* Read it on web layout

Recursive function :

Function is calling it self

It will replace loop

P s v call (int n) //0 1 2 3 4 5

{

If(n==5)

return;

n++;//1 2 3 4

call(n);//1 2 3 4 5 call(n++)

o.p(n) 5 4 3 2 1

}

It requires a condition to come out of recursive function.

It is concept of stack. Last value you will get first

Advantage it reduce line of code and execute faster

Disadvantage it occupy more memory space.

//Print 1 to 5 using recursive function

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| import static java.lang.System.out;  public class Main  { public static void main(String[] args) {  int no=6;  call(no);  }  } | public static  void call(int n) //6  **{**  if(n==1)  return;  n--;  **call(n); //5**  print(n);//5  **}** | public static  void call(int n)//5  {  if(n==1)  return;  n--;  **call(n);//4**  print(n);//4  } | public static  void call(int n)//4  {  if(n==1)  return;  n--;  **call(n);**//3  print(n);//3  } | public static  void call(int n)//3  {  if(n==1)  return;  n--;  **call(n);//2**  print(n);//2  } | public static  void call(int n)//2  {  if(n==1)  return;  n--;  **call(n);//1**  print(n);//1  } | public static  void call(int n)//1  {  if(n==1)  return;  n--;  call(n);//1  printf("%d",n);  }   |  | | --- | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | |

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online Java Compiler.

Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*print 1 to 5\*/

import static java.lang.System.out;

public class Main

{ public static void main(String[] args) {

int no=6;

call(no);

}

public static void call(int n)

{

if(n==1)

return;

n--;

call(n);//5

out.println(n);//5

}

}

Print binary number

|  |  |  |  |
| --- | --- | --- | --- |
| Public static //3  void binary(int num)//4  { int bin;  if(num<=0)  return;  bin = num % 2;//4%2=0  binary(num/2);4/2=2  print(bin);//0  } | Public static  void binary (int num)//2  { int bin;  if(num<=0)  return;  bin = num % 2;//2%2=0  binary(num/2);2/2=1  print(bin);//0  } | void binary(int num)//1  { int bin;  if(num<=0)  return;  bin = num % 2;//1%2=1  binary(num/2);//1/2=0  print(bin);1  } | void binary(int num)//0  { int bin;  if(num<=0) //num==0  return;  bin = num % 2;  binary(num/2);  print(bin);  } |

4

Int no=4;

Int r,i;=0

Int []arr;

While(no!=0)

{

r =no%2; 🡺0 0 1

arr[i++]=r;

no=no/2;

}

For(int j=i-1;j>=0;j--)

Out.print(arr[j]);//1 0 0

0=4%2 0=2%2 1=1%2

2=4/2 1=2/2 0=1/2

1

0

1

0

2

2

4

4

0

2

2

1

2

2

0

import static java.lang.System.out;import java.util.\*;public class Main{ public static void main(String[] args) { Scanner sc=new Scanner(System.in); out.println("enter no "); int no=sc.nextInt(); binary(no); } public static void binary(int num)//4

{ int bin;

if(num<=0)

return;

bin = num % 2;

binary(num/2);

out.print(bin);

}

}

Factorial

#include<stdio.h>

import static java.lang.System.out;import java.util.\*;public class Main{ public static void main(String[] args) { Scanner sc=new Scanner(System.in); out.println("enter no "); int no=sc.nextInt(); int k=fact(no); out.println(k); } public static int fact(int x)//4

{ int f;

if(x==1||x==0)

return 1;

else

{ f=x\*fact(x-1);

//24=4\*4-1(3) 6

return f;//24

}

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| Public static  int fact(int x)//4  { int f;  if(x==1||x==0)  return 1;  else  { f=**x**\*fact(x-1);  24=4\*4-1(3) 6  return f;//24  } | int fact(int x)//3  { int f;  if(x==1||x==0)  return 1;  else  { f=x\*fact(x-1);  2=3\*3-1(2)//2  return f;//6  } | int fact(int x)//2  { int f;  if(x==1||x==0)  return 1;  else  { f=x\*fact(x-1);  2=2\*2-1(1) 1  return f;//2  } | int fact(int x)//1  { int f;  if(x==1||x==0)  return 1;  else  { f=x\*fact(x-1);  return f;  } |

0 3 5 6

// Fibonacci series using recursion

class HelloWorld {

static int x=0,y=1;

public static void main(String[] args) {

System.out.println(x+" "+y);

fib(5);

System.out.println("Hello, World!");

}

static void fib(int n)// 5

{

int c;

c=x+y;

System.out.println(c);

x=y;

y=c;

n--;//

if(n>2)

fib(n);

}

}

Task🡺 do the sum of digit using recursive function

//one more answer

Public static

int sum(int m)//0

{

if(m==0)

return 0;

return sum(m/10)+m%10;

12/10

}

Public static

int sum(int m)//1

{

if(m==0)

return 0;

return sum(m/10)+m%10;

1/10

} 0+1%10=1

Public static

int sum(int m)//12

{

if(m==0)

return 0;

return sum(m/10)+m%10;

12/10

} 1+12%10=2

3

Public static int sum(int m)//123

{

if(m==0)

return 0;

return sum(m/10)+m%10;

123/10

3+123%10=3

} 6

Public static void main()

{

int m,res;

printf("Enter the value of m");

scanf("%d",&m);//123

res = sum(m);

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

getch();

}

what value will be return by Num function

public int Num(int no)//11

{

if(no/2==0)

return 1;

else 11/2

return Num(no/2)\*10+no%2;

}

-------------------------------------------------------------------------------

no=11

int Num(int no)//1

{

if(no/2==0)

return 1;

else

return Num(no/2)\*10+no%2;

}

int Num(int no)//2

{

if(no/2==0)

return 1;

else 2/2

return Num(no/2)\*10+no%2;

1 \*10+ 2%2;

` 10+0

} 10

int Num(int no)//5

{

if(no/2==0)

return 1;

else 5/2

return Num(no/2)\*10+no%2;

10\*10+5%2

100+1

101

}

what value will be return by Num function

int Num(int no)//11

{

if(no/2==0)

return 1;

else 11/2

return Num(no/2)\*10+no%2;

} 101\*10+11%2

1010+1

**1011**

101 1011 1101 100

import java.util.Scanner;public class Day8\_1 { public static void main(String[] args) { Scanner sc=new Scanner(System.in); System.out.println("Enter the number"); int num=sc.nextInt(); int res=cal(num); System.out.println("result="+res);

}

private static int cal(int num) {

if(num/2==0)

return 1;

else

return cal(num/2)\*10+num%2; //1011

}

}