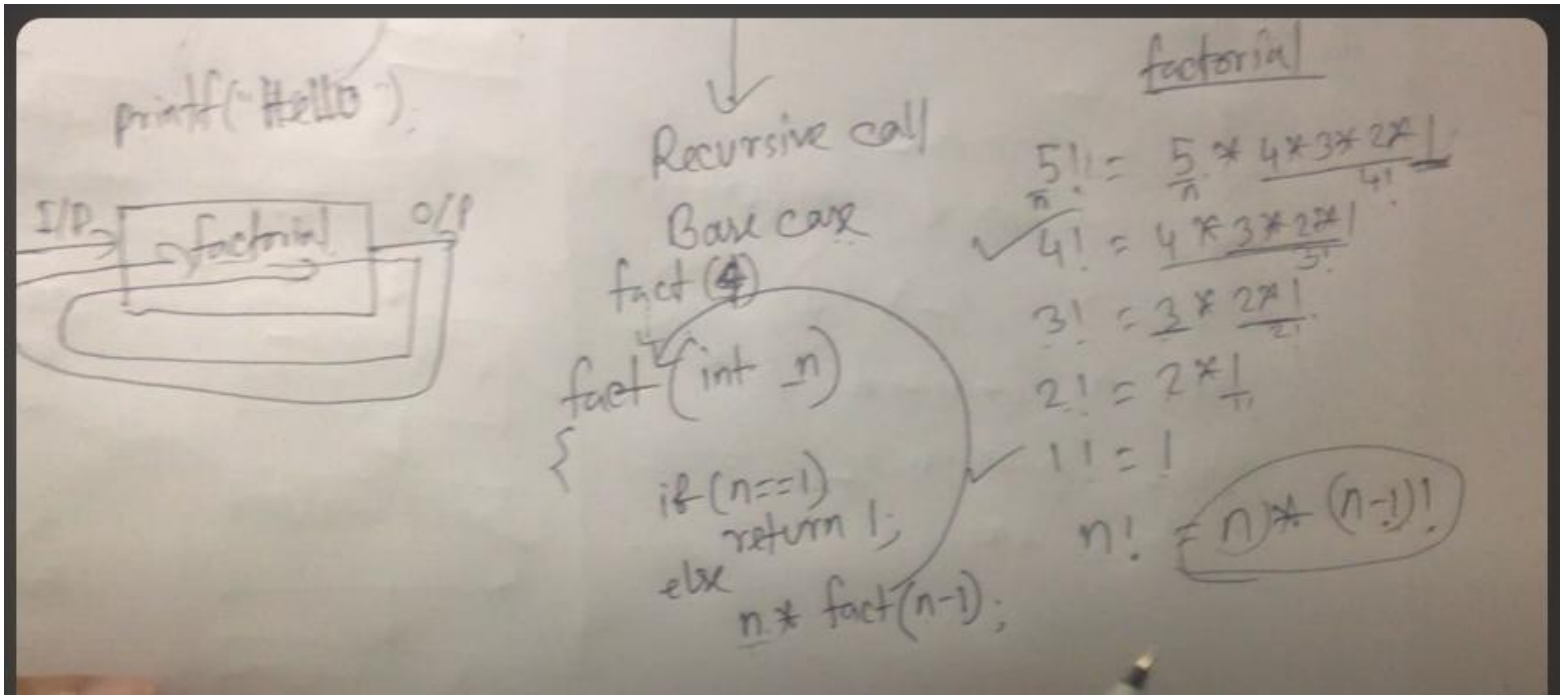
Code:

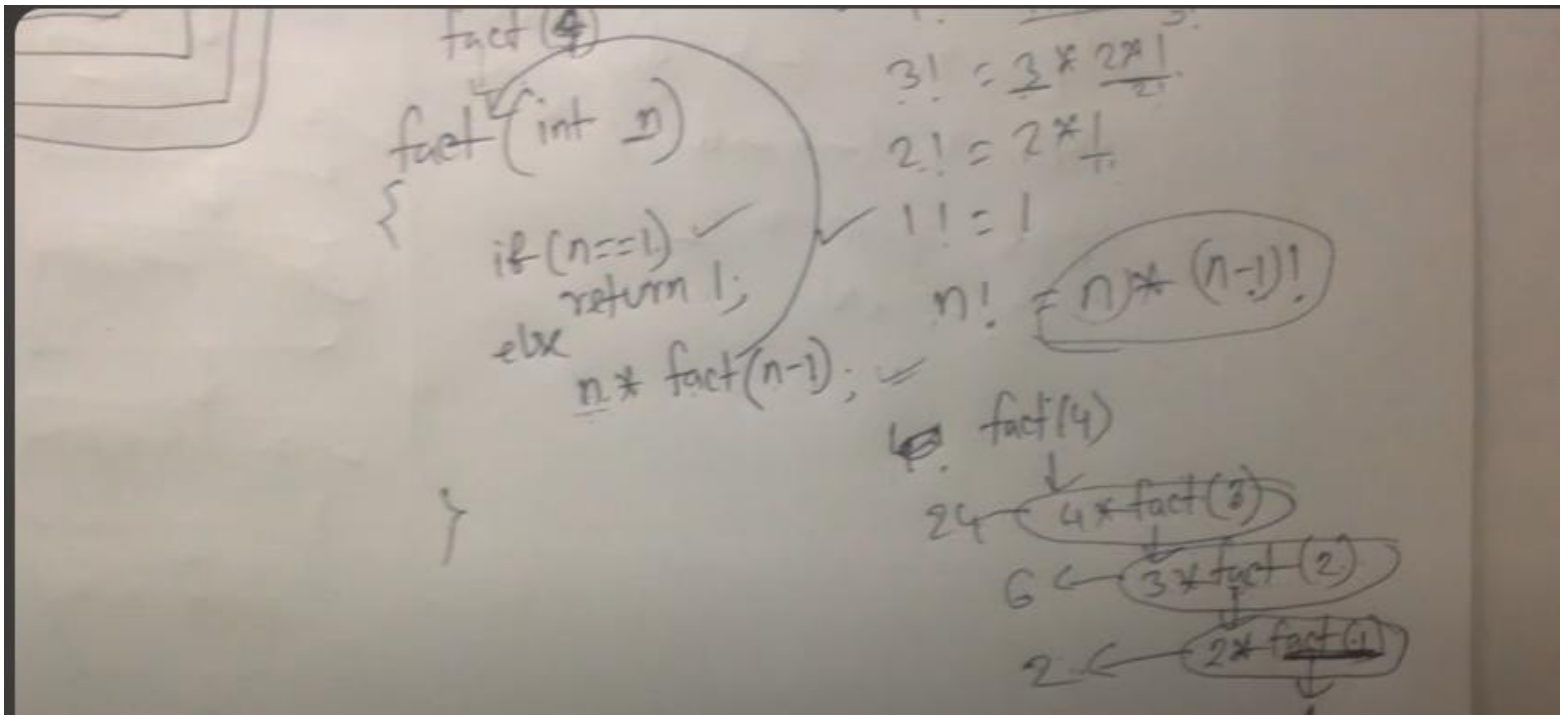
```
#include<stdio.h>

void display(char str[])
{
    int i=0;
    while(str[i] != '\0')
    {
        printf("%c\n",str[i]);
        i++;
    }
}

int main()
{
    char str[] = "Rangan";
    display(str);
}
```

Result:

```
R
a
n
g
a
n
```

Recursion


printf("Hello");
I/P
factrial
O/P
Recursive call
Base case
fact(4)
fact(int n)
{
if(n==1)
return 1;
else
n * fact(n-1);
factorial
5! = 5 * 4*3*2*1
4! = 4*3*2*1
3! = 3*2*1
2! = 2*1
1! = 1
n! = n * (n-1)!


fact(4)
fact(int n)
{
if(n==1)
return 1;
else
n * fact(n-1);
}
3! = 3*2*1
2! = 2*1
1! = 1
n! = n * (n-1)!
fact(4)
24
4 * fact(3)
6
3 * fact(2)
2
2 * fact(1)

Factorial Using Recursion

Code:

```c
#include<stdio.h>

int fact(int n)
{
    if(n==1)
        return 1;
    else
        return n*fact(n-1);
}

int main()
{
    int result = fact(5);

    printf("Factorial of 5 is = %d",result);

    return 0;
}
```

Result:

```
Factorial of 5 is = 120
```