**TASK 1a:**

**PROGRAM:**

import java.util.Scanner;

public class task1a {

public static int squareRoot(int x) {

if (x == 0 || x == 1) {

return x;

}

int start = 1, end = x, ans = 0;

while (start <= end) {

int mid = start + (end - start) / 2;

if (mid == x / mid) {

return mid;

}

if (mid <= x / mid) {

start = mid + 1;

ans = mid;

} else {

end = mid - 1;

}

}

return ans;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int x = sc.nextInt();

int result = squareRoot(x);

System.out.println("Square root (rounded down): " + result);

sc.close();

}

}

**TASK 1b:**

**PROGRAM:**

import java.util.Scanner;

public class task1b{

public static boolean isUgly(int num) {

if (num <= 0) return false;

while (num % 2 == 0) num /= 2;

while (num % 3 == 0) num /= 3;

while (num % 5 == 0) num /= 5;

return num == 1;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int input = scanner.nextInt();

if (isUgly(input)) {

System.out.println(input + " is an Ugly Number");

} else {

System.out.println(input + " is NOT an Ugly Number");

}

scanner.close();

}

}

**TASK 1c:**

**PROGRAM:**

import java.util.Scanner;

import java.util.Arrays;

public class task1c {

public static int[] productExceptSelf(int[] nums) {

int n = nums.length;

int[] result = new int[n];

int[] left = new int[n];

int[] right = new int[n];

left[0] = 1;

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

left[i] = nums[i - 1] \* left[i - 1];

}

right[n - 1] = 1;

for (int i = n - 2; i >= 0; i--) {

right[i] = nums[i + 1] \* right[i + 1];

}

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

result[i] = left[i] \* right[i];

}

return result;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of the array: ");

int size = scanner.nextInt();

int[] nums = new int[size];

System.out.println("Enter " + size + " integers separated by space or enter:");

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

nums[i] = scanner.nextInt();

}

int[] result = productExceptSelf(nums);

System.out.println("Output: " + Arrays.toString(result));

scanner.close();

}

}

**TASK 2a:**

**PROGRAM:**

import java.util.\*;

public class task2a {

public static int[][] intervalIntersection(int[][] firstList, int[][] secondList) {

List<int[]> result = new ArrayList<>();

int i = 0, j = 0;

while (i < firstList.length && j < secondList.length) {

int start1 = firstList[i][0];

int end1 = firstList[i][1];

int start2 = secondList[j][0];

int end2 = secondList[j][1];

int startMax = Math.max(start1, start2);

int endMin = Math.min(end1, end2);

if (startMax <= endMin) {

result.add(new int[]{startMax, endMin});

}

if (end1 < end2) {

i++;

} else {

j++;

}

}

return result.toArray(new int[result.size()][]);

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of intervals in the first list: ");

int n1 = scanner.nextInt();

int[][] firstList = new int[n1][2];

System.out.println("Enter intervals (start end) for the first list:");

for (int i = 0; i < n1; i++) {

firstList[i][0] = scanner.nextInt();

firstList[i][1] = scanner.nextInt();

}

System.out.print("Enter number of intervals in the second list: ");

int n2 = scanner.nextInt();

int[][] secondList = new int[n2][2];

System.out.println("Enter intervals (start end) for the second list:");

for (int i = 0; i < n2; i++) {

secondList[i][0] = scanner.nextInt();

secondList[i][1] = scanner.nextInt();

}

int[][] intersections = intervalIntersection(firstList, secondList);

System.out.println("\nIntersections:");

for (int[] interval : intersections) {

System.out.println("[" + interval[0] + ", " + interval[1] + "]");

}

scanner.close();

}

}

**TASK 2b:**

**PROGRAM:**

import java.util.Scanner;

import java.util.Arrays;

public class task2b {

public static void merge(int[] arr1, int m, int[] arr2, int n) {

int i = m - 1;

int j = n - 1;

int k = m + n - 1;

while (i >= 0 && j >= 0) {

if (arr1[i] > arr2[j]) {

arr1[k--] = arr1[i--];

} else {

arr1[k--] = arr2[j--];

}

}

while (j >= 0) {

arr1[k--] = arr2[j--];

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of elements in first sorted array: ");

int m = scanner.nextInt();

int[] arr1 = new int[m \* 2];

System.out.println("Enter " + m + " sorted elements for arr1:");

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

arr1[i] = scanner.nextInt();

}

System.out.print("Enter number of elements in second sorted array: ");

int n = scanner.nextInt();

int[] arr2 = new int[n];

System.out.println("Enter " + n + " sorted elements for arr2:");

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

arr2[i] = scanner.nextInt();

}

merge(arr1, m, arr2, n);

System.out.print("Merged Sorted Array: ");

System.out.println(Arrays.toString(Arrays.copyOf(arr1, m + n)));

scanner.close();

}

}

**TASK 2c:**

**PROGRAM:**

import java.util.\*;

public class task2c {

public static List<List<Integer>> threeSum(int[] nums) {

List<List<Integer>> result = new ArrayList<>();

Arrays.sort(nums);

int n = nums.length;

for (int i = 0; i < n - 2; i++) {

if (i > 0 && nums[i] == nums[i - 1]) continue;

int left = i + 1;

int right = n - 1;

while (left < right) {

int sum = nums[i] + nums[left] + nums[right];

if (sum == 0) {

result.add(Arrays.asList(nums[i], nums[left], nums[right]));

while (left < right && nums[left] == nums[left + 1]) left++;

while (left < right && nums[right] == nums[right - 1]) right--;

left++;

right--;

} else if (sum < 0) {

left++;

} else {

right--;

}

}

}

return result;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter integers : ");

String input = scanner.nextLine();

String[] parts = input.split(",");

int[] nums = new int[parts.length];

for (int i = 0; i < parts.length; i++) {

nums[i] = Integer.parseInt(parts[i].trim());

}

List<List<Integer>> triplets = threeSum(nums);

System.out.println("Unique triplets whose sum is zero:");

for (List<Integer> triplet : triplets) {

System.out.println(triplet);

}

scanner.close();

}

}