Q1: Voting Eligibility Checker

/* Q1: Write a program to take user input for the age of all 10 students in a class and check whether the student can vote depending on his/her age is greater or equal to 18.

Hint:

Define an array of 10 integer elements and take user input for the student's age.

Loop through the array using the length property and for the element of the array check

If the age is a negative number print an invalid age and if 18 or above, print:

The student with the age ___ can vote. Otherwise, print The student with the age ___ cannot vote. */

```
import java.util.Scanner;


public class VotingEligibility {
 @SuppressWarnings("ConvertToTryWithResources")
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 int[] ages = new int[10];

 // Taking user input for all student ages at once
 System.out.println("Enter ages of 10 students:");
 for (int i = 0; i < ages.length; i++) {
 ages[i] = scanner.nextInt();
 }

 // Checking voting eligibility
 for (int age : ages) {
```

```
if (age < 0) {
System.out.println("Invalid age: " + age);

} else if (age >= 18) {

System.out.println("The student with the age " + age + " can vote.");  } else {

System.out.println("The student with the age " + age + " cannot vote.");  }

}



scanner.close();

}

}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac VotingEligibility.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java VotingEligibility
Enter ages of 10 students:
18 17 17 18 19 20 19 17 18 19
The student with the age 18 can vote.
The student with the age 17 cannot vote.
The student with the age 17 cannot vote.
The student with the age 18 can vote.
The student with the age 19 can vote.
The student with the age 20 can vote.
The student with the age 19 can vote.
The student with the age 17 cannot vote.
The student with the age 18 can vote.
The student with the age 19 can vote.
```

Q2: Number Check (Positive, Negative, Zero)

/*Q2: Write a program to take user input for 5 numbers and check whether a number is positive,negative,

or zero.

Further for positive numbers check if the number is even or odd.

Finally compare the first and last elements of the array and display if they equal, greater or

less Hint =>

Define an integer array of 5 elements and get user input to store in the array.

Loop through the array using the length If the number is positive, check for even or odd numbers and print

accordingly If the number is negative, print negative. Else if the number is zero, print zero.

Finally compare the first and last element of the array and display if they equal, greater or less*/

```
import java.util.Scanner;


public class NumberCheck {
 @SuppressWarnings("ConvertToTryWithResources")
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 int[] numbers = new int[5];

 System.out.println("Enter 5 numbers separated by spaces:");
 for (int i = 0; i < 5; i++) numbers[i] = scanner.nextInt();

 for (int num : numbers)
 System.out.println(num + (num > 0 ? (num % 2 == 0 ? " is positive and even." : " is positive and odd.") : num < 0 ? " is
negative." : " is zero."));

 System.out.println("First element " + numbers[0] + (numbers[0] > numbers[4] ? " is greater than " : numbers[0] <
numbers[4] ? " is less than " : " is equal to ") + "last element " + numbers[4] + ".");

 scanner.close();
 }
}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac NumberCheck.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java NumberCheck
Enter 5 numbers separated by spaces:
6 7 8 9 10
6 is positive and even.
7 is positive and odd.
8 is positive and even.
9 is positive and odd.
10 is positive and even.
First element 6 is less than last element 10.
```

Q3: Multiplication Table Generator

/* Q3: Create a program to print a multiplication table of a number.

Hint =>

Get an integer input and store it in the number variable. Also, define a integer array to store the results of multiplication from 1 to 10

Run a loop from 1 to 10 and store the results in the multiplication table array

Finally, display the result from the array in the format number * i = ___ */

```
import java.util.Scanner;

public class MultiplicationTable {
 @SuppressWarnings("ConvertToTryWithResources")
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);

 System.out.print("Enter a number: ");
 int number = scanner.nextInt();
```

```
int[] table = new int[10];


for (int i = 1; i <= 10; i++) table[i - 1] = number * i;


for (int i = 1; i <= 10; i++)

System.out.println(number + " * " + i + " = " + table[i - 1]);


scanner.close();  }

}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac MultiplicationTable.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java MultiplicationTable
Enter a number: 7
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70
```

Q4: Store and Sum Numbers

/*Q4: Write a program to store multiple values in an array up to a maximum of 10 or until the user enters a 0 or a negative number. Show all the numbers as well as the sum of all numbers

Hint =>

Create a variable to store an array of 10 elements of type double as well as a variable to store the total of type double initializes to 0.0. Also, the index variable is initialized to 0 for the array

Use infinite while loop as in while (true)

Take the user entry and check if the user entered 0 or a negative number to break the loop

Also, break from the loop if the index has a value of 10 as the array size is limited to 10.

If the user entered a number other than 0 or a negative number inside the while loop then assign the number to the array element and increment the index value

Take another for loop to get the values of each element and add it to the total

Finally display the total value*/

```
import java.util.Scanner;

public class StoreAndSumNumbers {
 @SuppressWarnings("ConvertToTryWithResources")
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 double[] numbers = new double[10];
 double total = 0.0;
 int index = 0;

 System.out.println("Enter numbers (up to 10). Enter 0 or a negative number to stop:");

 while (true) {
 double num = scanner.nextDouble();  if
(num <= 0 || index == 10) break;
numbers[index++] = num;
 }

 System.out.println("Numbers entered:");  for
(int i = 0; i < index; i++) {
 System.out.print(numbers[i] + " ");  total
```

```
+= numbers[i];

 }


 System.out.println("\nTotal sum: " + total);

scanner.close();

 }

}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac StoreAndSumNumbers.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java StoreAndSumNumbers
Enter numbers (up to 10). Enter 0 or a negative number to stop:
8 7 9 6 15 12 3 4 2 1
0
Numbers entered:
8.0 7.0 9.0 6.0 15.0 12.0 3.0 4.0 2.0 1.0
Total sum: 67.0
```

Q5: Multiplication Table from 6 to 9

```
/*Q5: Create a program to find the multiplication table of a number entered by the user from 6 to 9 and display the result

Hint =>

Take integer input and store it in the variable number as well as define an integer array to store the multiplication result

in the variable multiplicationResult

Using a for loop, find the multiplication table of numbers from 6 to 9 and save the result in the

array Finally, display the result from the array in the format number * i = ___*/


import java.util.Scanner;


public class SixTable {

 @SuppressWarnings("ConvertToTryWithResources")

 public static void main(String[] args) {

 Scanner scanner = new Scanner(System.in);
```

```
System.out.print("Enter a number: ");

int number = scanner.nextInt();

scanner.close();



int[] multiplicationResult = new int[4];



for (int i = 6; i <= 9; i++) {

multiplicationResult[i - 6] = number * i;

}



System.out.println("Multiplication table of " + number + " from 6 to 9:");

for (int i = 6; i <= 9; i++) {
System.out.println(number + " * " + i + " = " + multiplicationResult[i - 6]);  }

}

}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac SixTable.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java SixTable
Enter a number: 8
Multiplication table of 8 from 6 to 9:
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
```

Q6: Mean Height Calculator

/*Q6: Create a program to find the mean height of players present in a football team.

Hint =>

The formula to calculate the mean is: mean = sum of all elements / number of elements

Create a double array named heights of size 11 and get input values from the user.

Find the sum of all the elements present in the array.

Divide the sum by 11 to find the mean height and print the mean height of the football team*/

```
import java.util.Scanner;



public class MeanHeight {

 @SuppressWarnings("ConvertToTryWithResources")

 public static void main(String[] args) {

 Scanner scanner = new Scanner(System.in);

 double[] heights = new double[11]; // Array to store heights of 11 players

double sum = 0;



 // Taking input from user

 System.out.println("Enter the heights of 11 players:");

 for (int i = 0; i < 11; i++) {

 heights[i] = scanner.nextDouble();

 sum += heights[i]; // Adding each height to sum

 }



 scanner.close();
 // Calculating mean height

 double mean = sum / 11;



 // Printing the result

 System.out.printf("The mean height of the football team is: %.2f\n", mean);  }

}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac MeanHeight.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java MeanHeight
Enter the heights of 11 players:
171 169 150 164 165 180 172 176 171 169 168
The mean height of the football team is: 168.64
```

Q7: Odd and Even Numbers Arrays

/*Q7: Create a program to save odd and even numbers into odd and even arrays between 1 to the number entered by the user. Finally, print the odd and even numbers array

Hint =>

Get an integer input from the user, assign it to a variable number, and check for Natural Number. If not a natural number then print an error and exit the program

Create an integer array for even and odd numbers with size = number / 2 + 1

Create index variables for odd and even numbers and initialize them to zero

Using a for loop, iterate from 1 to the number, and in each iteration of the loop, save the odd or even number into the corresponding array

Finally, print the odd and even numbers array using the odd and even index*/

```
import java.util.Scanner;


public class OddEvenArrays {
 public static void main(String[] args) {
 @SuppressWarnings("resource")
 Scanner scanner = new Scanner(System.in);
 System.out.print("Enter a natural number: ");
 int number = scanner.nextInt();
 if (number < 1) {
 System.out.println("Error: Enter a natural number.");
```

```
return;

}


int[] even = new int[number / 2 + 1], odd = new int[number / 2 + 1];
int eIdx = 0, oIdx = 0;


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

if (i % 2 == 0) even[eIdx++] = i;

else odd[oIdx++] = i;

}


System.out.print("Even numbers: ");

for (int i = 0; i < eIdx; i++) System.out.print(even[i] + " ");

System.out.print("\nOdd numbers: ");

for (int i = 0; i < oIdx; i++) System.out.print(odd[i] + " ");  }

}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac OddEvenArrays.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java OddEvenArrays
Enter a natural number: 6
Even numbers: 2 4 6
Odd numbers: 1 3 5
```

Q8: Find Factors of a Number

/*Q8: Create a program to find the factors of a number taken as user input, store the factors in an array, and display the factors

Hint =>

Take the input for a number

Find the factors of the number and save them in an array. For this create integer variable maxFactor and initialize to 10,

factors array of size maxFactor and index variable to reflect the index of the array.

To find factors loop through the numbers from 1 to the number, find the factors, and add them to the array element by incrementing the index. If the index is equal to maxIndex, then need factors array to store more elements To store more elements, reset the maxIndex to twice its size, use the temp array to store the elements from the factors array, and eventually assign the factors array to the temp array

Finally, Display the factors of the number*/

```
import java.util.Arrays;

import java.util.Scanner;


public class FindFactors {

 @SuppressWarnings("ConvertToTryWithResources")

 public static void main(String[] args) {

 Scanner scanner = new Scanner(System.in);

 System.out.print("Enter a number: ");

 int number = scanner.nextInt();

 scanner.close();


 int maxFactor = 10, index = 0;

 int[] factors = new int[maxFactor];
 for (int i = 1; i <= number; i++) {

 if (number % i == 0) {

 if (index == maxFactor) {

 maxFactor *= 2;

 factors = Arrays.copyOf(factors, maxFactor);  }

 factors[index++] = i;

 }
```

```
}


 System.out.print("Factors of " + number + ": ");  for

(int i = 0; i < index; i++) {

 System.out.print(factors[i] + " ");

 }

 }

}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac FindFactors.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java FindFactors
Enter a number: 9
Factors of 9: 1 3 9
```

Q9: 2D to 1D Array Converter

/*Q9: Working with Multi-Dimensional Arrays. Write a Java program to create a 2D Array and Copy the 2D Array into a single dimension array

Hint =>

Take user input for rows and columns, create a 2D array (Matrix), and take the user input

Copy the elements of the matrix to a 1D array. For this create a 1D array of size rows*columns as in int[] array = new int[rows * columns];

Define the index variable and Loop through the 2D array. Copy every element of the 2D array into the 1D array and increment the index

Note: For looping through the 2D array, you will need Nested for loop, Outer for loop for rows, and the inner for loops to access each element*/

```
import java.util.Scanner;
```

```
public class MultiDimensionalArray {

 @SuppressWarnings("ConvertToTryWithResources")

 public static void main(String[] args) {

 Scanner scanner = new Scanner(System.in);



 // Taking user input for rows and columns

 System.out.print("Enter the number of rows: ");

 int rows = scanner.nextInt();

 System.out.print("Enter the number of columns: ");

 int columns = scanner.nextInt();



 // Creating a 2D array
 int[][] matrix = new int[rows][columns];



 // Taking user input for the 2D array elements

System.out.println("Enter the elements of the matrix:");  for

(int i = 0; i < rows; i++) {

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

 matrix[i][j] = scanner.nextInt();

 }

 }



 // Creating a 1D array to copy elements from 2D array

int[] array = new int[rows * columns];

 int index = 0;
```

```
 // Printing the 2D array
 System.out.println("The elements in the 2D array:");  for
(int i = 0; i < rows; i++) {
 for (int j = 0; j < columns; j++) {
 System.out.print(matrix[i][j] + " ");
 }
 System.out.println(); // Move to the next line after each row  }



 // Copying elements from 2D array to 1D array  for
(int i = 0; i < rows; i++) {
 for (int j = 0; j < columns; j++) {
 array[index++] = matrix[i][j];
 }
 }

 // Printing the 1D array
 System.out.println("The elements in the 1D array:");  for
(int num : array) {
 System.out.print(num + " ");
 }

 scanner.close();
 }
```

}

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac MultiDimensionalArray.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java MultiDimensionalArray
Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the matrix:
1 2 3
4 5 6
7 8 9
The elements in the 2D array:
1 2 3
4 5 6
7 8 9
The elements in the 1D array:
1 2 3 4 5 6 7 8 9
```

Q10: FizzBuzz Generator

/*Q10: Write a program FizzBuzz, take a number as user input and if it is a positive integer loop from 0 to the number and save the number, but for multiples of 3 save "Fizz" instead of the number, for multiples of 5 save "Buzz", and for multiples of both save "FizzBuzz". Finally, print the array results for each index position in the format Position 1 = 1, ?, Position 3 = Fizz,...

Hint =>

Create a String Array to save the results and

Finally, loop again to show the results of the array based on the index position*/

```
import java.util.Scanner;


public class FizzBuzz {
 @SuppressWarnings({ "ConvertToTryWithResources", "resource" })
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);


 // Taking user input
 System.out.print("Enter a positive integer: ");
```

```
int number = scanner.nextInt();



if (number <= 0) {

System.out.println("Please enter a positive integer.");

return;

}



// Creating a String array to store results
String[] results = new String[number + 1];

// Looping from 1 to the given number

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

if (i % 3 == 0 && i % 5 == 0) {

results[i] = "FizzBuzz";

} else if (i % 3 == 0) {

results[i] = "Fizz";

} else if (i % 5 == 0) {

results[i] = "Buzz";

} else {

results[i] = String.valueOf(i);

}

}



 // Printing the results with index positions  for

(int i = 0; i <= number; i++) {

 System.out.println("Position " + i + " = " + results[i]);  }
```

```
scanner.close();

}

}
```

```
C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>javac FizzBuzz.java

C:\Users\touri\Documents\STEP-Bootcamp\Java Array Concepts\Level 1>java FizzBuzz
Enter a positive integer: 8
Position 0 = null
Position 1 = 1
Position 2 = 2
Position 3 = Fizz
Position 4 = 4
Position 5 = Buzz
Position 6 = Fizz
Position 7 = 7
Position 8 = 8
```