<https://t.me/+1lANXAG7IplkMTE1>

**public** **class** **HelloWorld**{

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

System.out.println("" + **7** + **6**);

} }

### QPattern Printing

**Description**

Given an integer n, print the triangle. You need to read the integer as input and output the triangle as shown below.

**Input**: 6

**Output**:

     \*

    \* \*

   \* \* \*

  \* \* \* \*

 \* \* \* \* \*

\* \* \* \* \* \*

int n = scanner.nextInt();

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

for (int j = n; j >= i; j--) {

System.out.print(" ");

}

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

System.out.print("\* ");

}

System.out.println();

}

### Q2 -- Power of 2

**Description**

Write a code that prints the highest power of 2, less than or equal to a given number. For example, if the input number is 9, the code should print 8, as 8 i.e 2 to power 3 is the highest power of two, which is less than 9.

Hint: This can be achieved if the loop starts with number 1 and keeps multiplying it with 2 until it reaches the power of 2, which is less than or equal to the given number.

For example:

Given number = 9.

We start with 1.

As 1<9, we multiply it with 2.

1\*2 = 2.

As 2<9, we multiply it with 2 again.

2\*2 = 4.

As 4<9, we multiply it with 2 again.

4\*2 = 8.

As 8<9, we multiply it with 2 again.

8\*2 = 16

As 16>9, we stop the multiplication and get the previous number, i.e., 8.

Solution--- Graphical user interface, text, application

Description automatically generated

### QA3---Highest Common Factor

**Description**

Write a Java code to calculate the highest common factor (HCF) or the greatest common divisor of two numbers. Just complete the logic of the code.

Hint: HCF is the highest number that divides two numbers. So, your loop should start by checking if the lower number of the two numbers entered is the HCF and then check the numbers below, one by one, until the HCF is found.

Graphical user interface

Description automatically generated

**QUESTION: 5**

Comprehension: Income tax

Create a program for calculating the income tax to be paid by an individual earning less than 1 crore. Use conditional statements only.

Use the following rules:

|  |  |  |  |
| --- | --- | --- | --- |
| Income Tax Age slab | | | |
| **Income range** | **General (non-seniors)** | **Senior citizens (>= 60 & < 80)** | **Very senior citizens (>= 80)** |
| Up to 2,50,000 | Nil | Nil | Nil |
| Rs. 2,50,001 to Rs. 3,00,000 | 10% | Nil | Nil |
| Rs. 3,00,001 to Rs. 5,00,000 | 10% | 10% | Nil |
| Rs. 5,00,001 to Rs. 10,00,000 | 20% | 20% | 20% |
| Above Rs. 10,00,000 | 30% | 30% | 30% |

|  |
| --- |
|  |

Additional information:

* The basic exemption limit for individuals (i.e. below 60 years of age) is Rs. 2.50 lakhs.
* The basic exemption limit for senior citizens (60 years to 80 years) is Rs. 3.00 lakhs.
* The basic exemption limit for very senior citizens (80 years and above) is Rs. 5.00 lakhs.
* No extra cess is to be levied.

Take the income and age as inputs and return the income tax.

For example, if the income of an individual is 6 lacs and his/her age is < 60, then the income tax to be paid is calculated by the following set of rules:

|  |  |  |  |
| --- | --- | --- | --- |
| **Income range** | **Tax rate** | **Taxable                               income** | **Tax to be paid** |
| Income up to Rs. 2,50,000 | Nil | - | Nil |
| Income from Rs. 2,50,000 – Rs. 5,00,000 | 10% | 0.10 \* (500000- 250000) | 25,000 |
| Income from Rs. 5,00,000 – Rs. 10,00,000 | 20% | 0.2 \* (600000 - 500000) | 20,000 |
| Income over Rs. 10,00,000 | - | - | 0 |
| Tax |  |  | 45,000 |

Income Tax Calculator

**Description**

Note: Please enter the income figures without commas. So if the annual income is Rs. 2,50,000, enter only 250000 in the Input Console. Enter the age only after you've entered the income. Also, make sure that you enter your inputs in the Input Console before clicking on Test Run.

Sol----

### Minimum in Array

**Description**

You are given an array. You are required to complete the function to print the minimum element in the array. No input is required from your end.

Sol---

Text

Description automatically generated

public static void FindMinimum(int arr[])

{

int n = arr.length;

int minvalue = arr[0];

for(int i=0; i<n; i++){

if(arr[i] < minvalue){

minvalue = arr[i];

}

}

System.out.print(minvalue);

//complete the function here to output the minimum value

}

### QA--Reverse the Array

**Description**

Given an array. You need to complete the function to print the element in reverse order. No input is required from your end. You are provided with an array. You need to output the reversed array.

**Input**: [1, 5, 7, 3, 4, 8, 0]

**Output**: 0 8 4 3 7 5 1

**Note**: Output Each Number with a space in the same line.

Text

Description automatically generated

public static void PrintReverseArray(int arr[])

{

//complete the function here to print reverse array.

int n = arr.length;

int[] b = new int[n];

int j = n;

for (int i = 0; i < n; i++) {

b[j - 1] = arr[i];

j = j - 1;

}

// printing the reversed array

for (int k = 0; k < n; k++) {

System.out.print(b[k]+" ");

}

}

### QA-- Element in the array

**Description**

You are given an array and a number. You are required to complete the function to print whether the element is present in the array or not. No input is required from your end. You must output either a yes or no.

Input: [1, 5, 7, 3, 4, 8, 0], **4**

Output: yes

Text

Description automatically generated

public static void Find(int arr[], int number)

{

int i;

int n = arr.length;

String test = "yes";

for (i= 0 ; i<n;i++) {

if (arr[i] == number) {

test = "yes";

break;

}else{

test ="no";

}

}

System.out.println(test);

### QA -- 2-D Array

**Description**

Write a Java code that takes two numbers x and y as the input and generates a two-dimensional (2D) array, where the value in the i-th row and j-th column of the array is (i+j)/2.

**Note:**The indexing in the array follows:i=0,1,...x-1 and j=0,1....,y-1

The input will have two integers: x and y.

The output should be a 2D array.

Scanner in = new Scanner(System.in);

int n = in.nextInt();

int m = in.nextInt();

double [][]arr = new double[n][m];

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

System.out.print((arr[i][j]/2) + " ");

}

System.out.println();

}

### QA-- LCM

**Description**

Write a function that takes two numbers as parameters and returns the LCM of these two numbers. So, if the values passed into the function are 12 and 20, the function will return the lowest common multiple of these two, i.e 60.

import java.util.Scanner;

class Source {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

// Enter the two numbers in the input console

int number1 = scan.nextInt();

int number2 = scan.nextInt();

System.out.print(lcm(number1, number2));

}

// Write the LCM function here

public static Integer lcm(int n1, int n2){

int lcm, gcd=1;

for(int i = 1; i <= n1 && i <= n2; ++i) {

// Checks if i is factor of both integers

if(n1 % i == 0 && n2 % i == 0)

gcd = i;

}

lcm = (n1 \* n2) / gcd;

return lcm;

}

}

### QA -- Final Pattern

**Description**

You are given a number n. Print the pattern shown below.

Input: 4

Output:

\* \* \* \*

 \* \* \*

  \* \*

   \*

   \*

  \* \*

 \* \* \*

\* \* \* \*

for (int i= 0; i<= n-1 ; i++)

{

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

{

System.out.print(" ");

}

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

System.out.print("\*" + " ");

}

System.out.println("");

}

for (int l= n-1; l>= 0; l--)

{

for (int j=0; j<l ;j++)

{

System.out.print(" ");

}

for (int k=l; k<=n-1; k++)

{

System.out.print("\*" + " ");

}

System.out.println("");

}

QA febonici series:

### Exceptions

**Description**

**The following code throws a compilation error when executed. How would you handle the error?**

C**hange the code so that it is compiled without an error.**(Hint: A statement in the code throws a checked exception.)

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

class Source {

public static void main(String[] args) {

try{

// Enter a string in the input console

BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));

String inData;

inData = stdin.readLine();

System.out.println("You wrote: " + inData);

}

catch (Exception e) {

System.out.print("String not found");

}

}

}