Java programs –1

Fibonacci –0 1 1 2 3 5 8 13 21 34 55....

public class Fibonacci {

public static void main(String[] args) {

//0 1 1 2 3 5 8 13 21 34 55

int f1=0, f2=1,f3;

System.out.print(f1+" ");

System.out.print(f2+" ");

for(int i=0;i<=8;i++)

{

f3=f1+f2;

System.out.print(f3+" ");

f1=f2;

f2=f3;

}

}

}

Output: 0 1 1 2 3 5 8 13 21 34 55

Prime or not –1 2 3 5 7 11 13 17 19 23 31 33

public class prime {

public static void main(String[] args) {

// TODO Auto-generated method stub

// 3 5 7 11 13 15 17 19 23 31

boolean status=true;

System.out.print("the 1st 10 prime numbers are");

for(int a=2; a<=35;a++)

{

for(int i=2;i<a;i++)

{

if(a%i==0)

{

status=false;

break;

}}

if(status==true)

{

System.out.print(" "+a+" ");

}

status=true;

}

}

}

Output: the 1st 10 prime numbers are 2 3 5 7 11 13 17 19 23 29 31

Find the factorial …..

public class factorial {

public static void main(String[] args) {

// TODO Auto-generated method stub

// 5!=5\*4\*3\*2\*1

int a=1,p=1;

for(a=5; a<=5 && a!=0; a--)

{

p=p\*a;

System.out.println(p);

}

}

}

Output: 5

20

60

120

120

Sum of digits........

public class sum\_of\_digits {

public static void main(String[] args) {

// TODO Auto-generated method stub

//164......1+6+4.....11

int num=164,rev=0,rem=0;

while(num>0)

{

rem=num%10;

num=num/10;

rev=rev+rem;

}

System.out.println("sum of numbers "+rev);

}

}

Output: sum of numbers 11

Pattern 1:

public class patttern\_1 {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\*

1

1 1

1 1 1

1 1 1 1

1 1 1 1 1

\*/

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

{

for(int j=1;j<=i;j++)

{

if(i>=j)

{

System.out.print("1 ");

}

}

System.out.println(" ");

}

}

}

Output:

1

1 1

1 1 1

1 1 1 1

1 1 1 1 1

Pattern 2:

public class pattern\_2 {

public static void main(String[] args) {

/\*

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

\*/

for(int i=1;i<=5;i++)

{

for(int j=1;j<=5;j++)

{

if(j<=i)

{

System.out.print(j+" ");

}

}

System.out.println(" ");

}

}

}

Output:

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5