1. Write a program to print numbers from 1 to 100.

IPO

**Input:**  
 The program takes an integer input a from the user.

**Process:**  
It uses a for loop to count from 1 up to the value of a, printing each number one by one.

**Output:**  
The program displays all numbers from 1 to a, each printed on a new line.

**CODE**

#include <stdio.h>

void main()

{

int a,i;

printf("ENTER A NUMBER");

scanf("%d",&a);

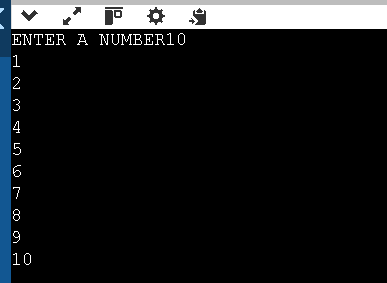
for(i=1;i<=a;i++)

{

printf("%d\n",i);

}

OUTPUT



1. Write a program to print even numbers from 1 to 50.

IPO

**Input:**  
The program accepts an integer a from the user.

**Process:**  
It runs a loop from 1 to a. During each iteration, it checks if the number is divisible by 2 (i % 2 == 0). If true, it prints that number — meaning it's an even number.

**Output:**  
 The program prints all even numbers from 1 to a, each on a new line.

**CODE  
 #include <stdio.h>**

**void main()**

**{**

**int a,i,even;**

**printf("ENTER A NUMBER");**

**scanf("%d",&a);**

**for(i=1;i<=a;i++)**

**{**

**if(i%2==0)**

**{**

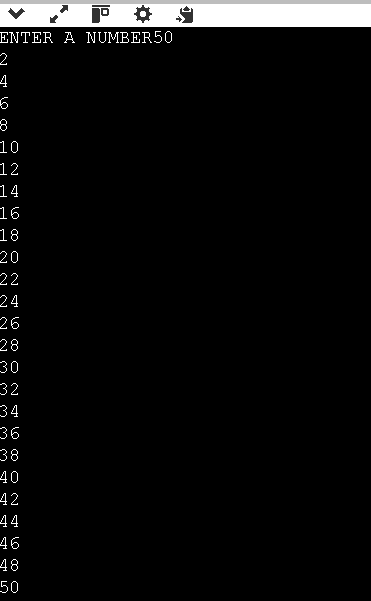
**printf("%d\n",i);**

**}**

**}**

**}**

**OUTPUT**

****

1. Write a program to find the factorial of a number.

IPO

**Input:**  
A fixed number n = 5 is used for which the factorial is to be calculated.

**Process:**  
The program multiplies all numbers from 1 to n using a loop and stores the result in fact.

**Output:**  
The program prints the factorial of the number n.

**CODE**

**#include <stdio.h>**

**void main()**

**{**

**int n=5,fact=1,i;**

**for(i=1;i<=n;i++)**

**{**

**fact=fact\*i;**

**}**

**printf("%d",fact);**

**}**

**OUTPUT**

****

1. Write a program to calculate the sum of digits of a number.

IPO

* **Input:**  
  The program accepts an integer a from the user, which is the upper limit of the range.
* **Process:**  
  It loops from 1 to a, checking each number:
  + If the number is odd, it adds to the odd sum and increases the odd count.
  + If the number is even, it adds to the even sum and increases the even count.
* **Output:**  
  The program prints:
  + Count of odd numbers
  + Count of even numbers
  + Sum of all odd numbers
  + Sum of all even numbers

**CODE**

**#include <stdio.h>**

**void main()**

**{**

**int sum,i,a,oddcount=0,evencount=0,oddnum=0,evennum=0;**

**scanf("%d",&a);**

**for(i=1;i<=a;i++)**

**{**

**if(i%2!=0)**

**{**

**oddnum=oddnum+i;**

**oddcount++;**

**}**

**else**

**{**

**evennum=evennum+i;**

**evencount++;**

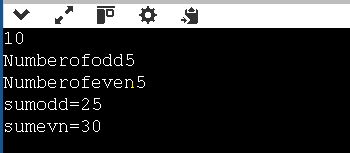
**}**

**}**

**printf("Numberofodd=%d\nNumberofeven=%d\nsumodd=%d\nsumevn=%d\n",oddcount,evencount,oddnum,evennum);**

**}**

**OUTPUT**

****

1. Write a program to reverse a number.

IPO

**Input:**  
The program takes an integer number from the user.

**Process:**  
It extracts each digit of the number from the end using modulo (% 10) and builds a new number in reverse order by multiplying the current reversed number by 10 and adding the digit. This continues until the number becomes 0.

**Output:**  
The program prints the reversed number.

CODE

#include <stdio.h>

void main()

{

int a,reverse=0,digit,b,c=100;

scanf("%d",&a);

b=a;

while(b>0)

{

digit=b%10;

reverse=reverse\*10+digit;

b=b/10;

}

printf("%d",reverse);

}

OUTPUT



1. Write a program to check whether a number is a palindrome.

IPO

**Input:**  
The program takes an integer input from the user.

**Process:**  
It reverses the number by extracting each digit (using % 10), building a reversed number (using reverse = reverse \* 10 + digit). Then it checks whether the reversed number is equal to the original.

CODE

#include <stdio.h>

void main()

{

int a,reverse=0,digit,b,c=100;

scanf("%d",&a);

b=a;

while(b>0)

{

digit=b%10;

reverse=reverse+digit\*c;

b=b/10;

c=c/10;

}

if(reverse==a)

{

printf("its palindrome");

}

else

{

printf("its not a palindrome");

}

}

OUTPUT



1. Write a program to print multiplication table of a number.

IPO

**Input:**  
The program accepts an integer n from the user.

**Process:**  
It uses a for loop to multiply n with numbers from 1 to 10 and prints each multiplication step.

**Output:**  
The program displays the multiplication table of the entered number up to 10.

CODE

#include <stdio.h>

void main()

{

int n,i;

scanf("%d",&n);

for(i=1;i<=10;i++)

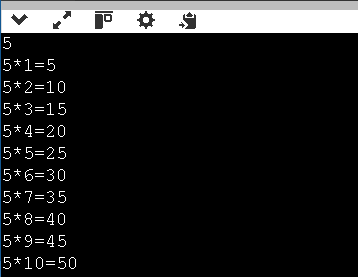
{

printf("%d\*%d=%d\n",n,i,n\*i);

}

}

OUTPUT



1. Write a program to count the number of digits in a number.

IPO

**Input:**  
The program reads an integer number from the user.

**Process:**  
It divides the number by 10 in a loop and counts how many times this can be done before the number becomes 0 — this gives the number of digits.

**Output:**  
The program prints how many digits the number has.

CODE

#include <stdio.h>

void main()

{

int a,count=0,i;

scanf("%d",&a);

if(a==0)

{

printf("its not a number");

}

while(a!=0)

{

a=a/10;

count++;

}

printf("%d",count);

}

OUTPUT



1. Write a program to print the Fibonacci series up to n terms.

IPO

**Input:**  
No user input is required; the program generates the sequence directly.

**Process:**  
The program initializes the first two Fibonacci numbers as 0 and 1. Then, using a loop, it adds the last two numbers to generate the next number in the sequence and repeats this for 5 more terms.

**Output:**  
The program prints the first 7 Fibonacci numbers separated by spaces.

**CODE**

#include <stdio.h>

void main()

{

int a=0,b=1,c,i,k;

printf("%d%d",a,b);

for(c=1;c<=5;c++)

{

k=a+b;

printf("%d",k);

a=b;

b=k;

}

}

1. Write a program to calculate the sum of the first n natural numbers.

IPO

**Input:**  
The program takes an integer a as input.

**Process:**  
It adds all numbers from 1 to a using a loop (sum = sum + i).

**Output:**  
The program prints the total sum from 1 to a.

**CODE**

**#include <stdio.h>**

**void main()**

**{**

**int a,b,sum=0,i;**

**scanf("%d",&a);**

**for(i=1;i<=a;i++)**

**{**

**sum=sum+i;**

**}**

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

**}**

**OUTPUT**

****