SOAL BASIC JAVA

1. Write a Java program to print 'Hello' on screen and then print your name on a separate line. Go to the editor

Expected Output :

Hello

Alexandra Abramov

2. Write a Java program to print the sum of two numbers. Go to the editor

Test Data:

74 + 36

Expected Output :

110

3. Write a Java program to divide two numbers and print on the screen. Go to the editor

Test Data :

50/3

Expected Output :

16

Click me to see the solution

4. Write a Java program to print the result of the following operations. Go to the editor

Test Data:

a. -5 + 8 * 6

b. (55+9) % 9

c. 20 + -3*5 / 8

d. 5 + 15 / 3 * 2 - 8 % 3

Expected Output :

43

1

19

13

Click me to see the solution

5. Write a Java program that takes two numbers as input and display the product of two numbers. Go to the editor

Test Data:

Input first number: 25

Input second number: 5

Expected Output :

25 x 5 = 125

Click me to see the solution

6. Write a Java program to print the sum (addition), multiply, subtract, divide and remainder of two numbers. Go to the editor

Test Data:

Input first number: 125

Input second number: 24

Expected Output :

```
125 + 24 = 149
125 - 24 = 101
125 x 24 = 3000
125 / 24 = 5
125 mod 24 = 5
```

Click me to see the solution

7. Write a Java program that takes a number as input and prints its multiplication table upto 10. Go to the editor

Test Data:

Input a number: 8

Expected Output :

```
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
...
8 x 10 = 80
```

Click me to see the solution

8. Write a Java program to display the following pattern. Go to the editor

Sample Pattern :

```
   J    a   v     v  a
   J   a a   v   v  a a
J  J  aaaaa   V V  aaaaa
 JJ  a     a   V  a     a
```

Click me to see the solution

9. Write a Java program to compute the specified expressions and print the output. Go to the editor
Test Data:
((25.5 * 3.5 - 3.5 * 3.5) / (40.5 - 4.5))
Expected Output
2.138888888888889

Click me to see the solution

10. Write a Java program to compute a specified formula. Go to the editor
Specified Formula :
4.0 * (1 - (1.0/3) + (1.0/5) - (1.0/7) + (1.0/9) - (1.0/11))
Expected Output
2.9760461760461765

Click me to see the solution

11. Write a Java program to print the area and perimeter of a circle. Go to the editor
Test Data:
Radius = 7.5
Expected Output
Perimeter is = 47.12388980384689
Area is = 176.71458676442586

Click me to see the solution

12. Write a Java program that takes three numbers as input to calculate and print the average of the numbers. Go to the editor
Click me to see the solution

13. Write a Java program to print the area and perimeter of a rectangle. Go to the editor
Test Data:
Width = 5.5 Height = 8.5

Expected Output
Area is 5.6 * 8.5 = 47.60
Perimeter is 2 * (5.6 + 8.5) = 28.20

Click me to see the solution

14. Write a Java program to print an American flag on the screen. Go to the editor

Expected Output

```
* * * * * * ==================================
 * * * * *  ==================================
* * * * * * ==================================
 * * * * *  ==================================
* * * * * * ==================================
 * * * * *  ==================================
* * * * * * ==================================
 * * * * *  ==================================
* * * * * * ==================================
==============================================
==============================================
==============================================
==============================================
==============================================
==============================================
```

Click me to see the solution

15. Write a Java program to swap two variables. Go to the editor

Click me to see the solution

16. Write a Java program to print a face. Go to the editor

Expected Output

```
 +"""""+
[| o o |]
 |  ^  |
 | '-' |
 +-----+
```

Click me to see the solution

17. Write a Java program to add two binary numbers. Go to the editor

Input Data:

Input first binary number: 10

Input second binary number: 11

Expected Output

```
Sum of two binary numbers: 101
```

Click me to see the solution

18. Write a Java program to multiply two binary numbers. Go to the editor

Input Data:

Input the first binary number: 10
Input the second binary number: 11
Expected Output

```
Product of two binary numbers: 110
```

Click me to see the solution

19. Write a Java program to convert a decimal number to binary number. Go to the editor
Input Data:
Input a Decimal Number : 5
Expected Output

```
Binary number is: 101
```

Click me to see the solution

20. Write a Java program to convert a decimal number to hexadecimal number. Go to the editor
Input Data:
Input a decimal number: 15
Expected Output

```
Hexadecimal number is : F
```

1. Write a Java program to sort a numeric array and a string array. Go to the editor

Click me to see the solution

2. Write a Java program to sum values of an array. Go to the editor

Click me to see the solution

3. Write a Java program to print the following grid. Go to the editor

Expected Output :

```
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
```

Click me to see the solution

4. Write a Java program to calculate the average value of array elements. Go to the editor

Click me to see the solution

5. Write a Java program to test if an array contains a specific value. Go to the editor
Click me to see the solution

6. Write a Java program to find the index of an array element. Go to the editor

Click me to see the solution

1. Write a Java program to convert temperature from Fahrenheit to Celsius degree. Go to the editor

Test Data
Input a degree in Fahrenheit: 212
Expected Output:
212.0 degree Fahrenheit is equal to 100.0 in Celsius

Click me to see the solution

2. Write a Java program that reads a number in inches, converts it to meters. Go to the editor
Note: One inch is 0.0254 meter.
Test Data
Input a value for inch: 1000
Expected Output :
1000.0 inch is 25.4 meters

Click me to see the solution

3. Write a Java program that reads an integer between 0 and 1000 and adds all the digits in the integer. Go to the editor

Test Data
Input an integer between 0 and 1000: 565
Expected Output :
The sum of all digits in 565 is 16

Click me to see the solution

4. Write a Java program to convert minutes into a number of years and days. Go to the editor

Test Data
Input the number of minutes: 3456789
Expected Output :
3456789 minutes is approximately 6 years and 210 days

Click me to see the solution

5. Write a Java program that prints the current time in GMT. Go to the editor

Test Data
Input the time zone offset to GMT: 256
Expected Output:
Current time is 23:40:24

Click me to see the solution

6. Write a Java program to compute body mass index (BMI). Go to the editor

Test Data
Input weight in pounds: 452
Input height in inches: 72
Expected Output:
Body Mass Index is 61.30159143458721

Click me to see the solution

7. Write a Java program to takes the user for a distance (in meters) and the time was taken (as three numbers: hours, minutes, seconds), and display the speed, in meters per second, kilometers per hour and miles per hour (hint: 1 mile = 1609 meters). Go to the editor

Test Data
Input distance in meters: 2500
Input hour: 5
Input minutes: 56
Input seconds: 23
Expected Output :
Your speed in meters/second is 0.11691531
Your speed in km/h is 0.42089513
Your speed in miles/h is 0.26158804

Click me to see the solution

8. Write a Java program that reads a number and display the square, cube, and fourth power. Go to the editor

Expected Output:
Square: .2f

Cube: .2f
Fourth power: 50625.00
Click me to see the solution

9. Write a Java program that accepts two integers from the user and then prints the sum, the difference, the product, the average, the distance (the difference between integer), the maximum (the larger of the two integers), the minimum (smaller of the two integers). Go to the editor

Test Data
Input 1st integer: 25
Input 2nd integer: 5
Expected Output :

```
Sum of two integers: 30
Difference of two integers: 20
Product of two integers: 125
Average of two integers: 15.00
Distance of two integers: 20
Max integer: 25
Min integer: 5
```

Click me to see the solution

1. Write a Java program to get the character at the given index within the String. Go to the editor

Sample Output:

```
Original String = Java Exercises!
The character at position 0 is J
The character at position 10 is i
```

Click me to see the solution

2. Write a Java program to get the character (Unicode code point) at the given index within the String. Go to the editor

Sample Output:

```
Original String : w3resource.com
Character(unicode point) = 51
Character(unicode point) = 101
```

Click me to see the solution

3. Write a Java program to get the character (Unicode code point) before the specified index within the String. Go to the editor

Sample Output:

```
Original String : w3resource.com
Character(unicode point) = 119
Character(unicode point) = 99
```

Click me to see the solution

4. Write a java program to count a number of Unicode code points in the specified text range of a String. Go to the editor

Sample Output:

```
Original String : w3rsource.com
Codepoint count = 9
```

Click me to see the solution

5. Write a java program to compare two strings lexicographically. Go to the editor

Sample Output:

```
String 1: This is Exercise 1
String 2: This is Exercise 2
"This is Exercise 1" is less than "This is Exercise 2"
```

Click me to see the solution

6. Write a java program to compare two strings lexicographically, ignoring case differences. Go to the editor

Sample Output:

```
String 1: This is exercise 1
String 2: This is Exercise 1
"This is exercise 1" is equal to "This is Exercise 1"
```

Click me to see the solution

7. Write a Java program to concatenate a given string to the end of another string. Go to the editor

Sample Output:

```
String 1: PHP Exercises and
String 2: Python Exercises
The concatenated string: PHP Exercises and Python Exercises
```

Click me to see the solution

8. Write a Java program to test if a given string contains the specified sequence of char values. Go to the editor

Sample Output:

```
Original String: PHP Exercises and Python Exercises
Specified sequence of char values: and
true
```

Click me to see the solution

9. Write a Java program to compare a given string to the specified character sequence. Go to the editor

Sample Output:

```
Comparing example.com and example.com: true
Comparing Example.com and example.com: false
```

Click me to see the solution

10. Write a Java program to compare a given string to the specified string buffer. Go to the editor

Sample Output:

```
Comparing example.com and example.com: true
Comparing Example.com and example.com: false
```

Click me to see the solution

11. Write a Java program to create a new String object with the contents of a character array. Go to the editor

Sample Output:

```
The book contains 234 pages.
```

Click me to see the solution

SOAL METHOD

1. Write a Java method to find the smallest number among three numbers. Go to the editor
Test Data:
Input the first number: 25
Input the Second number: 37
Input the third number: 29
Expected Output:

```
The smallest value is 25.0
```

Click me to see the solution

2. Write a Java method to compute the average of three numbers. Go to the editor
Test Data:
Input the first number: 25
Input the second number: 45
Input the third number: 65
Expected Output:

```
The average value is 45.0
```

Click me to see the solution

3. Write a Java method to display the middle character of a string. Go to the editor
Note: a) If the length of the string is odd there will be two middle characters.

b) If the length of the string is even there will be one middle character.
Test Data:
Input a string: 350
Expected Output:

```
The middle character in the string: 5
```

Click me to see the solution

4. Write a Java method to count all vowels in a string. Go to the editor
Test Data:
Input the string: w3resource
Expected Output:

```
Number of  Vowels in the string: 4
```

Click me to see the solution

5. Write a Java method to count all words in a string. Go to the editor
Test Data:
Input the string: The quick brown fox jumps over the lazy dog.
Expected Output:

```
Number of words in the string: 9
```

Click me to see the solution

6. Write a Java method to compute the sum of the digits in an integer. Go to the editor
Test Data:
Input an integer: 25
Expected Output:

```
The sum is 7
```

Click me to see the solution

SOAL SEARCHING

1. Write a Java program to find a specified element in a given array of elements using Binary Search. Go to the editor
Click me to see the solution

2. Write a Java program to find a specified element in a given array of elements using Linear Search. Go to the editor
Click me to see the solution

3. Write a Java program to find a specified element in a given sorted array of elements using Jump Search. Go to the editor
From Wikipedia, in computer science, a jump search or block search refers to a search algorithm for ordered lists. It works by first checking all items L_{km}, where $\Re \in \aleph$ and m is the block size, until an item is found that is larger than the search key. To find the exact position of the search key in the list a linear search is performed on the sublist $L_{[(k-1)m,\ km]}$.
Click me to see the solution

SOAL SORTING

1. Write a Java program to sort an array of given integers using Quick sort Algorithm. Go to the editor
Quick sort is a comparison sort, meaning that it can sort items of any type for which a "less-than" relation (formally, a total order) is defined.
Click me to see the solution

2. Write a Java program to sort an array of given integers using the Bubble sorting Algorithm. Go to the editor
According to Wikipedia "Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller elements "bubble" to the top of the list. Although the algorithm is simple, it is too slow and impractical for most problems even when compared to insertion sort. It can be practical if the input is usually in sort order but may occasionally have some out-of-order elements nearly in

position."
Click me to see the solution

SOAL MATH

1. Write a Java program to round up the result of integer division. Go to the editor
Click me to see the solution

2. Write a Java program to get whole and fractional parts from a double value. Go to the editor
Click me to see the solution

3. Write a Java program to test if a double number is an integer. Go to the editor
Click me to see the solution

4. Write a Java program to round a float number to specified decimals. Go to the editor
Click me to see the solution

5. Write a Java program to count the absolute distinct value in an array. Go to the editor
Click me to see the solution

6. Write a Java program to reverse an integer number. Go to the editor
Click me to see the solution

7. Write a Java program to convert Roman number to an integer number. Go to the editor
Click me to see the solution

8. Write a Java program to convert an integer value to absolute value. Go to the editor
Click me to see the solution

9. Write a Java program to convert a float value to absolute value. Go to the editor
Click me to see the solution

10. Write a Java program to accept a float value of number and return a rounded float value. Go to the editor
Click me to see the solution

SOAL NUMBER

1. Write a Java program to check whether a given number is an ugly number. Go to the editor
In number system, ugly numbers are positive numbers whose only prime factors are 2, 3 or 5. First 10 ugly numbers are 1, 2, 3, 4, 5, 6, 8, 9, 10, 12. By convention, 1 is included.
Test Date:Input an integer number: 235
Expected Output :
It is not an ugly number.

Click me to see the solution

2. Write a Java program to classify Abundant, deficient and perfect number (integers) between 1 to 10,000. Go to the editor
In number theory, an abundant number is a number for which the sum of its proper divisors is greater than the number itself.
Example :
The first few abundant numbers are:
12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56, 60, 66, 70, 72, 78, 80, 84, 88, 90, 96, 100, 102,…
The integer 12 is the first abundant number. Its proper divisors are 1, 2, 3, 4 and 6 for a total of 16.
Deficient number: In number theory, a deficient number is a number n for which the sum of divisors σ(n)<2n, or, equivalently, the sum of proper divisors (or aliquot sum) s(n)<n. The value 2n − σ(n) (or n − s(n)) is called the number's deficiency.
As an example, divisors of 21 are 1, 3 and 7, and their sum is 11. Because 11 is less than 21, the number 21 is deficient. Its deficiency is 2 × 21 − 32 = 10.
The first few deficient numbers are:
1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 21, 22, 23, 25, 26, 27, 29, 31, 32, 33, …….
Perfect number: In number system, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself.

Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself) i.e. σ1(n) = 2n.
The first perfect number is 6. Its proper divisors are 1, 2, and 3, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.
Expected Output :
Number Counting [(integers) between 1 to 10,000]:
Deficient number: 7508
Perfect number: 4
Abundant number: 2488

Click me to see the solution

3. Write a Java program to generate random integers in a specific range. Go to the editor
Click me to see the solution

4. Write a Java program to generate and show all Kaprekar numbers less than 1000. Go to the editor
Expected Output :

```
1        1          0 + 1
9        81         8 + 1
45       2025       20 + 25
55       3025       30 + 25
99       9801       98 + 01
297      88209      88 + 209
703      494209     494 + 209
999      998001     998 + 001
8 Kaprekar numbers.
```

Click me to see the solution

5. Write a Java program to find the number of seed Lychrel number candidates and related numbers for n in the range 1..10000 inclusive. (With that iteration limit of 500). Go to the editor
A Lychrel number is a natural number that cannot form a palindrome through the iterative process of repeatedly reversing its digits and adding the resulting numbers. This process is sometimes called the 196-algorithm, after the most famous number associated with the process.
The first few Lychrel numbers are 196, 295, 394, 493, 592, 689, 691, 788, 790, 879, 887, ...

Expected Output :
5 Lychrel seeds: [196, 879, 1997, 7059, 9999]
244 Lychrel related
5 Lychrel palindromes: [196, 879, 1997, 7059, 9999]

Click me to see the solution

6. Write a Java program to generate and show the first 15 narcissistic decimal numbers. Go to the editor
Expected Output :

```
0 1 2 3 4 5 6 7 8 9 153 370 371 407 1634
```

Click me to see the solution

7. Write a Java program to display first 10 lucus numbers. Go to the editor
The Lucas numbers or series are an integer sequence named after the mathematician François Édouard Anatole Lucas, who studied both that sequence and the closely related Fibonacci numbers. Lucas numbers and Fibonacci numbers form complementary instances of Lucas sequences.
The sequence of Lucas numbers is: 2, 1, 3, 4, 7, 11, 18, 29, ….
Expected Output :

```
   First ten Lucas a numbers:
2
1
3
4
7
11
18
29
47
76
```

Click me to see the solution

8.Write a Java program to print out the first 10 Catalan numbers by extracting them from Pascal's triangle. Go to the editor
In combinatorial mathematics, the Catalan numbers form a sequence of natural numbers that occur in various counting problems, often involving recursively-defined objects. They are named after the Belgian mathematician Eugène Charles Catalan.
The first Catalan numbers for n = 0, 1, 2, 3, … are 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845, 35357670,

129644790, 477638700, 1767263190, 6564120420, 24466267020, 91482563640, 343059613650, 1289904147324, 4861946401452,

```
List 10 Catalan numbers:-

1
2
5
14
42
132
429
1430
4862
16796
```

Click me to see the solution

9. Write a Java program to find and print the first 10 happy numbers. Go to the editor

Happy number: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1, or it loops endlessly in a cycle which does not include 1.

Example: 19 is a happy number

$1^2 + 9^2$=82

$8^2 + 2^2$=68

$6^2 + 8^2$=100

$1^2 + 0^2 + 0^2$=1

Expected Output

```
First 10 Happy numbers:
1
7
10
13
19
23
28
31
```

Click me to see the solution

10. Write a Java program to check whether a given number is a happy number or unhappy number. Go to the editor

Happy number: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number

equals 1, or it loops endlessly in a cycle which does not include 1.
An unhappy number is a number that is not happy.
The first few unhappy numbers are 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 20.

Expected Output

```
Input a number: 5
Unhappy Number
```

Click me to see the solution

SOAL CONDITIONAL STATEMENT

1. Write a Java program to get a number from the user and print whether it is positive or negative. Go to the editor

Test Data
Input number: 35
Expected Output :
Number is positive

Click me to see the solution

2. Write a Java program to solve quadratic equations (use if, else if and else). Go to the editor

Test Data
Input a: 1
Input b: 5
Input c: 1
Expected Output :
The roots are -0.20871215252208009 and -4.7912878474779195

Click me to see the solution

3. Take three numbers from the user and print the greatest number. Go to the editor

Test Data
Input the 1st number: 25

Input the 2nd number: 78
Input the 3rd number: 87
Expected Output :
The greatest: 87

Click me to see the solution

4. Write a Java program that reads a floating-point number and prints "zero" if the number is zero. Otherwise, print "positive" or "negative". Add "small" if the absolute value of the number is less than 1, or "large" if it exceeds 1,000,000. Go to the editor

Test Data
Input a number: 25
Expected Output :
Input value: 25
Positive number

Click me to see the solution

5. Write a Java program that keeps a number from the user and generates an integer between 1 and 7 and displays the name of the weekday. Go to the editor

Test Data
Input number: 3
Expected Output :
Wednesday

Click me to see the solution

6. Write a Java program that reads in two floating-point numbers and tests whether they are the same up to three decimal places. Go to the editor

Test Data
Input floating-point number: 1256
Input floating-point another number: 3254
Expected Output :
They are different

Click me to see the solution

7. Write a Java program to find the number of days in a month. Go to the editor

Test Data
Input a month number: 2
Input a year: 2016
Expected Output :
February 2016 has 29 days

Click me to see the solution

8. Write a Java program that takes the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message. Go to the editor

Test Data
Input an alphabet: p
Expected Output :
Input letter is Consonant

Click me to see the solution

9. Write a Java program that takes a year from user and print whether that year is a leap year or not. Go to the editor

Test Data
Input the year: 2016
Expected Output :
2016 is a leap year

Click me to see the solution

10. Write a program in Java to display the first 10 natural numbers. Go to the editor

Expected Output :

```
The first 10 natural numbers are:

1
2
3
```

```
4
5
6
7
8
9
10
```

Click me to see the solution