**public** **class** RecursiveMethodsAT {

/\*\*

\* 1. Recursively find and return

\* the product of the first y whole numbers.

\*

\* **@param** y

\* **@return** y! as a long integer

\*

\* @

\*/

**public** **static** **long** factorial(**int** y) {

}

/\*\*

\* 2. Recursively find and return

\* the sum of the first y whole numbers.

\*

\* **@param** y

\* **@return** sum as an int

\*/

**public** **static** **int** sumInt(**int** y)

{

}

/\*\*

\* 3. Recursively finds the value of 2 to the yth power

\* **@param** y

\* **@return** 2^y as a long integer

\*/

**public** **static** **long** powerOfTwo(**int** y)

{

}

/\*\*

\* 4. Recursively find the nth term of the Fibonacci Sequence

\* 1,1,2,3,5,8,13.......

\*

\* **@param** n

\* **@return** nth term as an int

\*/

**public** **static** **int** fib(**int** n)

{

}

/\*\*

\* 5. Recursively finds and returns the sum of the digits of n

\* **@param** n

\* **@return**

\*/

**public** **static** **int** sumDigits(**int** n) {

}

/\*\*

\* 6. Recursively find and return the reverse of a String s

\* **@param** s

\* **@return** the reverse of s

\*/

**public** **static** String reverse(String s) {

}

/\*\*

\* 7. Recursively the sum of the first y terms of the binary series.

\* 1/2 + 1/4 + 1/8+ ...

\* **@param** y the number of terms of the sequence

\* **@return** the sum of y terms as a double

\*/

**public** **static** **double** binarySeries(**int** y){

}

/\*\*

\* 8. Recursively finds the maximum element of an array

\* **@param** arr

\* **@return** the maximum element in the array

\*/

**public** **static** **int** maxValue(**int**[] arr) {

}

/\*\*

\* 9. Recursively finds the sum element of an int array

\* **@param** arr

\* **@return** the sum of the elements in the array

\*/

**public** **static** **int** findSum(**int**[] a) {

}

/\*\*

\* 10. Recursively finds the index number of lookFor in an array

\* **@param** arr

\* **@return** the index number of lookFor. -1 if not found

\*/

**public** **static** **int** search(**int**[] arr, **int** lookFor) {

}

/\*\*

\* 11. Recursively finds and returns the sum of a 2DIM array

\* **@param** array

\* **@return** sum as an int

\*/

**public** **static** **int** sumOfArray(**int**[][] array)

{

}

/\*\*

\* 12. recursively fills a 2Dim array with the chararacter c

\* **@param** array

\* **@param** c

\*/

**public** **static** **void** fillArray2(**char**[][] array, **char** c) {

}

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

System.***out***.println(" 20! "+ *factorial*(20));

System.***out***.println("Sum of first 16 Positive integers: " + *sumInt*(16));

System.***out***.println("2^12: "+ *powerOfTwo*(12));

System.***out***.println("12th Fibonacci number: " + *fib*(12));

System.***out***.println("Digit Sum of 12345: " + *sumDigits*(12345));

System.***out***.println("RACECAR reversed: " + *reverse*("RACECAR"));

System.***out***.println("Sum of 5 terms of binary Series: "+ *binarySeries*(4));

**int**[] x = { 1, 22, 133, 34, 56, 62, 79 };

//System.out.println();

System.***out***.println("MaxValue: " +*findMax*(x));

System.***out***.println("Sum: " + *findSum*(x));

System.***out***.println("Search for 111: " + *search2*(x,111));

System.***out***.println("Search for 79: " + *search2*(x,79));

**int**[][] y = { { 1, 2 }, { 3, 4 }, { 5, 6 } };

System.***out***.println("Sum of 2Dim array: " + *sumOfArray*(y));

**char**[][] b = **new** **char**[5][19];

*fillArray2*(b, '@');

**for** (**char**[] r : b) {

**for** (**char** c : r)

System.***out***.print(c);

System.***out***.println();

}

/\*

Output

20! 2432902008176640000

Sum of first 16 Positive integers: 136

2^12: 4096

12th Fibonacci number: 144

Digit Sum of 12345: 15

RACECAR reversed: RACECAR

Sum of 5 terms of binary Series: 0.9375

MaxValue: 133

Sum: 387

Search for 111: -1

Search for 79: 6

Sum of 2Dim array: 21

@@@@@@@@@@@@@@@@@

@@@@@@@@@@@@@@@@@

@@@@@@@@@@@@@@@@@

@@@@@@@@@@@@@@@@@

@@@@@@@@@@@@@@@@@

\*/

\*/

}