Hackathon

Q1. Consider there is a 3 Boolean variable called a, b, c. Check if at least two out of three Booleans are true  \*

import java.util.Scanner;

public class Q1CheckBooleans {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your word:");

String a = sc.nextLine();

String b = sc.nextLine();

String c = sc.nextLine();

if ((a.equals(b)) && (b.equals(c))) {

System.out.println("All are true");

} else if ((a.equals(b)) || (b.equals(c))) {

System.out.println("Two booloeans variable are True ");

}else {

System.out.println("All are different");

}

}

}Graphical user interface, text, application

Description automatically generated

}

}

Q2. write a program to find factorial (Non Recursive) \*

import java.util.Scanner;

public class Que2Factorial {

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

System.out.print("Enter your digit to find factorial: ");

int input = sc.nextInt();

int result = 1;

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

result = result\*i;

}

System.out.println("Factorial of " +input + " is: "+result);

}

}

Graphical user interface, text

Description automatically generated

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

Q3. Given an array of integers, sort the integer values. \*

import java.util.Scanner;

public class Q3SortArrayInteger {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("How many numbers you want to sort = ");

int size = sc.nextInt();

int arr[] = new int[size];

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

System.out.println("Enter number " + (i + 1) + " = ");

arr[i] = sc.nextInt();

}

System.out.println("Original array is -");

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

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

}

int temp = arr[0];

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

for (int j = 1; j < size - i; j++) {

if (arr[j - 1] > arr[j]) {

temp = arr[j - 1];

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

arr[j] = temp;

}

}

}

System.out.println("\nSorted Array is -");

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

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

}

}

}

Text

Description automatically generated

Q4. Given an array of integers check the Palindrome of the series. \*

import java.util.Scanner;

public class Q4IntergerPalindrom {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your number to check palindrome ");

int count = sc.nextInt();

int[] array = new int[count];

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

System.out.println("Enter number " + (i + 1) + " = ");

array[i] = sc.nextInt();

}

palindrome(array);

}

static void palindrome(int[] arr) {

int flag = 0;

int n = arr.length;

for (int i = 0; i <= n / 2 && n != 0; i++) {

if (arr[i] != arr[n - i - 1]) {

flag = 1;

break;

}

}

if (flag == 1)

System.out.println("Not Palindrome");

else

System.out.println("Palindrome");

}

}

Text

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Q5. Given an array prints the unique numbers and also print the number of occurrences of duplicate numbers. \*

import java.util.Arrays;

import java.util.Scanner;

public class Q5UniqueAndDuplicates {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter you 5 numbers of array ");

String old="";

int arr[]=new int[5];

for(int i=0;i<5;i++) {

arr[i]=sc.nextInt();

old=old+String.valueOf(arr[i]);

}

char[] cArray=old.toCharArray();

Arrays.sort(cArray);

int count =0;

String s="";

boolean flag=false;

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

flag=false;

for(int j=i+1;j<cArray.length;j++) {

if(cArray[i]==cArray[j]) {

flag=true;

break;

}

}

if(!flag) {

s=s+cArray[i];

}

}

String unique="";

String duplicate="";

s=s.trim();

char[] newArray=s.toCharArray();

for(char c:newArray) {

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

if(cArray[i]==c) {

count++;

}

}

System.out.println(c+" appeared "+count+" times");

if(count==1) {

unique= unique+ " "+c;

}else {

duplicate=duplicate+" "+c;

}

count=0;

}

System.out.println("Unique numbers in array are "+unique);

System.out.println("Duplicate numbers in array are "+duplicate);

}}

Text

Description automatically generated

Q6. WJP to perform ascending order Selection sort \*

import java.util.Arrays;

import java.util.Scanner;

public class Q6SelectionSortAscendingOrder {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the numbers in Array ");

int count = sc.nextInt();

int[] array = new int[count];

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

System.out.println("Enter number " + (i + 1) + " = ");

array[i] = sc.nextInt();

}

System.out.println("Elements of array "+Arrays.toString(array));

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

int min = i;

for (int j = i + 1; j < array.length; j++) {

if (array[j] < array[min]) {

min= j;

}

}

int tmp = array[i];

array[i] = array[min];

array[min] = tmp;

}

System.out.println("Elements of array after Selection sort "+Arrays.toString(array));

}

}

Text

Description automatically generated

Q7 What are different ways to create String Object? \*

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

Q8. How can we make String upper case to lower case? \*

import java.util.Scanner;

public class Q7UpperCAseTOlowerCAse {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Your Sentence:");

String sent = sc.nextLine();

sc.close();

System.out.println(sent.toLowerCase());

}

}Graphical user interface, text

Description automatically generated

Q9. How can we make String Lower case to Upper case? \*

import java.util.Scanner;

public class Q8LowerCaseUpperCase {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Your Sentence:");

String sent = sc.nextLine();

sc.close();

System.out.println(sent.toUpperCase());

}

}

Graphical user interface, text, application

Description automatically generated

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

Q12. Write a program to check palindrome (MalayalaM) for both numbers and string? \*

import java.util.Scanner;

public class Q12PalindromeforNumandString {

public static void main(String[] args) {

{

String original, reverse = "";

Scanner in = new Scanner(System.in);

System.out.println("Enter a string/number to check if it is a palindrome");

original = in.nextLine();

int length = original.length();

for (int i = length - 1; i >= 0; i--)

reverse = reverse + original.charAt(i);

if (original.equals(reverse))

System.out.println("Entered string/number is a palindrome.");

else

System.out.println("Entered string/number isn't a palindrome.");

}

}

}

Graphical user interface, text, application

Description automatically generated

Q13. Given a string print the reverse of the string.(Input: Java Code Output: edoC avaJ) \*

import java.util.Scanner;

public class Q13ReverseString {

public static void main(String[] args) {

Scanner scanner=new Scanner(System.in);

System.out.println(" Enter the String ");

String str,str2="";

str=scanner.nextLine();

for(int i=str.length()-1;i>=0;i--) {

str2=str2+str.charAt(i);

}

System.out.println(" Reverse String is --> "+ str2);

scanner.close();

}

}

Graphical user interface, text

Description automatically generated

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

Q14. Given a string print the reverse of the words string.(Input: Java Code Output: Code Java) \*

import java.util.Scanner;

public class Q14ReverseWord {

public static void main(String[] args) {

Scanner scanner=new Scanner(System.in);

String str,str2="";

System.out.println(" Enter string ");

str=scanner.nextLine();

String[] reverse=str.split(" ");

for(int i=reverse.length-1;i>=0;i--) {

str2=str2+reverse[i]+" ";

}

System.out.println(" Reverse String is "+ str2);

scanner.close();

}

}

A picture containing text, screenshot, black

Description automatically generated

Q15. Given a string print the unique words of the string. \*

import java.util.Arrays;

import java.util.HashSet;

import java.util.Scanner;

public class Q15UniqueWords {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your words:");

String str = sc.nextLine();

String[] words = str.split(" ");

HashSet<String> uniqueWords = new HashSet<String>(Arrays.asList(words));

for(String s:uniqueWords)

System.out.println(s);

}

}

Graphical user interface, text, application

Description automatically generated

Q16. Write a method that will remove given character from the String? \*

import java.util.Scanner;

public class Q16RemoveCharFromString {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter your string ");

String str=sc.nextLine();

System.out.println("enter char to remove ");

char c=sc.nextLine().charAt(0);

String newStr=str.replace(c, ' ');

System.out.println(newStr);

}

}

A picture containing text, screenshot, black, screen

Description automatically generated

Q17. WJP to find total number of integers, uppercase and lowercase character in the give string \*

import java.util.Scanner;

public class Q16COuntIntUpperLower {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your string ");

String str = sc.nextLine();

char[] cArray = str.toCharArray();

int digit = 0, uCase = 0, lCase = 0;

for (char c : cArray) {

if ((int) c >= 48 && (int) c <= 57)

digit++;

else if ((int) c >= 97 && (int) c <= 122)

lCase++;

else if ((int) c >= 65 && (int) c <= 90)

uCase++;

}

System.out.println("Digits in the String " + digit);

System.out.println("Upper case char in the String " + uCase);

System.out.println("Lower case in the String " + lCase);

}

}

Graphical user interface, text, application

Description automatically generated

Q18. WJP to display duplicate character in string \*

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class Q18DuplicateCharacter {

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

System.out.println("Enter your String ");

String str=sc.nextLine();

char[] arr=str.toCharArray();

Map<Character,Integer> countMap=new HashMap<Character,Integer>();

for(Character i:arr) {

if((int)i==32) {

}else

if(!countMap.containsKey(i)) {

countMap.put(i, 1);

}

else {

Integer count=countMap.get(i);

count++;

countMap.put(i, count);

}

}

System.out.println("Repeated chars in String = ");

for (Map.Entry<Character,Integer> entry : countMap.entrySet())

if(entry.getValue()!=1) {

System.out.println(entry.getKey()+"-> ocuured " +

+ entry.getValue()+" Times");

}

sc.close();

}

}

Graphical user interface, text, application

Description automatically generated

Q19. WJP to display number of occurrence of all character \*

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class Q19OccuranceOfChar {

public static void main(String[] args) {

Scanner scanner =new Scanner(System.in);

System.out.println("Enter your String ");

String str=scanner.nextLine();

char[] arr=str.toCharArray();

Map<Character,Integer> countMap=new HashMap<Character,Integer>();

for(Character i:arr) {

if((int)i==32) {

}else

if(!countMap.containsKey(i)) {

countMap.put(i, 1);

}

else {

Integer count=countMap.get(i);

count++;

countMap.put(i, count);

}

}

for (Map.Entry<Character,Integer> entry : countMap.entrySet())

System.out.println(entry.getKey()+"-> ocuured " +

+ entry.getValue()+" Times");

scanner.close();

}

}

A screenshot of a computer

Description automatically generated with medium confidence

Q20. WJP to find total number of repeated integers, uppercase and lowercase character in the give string \*

Scanner sc=new Scanner(System.in);

System.out.println("Enter your string to convert in int ");

String s=sc.nextLine();

int i=Integer.valueOf(s);

System.out.println("Converted value in Integer "+i);

A picture containing text, screenshot, screen

Description automatically generated

Q22. WJP to convert int to string \*

Scanner sc=new Scanner(System.in);

System.out.println("Enter your int to convert in String ");

int i=sc.nextInt();

String s=String.valueOf(i);

System.out.println("Converted value in String "+s);

A screenshot of a computer

Description automatically generated with medium confidence

Q23.WJP to differentiate input as string, int or bool \*

import java.util.Scanner;

public class DiffertiateInputStringInt {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

checkMethod();

checkMethod();

checkMethod();

}

private static void checkMethod() {

System.out.println("Enter your string to check ");

boolean flag = true;

Scanner sc1 = new Scanner(System.in);

String s = sc1.next();

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

flag = true;

if (Character.isDigit(s.charAt(i)) == false)

flag = false;

break;

}

if (flag == true) {

System.out.println("It is Digit!");

} else if (s.equalsIgnoreCase("true") || s.equalsIgnoreCase("false")) {

System.out.println("It is boolean!");

} else {

System.out.println("It is String!");

}

}

}

Text

Description automatically generated

Q24. Write a program which inputs a positive natural number N and prints the possible consecutive number combinations, which when added give N. INPUT: N = 9 OUTPUT: 4 + 5 2 + 3+ 4 \*

import java.util.Scanner;

public class Q23ConsecutiveNumbers {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your number ");

int n = sc.nextInt();

int sum = 0, j = 0;

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

sum = i;

j = i + 1;

while (sum < n) {

sum = sum + j;

j++;

}

if (sum == n) {

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

if (k == i)

System.out.print(k);

else

System.out.print(" + " + k);

}

System.out.println();

}

}

sc.close();

}

}

Text

Description automatically generated

import java.util.Scanner;

public class Q25BinarySeach {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int[]arr=new int[5];

for(int i=0;i<5;i++) {

System.out.println("Enter number in array ");

arr[i]=sc.nextInt();

}

System.out.println("enter number to find in array");

int num=sc.nextInt();

int len=arr.length;

int pos=binarySearch(arr,1,len-1,num);

if (pos == -1)

System.out.println("Element not present");

else

System.out.println("Element found at index " + pos);

}

static int binarySearch(int arr[], int l, int r, int x)

{

if (r >= l) {

int mid = l + (r - l) / 2;

if (arr[mid] == x)

return mid;

if (arr[mid] > x)

return binarySearch(arr, l, mid - 1, x);

return binarySearch(arr, mid + 1, r, x);

}

return -1;

}

}

Q25. Write a program for binary search. And 5 i/p has to take from user as binary elements. \*

Text

Description automatically generated

Q26. WJP to merge two sorted array.(Do not use third array) array1[10] = 1,2,4,6,9,10 array2[4] = 3, 5,7,8 After merge : array1[10] = 1,2,3,4,5,6,7,8,9,10 \*

public class Q26MergeTwoSortArray {

final static int INVALID\_NUM = 0;

public void inplaceMergeArrays(int[] arrayA, int[] arrayB) {

int validNumIndex = arrayA.length - 1;

for (int i = arrayA.length - 1; i >= 0; i--) {

if (arrayA[i] != INVALID\_NUM) {

arrayA[validNumIndex] = arrayA[i];

validNumIndex -= 1;

}

}

int i = validNumIndex + 1;

int j = 0, k = 0;

while ((i < arrayA.length) && (j < arrayB.length)) {

if (arrayA[i] < arrayB[j]) {

arrayA[k++] = arrayA[i++];

} else {

arrayA[k++] = arrayB[j++];

}

}

while (j < arrayB.length) {

arrayA[k++] = arrayB[j++];

}

}

public static void main(String[] args) {

Q26MergeTwoSortArray solution = new Q26MergeTwoSortArray();

int[] arrayA = new int[10];

arrayA[0] = 1;

arrayA[1] = 2;

arrayA[2] = 4;

arrayA[3] = 6;

arrayA[4] = 9;

arrayA[5] = 10;

int[] arrayB = { 3, 5, 7, 8 };

solution.inplaceMergeArrays(arrayA, arrayB);

System.out.println("Array after merging elements ");

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

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

}

}

}

Text

Description automatically generated

Q27. WJP to perform ascending order Quick sort \*

import java.util.Arrays;

import java.util.Scanner;

public class Q27Quicksort {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int[]arr=new int[5];

for(int i=0;i<5;i++) {

System.out.println("Enter number in array ");

arr[i]=sc.nextInt();

}

quickSort(arr, 0, arr.length-1);

System.out.println(Arrays.toString(arr));

}

public static void quickSort(int[] arr, int start, int end){

int partition = partition(arr, start, end);

if(partition-1>start) {

quickSort(arr, start, partition - 1);

}

if(partition+1<end) {

quickSort(arr, partition + 1, end);

}

}

public static int partition(int[] arr, int start, int end){

int pivot = arr[end];

for(int i=start; i<end; i++){

if(arr[i]<pivot){

int temp= arr[start];

arr[start]=arr[i];

arr[i]=temp;

start++;

}

}

int temp = arr[start];

arr[start] = pivot;

arr[end] = temp;

return start;

}

Text

Description automatically generated

Q28. WJP to find factorial of a number using recursion \*

import java.util.Scanner;

public class Q27FactorialRecursive {

public static void main(String[] args) {

int n, fact;

Scanner sc = new Scanner(System.in);

System.out.println("Enter number for factorial : ");

n = sc.nextInt();

fact = calculateFactorial(n);

System.out.println("The factorial of number " + n + " is " + fact);

sc.close();

}

public static int calculateFactorial(int n) {

if (n <= 1) {

return 1;

} else {

return n \* (calculateFactorial(n - 1));

}

}

}

A picture containing text, screenshot, screen, black

Description automatically generated

Q29. WJP to perform Merge sort using recursion \*

import java.util.Arrays;

import java.util.Scanner;

public class Q29MergeSortRecursive {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int[]arr=new int[5];

for(int i=0;i<5;i++) {

System.out.println("Enter number in array ");

arr[i]=sc.nextInt();

}

mergeSort(arr,0,arr.length-1);

System.out.println("Array after merge Sort ");

System.out.println(Arrays.toString(arr));

}

public static void mergeSort(int[] array, int low, int high) {

if (high <= low) return;

int mid = (low+high)/2;

mergeSort(array, low, mid);

mergeSort(array, mid+1, high);

merge(array, low, mid, high);

}

public static void merge(int[] array, int low, int mid, int high) {

int leftArray[] = new int[mid - low + 1];

int rightArray[] = new int[high - mid];

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

leftArray[i] = array[low + i];

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

rightArray[i] = array[mid + i + 1];

int leftIndex = 0;

int rightIndex = 0;

for (int i = low; i < high + 1; i++) {

if (leftIndex < leftArray.length && rightIndex < rightArray.length) {

if (leftArray[leftIndex] < rightArray[rightIndex]) {

array[i] = leftArray[leftIndex];

leftIndex++;

} else {

array[i] = rightArray[rightIndex];

rightIndex++;

}

} else if (leftIndex < leftArray.length) {

array[i] = leftArray[leftIndex];

leftIndex++;

} else if (rightIndex < rightArray.length) {

array[i] = rightArray[rightIndex];

rightIndex++;

}

}

}

}

Graphical user interface, text

Description automatically generated

Q30. Write a function to find out longest palindrome in a given string? \*

import java.util.Scanner;

public class Q30LongestPalindrome {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String newString = "";

int len = 0;

System.out.println("Enter the string to check = ");

String str1 = sc.nextLine();

String[] sarray = str1.split(" ");

len = sarray[0].length();

for (String str : sarray) {

int i = 0, j = str.length() - 1;

boolean flag = true;

while (i < j) {

flag = true;

if (str.charAt(i) != str.charAt(j)) {

flag = false;

break;

}

i++;

j--;

}

if (flag == true) {

if (str.length() > len)

newString = str;

}

}

System.out.println("Longest Palindrom is " + newString);

}

}

Graphical user interface, text, application

Description automatically generated

Q31. Read a file content and write it to a new file in reverse order.(reverse line 1-10 to line 10-1) \*

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.util.ArrayList;

import java.util.List;

public class Q31ReadnWrite {

public static class ReadWriteFileReverse {

public static void main(String[] args) throws Exception {

String file1="/Users/jyoti/eclipse-workspace/HACKATHON/src/Jyoti.txt";

FileReader fr=new FileReader(file1);

BufferedReader b=new BufferedReader(fr);

String file="/Users/jyoti/eclipse-workspace/HACKATHON/src/JyotiWrite.txt";

FileWriter fw=new FileWriter(file);

BufferedWriter bw=new BufferedWriter(fw);

String s=" ";

List<String> tmp = new ArrayList<String>();

do{

s = b.readLine();

System.out.println(s);

tmp.add(s);

}while(s!=null);

String str="";

for(int i=tmp.size()-1;i>=0;i--) {

if(tmp.get(i)==null) {

}else {

str=tmp.get(i);

System.out.println(tmp.get(i));

bw.write(str+"\n");

}}

bw.close();

}

}

Graphical user interface, text

Description automatically generated

}Graphical user interface, text

Description automatically generated

Q32. You are given two sorted arrays, A and B, and A has a large enough bu#er at the end to hold B. Write a method to merge B into A in sorted order. \*

public class Q33SortAnB {

final static int INVALID\_NUM = 0;

public void inplaceMergeArrays(int[] arrayA, int[] arrayB) {

int validNumIndex = arrayA.length - 1;

for (int i = arrayA.length - 1; i >= 0; i--) {

if (arrayA[i] != INVALID\_NUM) {

arrayA[validNumIndex] = arrayA[i];

validNumIndex -= 1;

}

}

int i = validNumIndex + 1;

int j = 0, k = 0;

while ((i < arrayA.length) && (j < arrayB.length)) {

if (arrayA[i] < arrayB[j]) {

arrayA[k++] = arrayA[i++];

} else {

arrayA[k++] = arrayB[j++];

}

}

while (j < arrayB.length) {

arrayA[k++] = arrayB[j++];

}

}

public static void main(String[] args) {

Q33SortAnB solution = new Q33SortAnB();

int[] arrayA = new int[10];

arrayA[0] = 1;

arrayA[1] = 2;

arrayA[2] = 4;

arrayA[3] = 6;

arrayA[4] = 9;

arrayA[5] = 10;

int[] arrayB = { 3, 5, 7, 8 };

solution.inplaceMergeArrays(arrayA, arrayB);

System.out.println("Array after merging array B into A ");

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

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

}

}

}

A picture containing text, monitor, screenshot, black

Description automatically generated

Q35. Write test cases for how to test just the withdrawing functionality from ATM ( Minimum 10 test cases required ) \*