CS162 ASSIGNMENT 3

NAME:

ARCHIT AGRAWAL

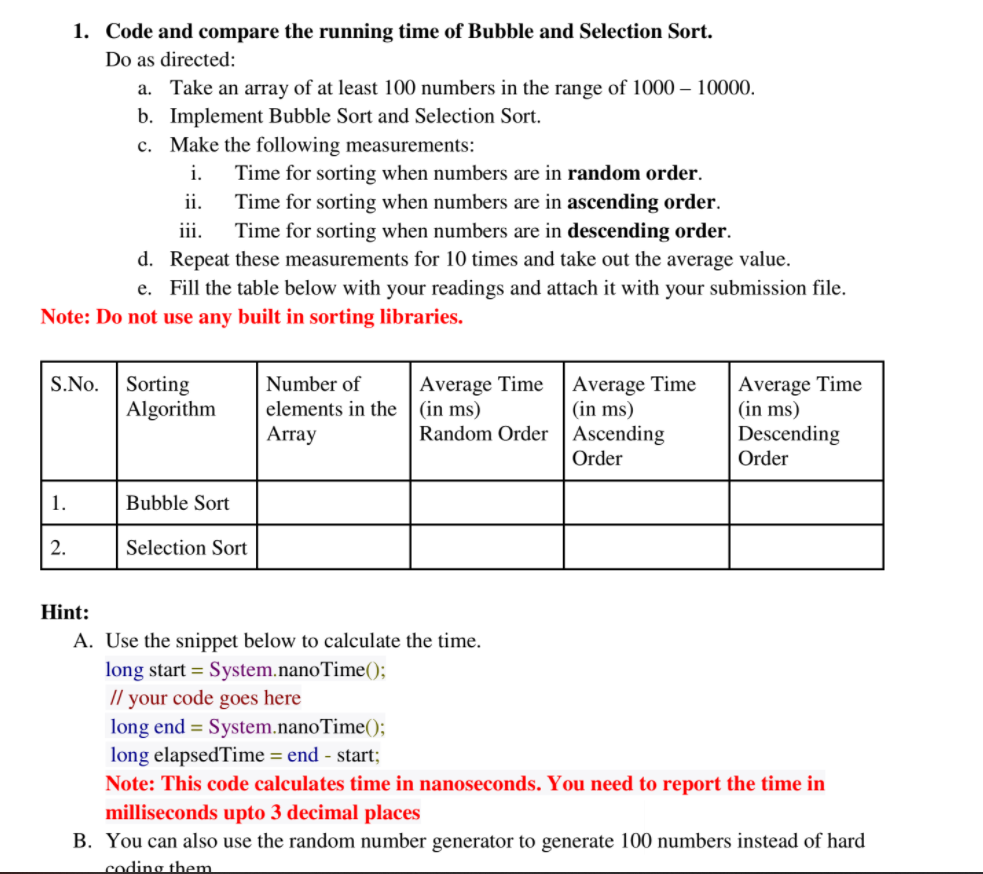
ROLL NO. :

202052307

SECTION:

A

**Question 1**



**Description -:**  The following program asks the user to input the size of array and asks to decide in what order(ascending, descending or random) the user wants to pass the array to sorting methods (bubble sort, selection sort).

The array is generated using random function, hence it is not sorted.

* If the user wants a random sorted to be passed to the sorting methods(bubble sort, selection sort), this array can directly be passed.
* If the user wants an ascending sorted array, this array will be first sorted in ascending order and then it will be passed to the sorting methods.
* If the user wants a descending sorted array, this array will be first sorted in descending order and then it will be passed to the sorting methods.

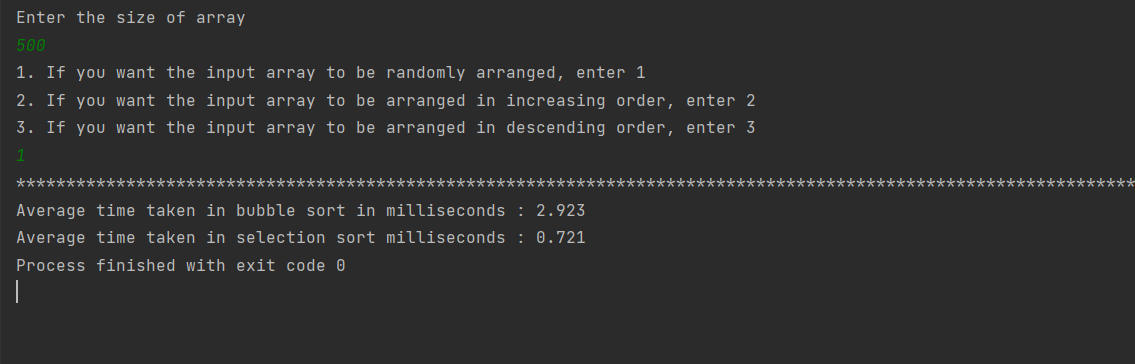
The program will calculate time taken to sort the array for both bubble and selection sort 10 times and gives the output as average time taken to do so for both the sorting methods.

***CODE***

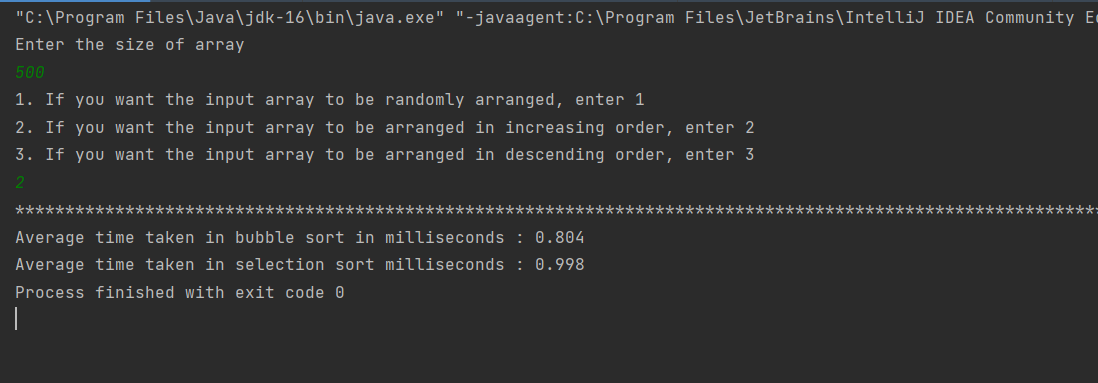
/\* This code is written to compare the average time between selection sort and bubble sort algorithms  
 You will be asked to enter the size of array as input  
 You will be asked to decide the order of input array  
 an array will be generated randomly of that size and time will be calculated for both the algorithms (this will take place 10 times)  
 the average of the time for both the methods will be calculated and printed  
 \*/  
package com.company;  
import java.util.\*;  
import java.util.Random;  
import java.util.Arrays;  
import java.util.Collections;  
  
public class SortingMethods{  
  
 public static double bubbleSort(Integer []array, int n){  
 double start = System.*nanoTime*();  
 //Bubble Sort Algorithm  
 for(int i = 0; i < n ; i++){  
 for(int j = 0; j < n - i - 1; j++){  
 if(array[j] > array[j + 1]){  
 int temp = array[j];  
 array[j] = array[j + 1];  
 array[j + 1] = temp ;  
 }  
 }  
 }  
 double end = System.*nanoTime*();  
 return (end - start)/1000000.0;  
 }  
  
 public static double selectionSort(Integer []array, int n){  
 double start = System.*nanoTime*();  
 //Selection Sort Algorithm  
 for(int i = 1; i <= n; i++){  
 int greatest = array[0];  
 int tempJ = 0;  
 for(int j = 1; j <= n - i ; j++){  
 if(array[j] > greatest ){  
 greatest = array[j] ;  
 tempJ = j ;  
 }  
 }  
 int tempA = array[n - i] ;  
 array[n - i] = greatest;  
 array[tempJ] = tempA;  
 }  
 double end = System.*nanoTime*();  
 return (end - start)/1000000.0;  
 }  
  
 public static void main(String []args){  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter the size of array");  
 int n = sc.nextInt();  
  
 System.*out*.println("1. If you want the input array to be randomly arranged, enter 1");  
 System.*out*.println("2. If you want the input array to be arranged in increasing order, enter 2");  
 System.*out*.println("3. If you want the input array to be arranged in descending order, enter 3");  
 int order = sc.nextInt();  
  
 double avg\_time\_bubble = 0.0;  
 double avg\_time\_selection = 0.0;  
  
 int t = 10;  
 while(t > 0) { //while loop is used to run bubble/selection sort for different arrays and compute the average.  
 Integer[] a = new Integer[n];  
 Integer[] b = new Integer[n]; //a copy of a[]  
 Random rand = new Random();  
  
 for (int i = 0; i < n; i++) {  
 a[i] = rand.nextInt(9000) + 1000;  
 //System.out.print(a[i] + " ");  
 }  
 for(int i = 0; i < n; i++){  
 b[i] = a[i];  
 }  
 //a[] will be bubble sorted  
 //b[] will be selection sorted  
 // as random will give a new number everytime that is why  
 //a copy of a[] is created to ensure that both the sorting  
 //methods gets the same array  
  
 if (order == 2) {  
 Arrays.*sort*(a);  
 Arrays.*sort*(b);  
 } else if (order == 3) {  
 Arrays.*sort*(a, Collections.*reverseOrder*());  
 Arrays.*sort*(b, Collections.*reverseOrder*());  
 }  
  
 double time\_in\_bubble = *bubbleSort*(a, n);  
 double time\_in\_selection = *selectionSort*(b, n);  
 avg\_time\_bubble = avg\_time\_bubble + time\_in\_bubble;  
 avg\_time\_selection = avg\_time\_selection + time\_in\_selection;  
  
 t = t - 1;  
 }  
  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 System.*out*.printf("Average time taken in bubble sort in milliseconds : %.3f ",(avg\_time\_bubble/10.0));  
 System.*out*.println();  
 System.*out*.printf("Average time taken in selection sort milliseconds : %.3f ",(avg\_time\_selection/10.0));  
 }  
}

***OUTPUT***

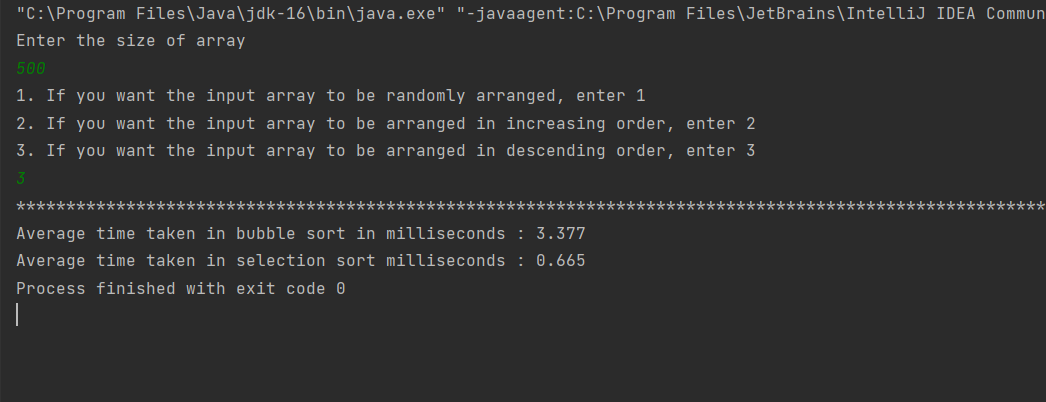
* ***For random ordered input array***



* ***For ascending ordered input array***



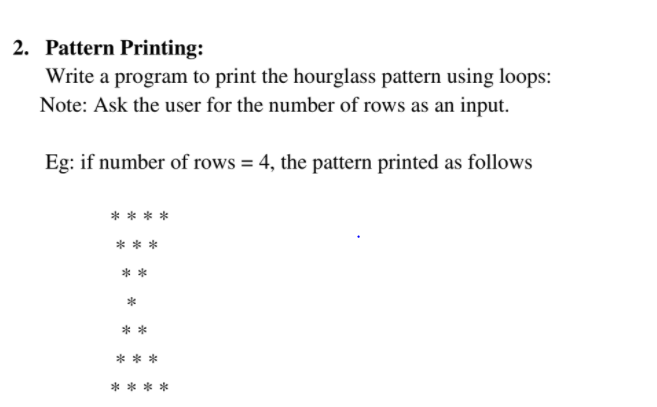
* ***For descending ordered input array***



The above data is tabularized below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No. | Sorting Algorithm | Number of elements in the array | Average Time (in ms) for Random Order Input | Average Time (in ms) for Ascending Order Input | Average Time (in ms) for Descending Order Input |
| 1. | Bubble Sort | 500 | 2.923 | 0.804 | 3.377 |
| 2. | Selection Sort | 500 | 0.721 | 0.998 | 0.665 |

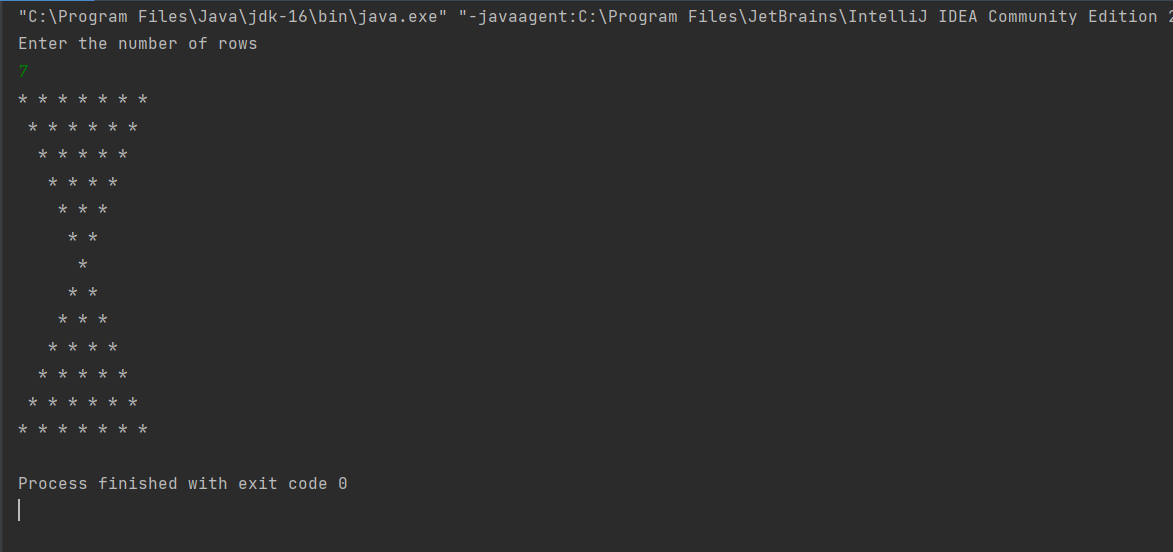
**Question 2**



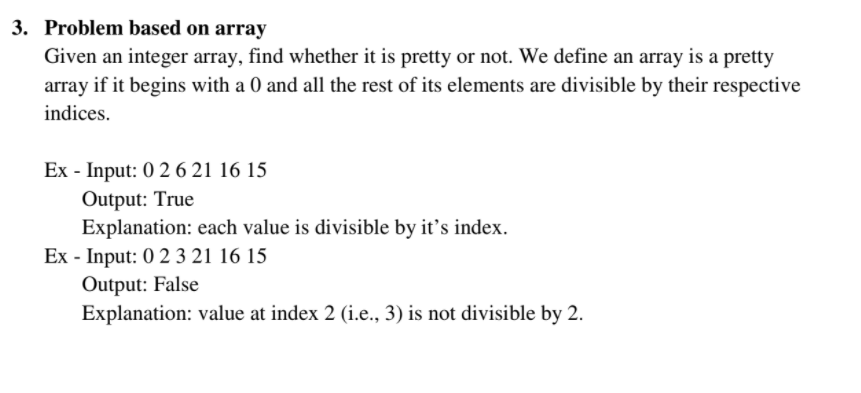
***CODE***

package com.company;  
import java.util.\*;  
  
public class Pattern {  
  
 public static void hourglass(int rows){  
 int n = 2 \* rows - 1;  
 int s;  
 for(int i = 0; i < n; i++){  
 if(i < rows) s = i;  
 else s = n - i - 1;  
  
  
 for(int j = 0; j < s; j++){  
 System.*out*.print(" ");  
 }  
 for(int j = 0; j < rows - s; j++){  
 System.*out*.print("\* ");  
 }  
  
 System.*out*.println();  
 }  
  
 }  
  
 public static void main(String []args){  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter the number of rows");  
 int n = sc.nextInt();  
 *hourglass*(n);  
 }  
}

***OUTPUT***



**Question 3**



***CODE***

package com.company;  
import java.util.\*;  
  
public class PrettyArray{  
  
 public static boolean prettyArray(int []array){  
  
 if(array[0] != 0) return false;  
 for(int i = 1; i < array.length; i++){  
 if(array[i] % i != 0) return false;  
 }  
 return true;  
 }  
  
 public static void main(String [] args){  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("An array is a pretty array if it begins with a 0 and all the rest of its elements are divisible by their respective indices(0 - based).");  
 System.*out*.println();  
 System.*out*.println("Enter the size of the array");  
 int n = sc.nextInt();  
 System.*out*.println("Enter array elements one-by-one with spaces");  
 int[] array = new int[n];  
 for(int i = 0; i < n; i++){  
 array[i] = sc.nextInt();  
 }  
 if(*prettyArray*(array)) System.*out*.println("The given array is a pretty array.");  
 else System.*out*.println("The given array is not a pretty array.");  
 }  
}

***OUTPUT***

