

Java - BigInteger Class Example Program

Is there any variable in any programming language to store 100! ?

My answer NO , So how can we achieve it ?

There are several tricks to do it In 'C' it can be done using simple logic.

Here is a sample program to find out factorial number without overflow.

The logic behind the program is count the factorial number then split it into smaller pieces and store it into an array at last print the array from reverse order.

// program - 01

```
/// Author : Md. shohanur Rahaman

#include<stdio.h>
#include<string.h>
#define N 250
#define B 10
int main()
{
    int fact[N],i,j,n,size=0,carry=0,tmp=0,a;

    scanf("%d",&n);
    memset(fact,0,N);
    fact[0]=1;
    size=1;
    carry=0;

    for(j=1;j<=n;j++){
        for(i=0;i<size;i++){
            tmp=fact[i]*j+carry;
            fact[i]=tmp%B;
            carry=tmp/B;
        }
        while(carry>0){
            fact[size++]=carry%B;
```

```

        carry=carry/B;
    }
}
for(a=size-1;a>=0;a--){
    printf("%d",fact[a]);
}
printf("\n");

return 0;
}

```

But sometimes its very disturbing to write such kind of code Don't worry there is a simple solution in java, use" BigInteger" class and calculate manipulate any number without Overflow.

Here is a simple program of using "BigInteger" Class

// program - 02

```

import java.math.BigInteger;
/**
 *
 * @author Shohan
 */
public class BigIntExample {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        BigInteger objb1=new BigInteger("920011111111111111111111");
        BigInteger objb2 =new BigInteger("444444444444444410444444");

        BigInteger mul;           // BigInteger Type variable 'mul'
        mul = objb1.multiply(objb2); // This statement means (objb1*objb2)

        BigInteger div;
    }
}

```

```

        div=objb1.divide(objb2);          // This statement means (objb1/objb2)

        BigInteger add=objb1.add(objb2);

        System.out.println("product : "+ mul);
        System.out.println("Division : "+ div);
        System.out.println("Sum : "+ add);

    }
}

```

But there is some problem of BigInteger class , It is slower, take memory space higher than other operations. Another important aspect of BigInteger class is that it is immutable. This means that you cannot change the stored value of a BigInteger object, instead you need to assign the changed value to a new variable.

Here is a simple to calculate Factorial number.

// Program – 03

```

import java.math.BigInteger;
/**
 *
 * @author Shohan
 */
public class Factorial {
    public static void main(String args[]){

        int n=99;
        String fact= calculateFactorial(n);
        System.out.println(n+"! = "+fact);

    }
    // created a function to calculate factorial

    public static String calculateFactorial(int n){          // 'BigInteger'
        BigInteger result=new BigInteger("1");

        for(int i=1;i<=n;i++){

```

```
        result=result.multiply(BigInteger.valueOf(i));  
    }  
    return result.toString();  
}  
  
}
```

- Author : Md. Shohanur Rahaman