

TestJNI.java

```
import java.util.Scanner;

import static java.lang.System.out;

public class TestJNI {

    static {

        System.loadLibrary("cal");

    }

    private native int add(int n1, int n2);
    private native int sub(int n1, int n2);
    private native int mul(int n1, int n2);
    private native int div(int n1, int n2);

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        out.println("Simple Caculator");

        int a;

        do{

            out.println("Enter Your choice");

            out.println("1.ADD \n 2.SUB \n 3.MUL 4.DIV");

            a=sc.nextInt();

            if(a==1)

            { int b;

                int c;

                out.println("Enter first Number");

                b=sc.nextInt();

                out.println("Enter second Number");

                c=sc.nextInt();

                System.out.println("Addition is="+new TestJNI().add(b,c));

            }

        }
```

```

else if(a==2){
int b; int c;
out.println("Enter first Number");
b=sc.nextInt();
out.println("Enter Second Number");
c=sc.nextInt();
System.out.println("Subtraction is="+new TestJNI().sub(b,c)); }
else if(a==3) {
int b; int c;
out.println("Enter first Number");
b=sc.nextInt();
out.println("Enter Second Number");
c=sc.nextInt();
System.out.println("Multiplication is="+new TestJNI().mul(b,c)); }
else if(a==4) {
int b; int c;
out.println("Enter first Number");
b=sc.nextInt();
out.println("Enter Second Number");
c=sc.nextInt();
System.out.println("Division is="+new TestJNI().div(b,c)); }
while(a!=0); }}

```

TestJNI.c

```

#include <jni.h>
#include <stdio.h>
#include "TestJNI.h"

// Implementation of native method add() of TestJNI class

```

```

JNIEXPORT jint JNICALL Java_TestJNI_add(JNIEnv *env , jobject
thisObj , jint n1 , jint n2) {
jint res;
res=n1+n2;
return res; }

JNIEXPORT jint JNICALL Java_TestJNI_sub(JNIEnv *env , jobject
thisObj , jint n1 , jint n2) {
jint res;
res=n2-n1;
return res; }

JNIEXPORT jint JNICALL Java_TestJNI_mul(JNIEnv *env , jobject
thisObj , jint n1 , jint n2) {
jint res;
res=n1*n2;
return res; }

JNIEXPORT jint JNICALL Java_TestJNI_div(JNIEnv *env , jobject
thisObj , jint n1 , jint n2) {
jint res;
res=n1/n2;
return res; }

```

TestJNI.h

```

/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>

/* Header for class TestJNI */

#ifndef _Included_TestJNI
#define _Included_TestJNI
#ifdef __cplusplus
extern "C" {
#endif

```

```

/*
 * Class:   TestJNI
 * Method:  add
 * Signature: (I)I
 */
JNIEXPORT jint JNICALL Java_TestJNI_add
    (JNIEnv *, jobject, jint, jint);

/*
 * Class:   TestJNI
 * Method:  sub
 * Signature: (I)I
 */
JNIEXPORT jint JNICALL Java_TestJNI_sub
    (JNIEnv *, jobject, jint, jint);

/*
 * Class:   TestJNI
 * Method:  mul
 * Signature: (I)I
 */
JNIEXPORT jint JNICALL Java_TestJNI_mul
    (JNIEnv *, jobject, jint, jint);

/*
 * Class:   TestJNI
 * Method:  div
 * Signature: (I)I
 */
JNIEXPORT jint JNICALL Java_TestJNI_div
    (JNIEnv *, jobject, jint, jint);
#ifdef __cplusplus }
#endif
#endif

```

output:

```
sl4lab@sl4lab-HP-ProDesk-400-G1-SFF:~/Desktop$ javac TestJNI.java
```

```
sl4lab@sl4lab-HP-ProDesk-400-G1-SFF:~/Desktop$ javah -jni TestJNI
```

```
sl4lab@sl4lab-HP-ProDesk-400-G1-SFF:~/Desktop$ gcc -fPIC -I/usr/lib/jvm/java-8-openjdk-amd64/include -I//usr/lib/jvm/java-8-openjdk-amd64/include/linux -o libcal.so -shared TestJNI.c
```

```
sl4lab@sl4lab-HP-ProDesk-400-G1-SFF:~/Desktop$ java -Djava.library.path=. TestJNI
```

Simple Caculator

Enter Your choice

1.ADD

2.SUB

3.MUL

4.DIV

4

Enter first Number

12

Enter Second Number

23

Division is=0

Enter Your choice

1.ADD

2.SUB

3.MUL

4.DIV

1

Enter first Number

12

Enter second Number

32

Addition is=44

Enter Your choice

1.ADD

2.SUB

3.MUL

4.DIV

2

Enter first Number

43

Enter Second Number

3

Subtraction is=-40

Enter Your choice

1.ADD

2.SUB

3.MUL

4.DIV

3

Enter first Number

4

Enter Second Number

3

Multiplication is=12

Enter Your choice

1.ADD

2.SUB

3.MUL

4.DIV