Assignment 1

21AIE111

Data Structure and Algorithms – SEM-II

Professor – Dr. Sachin Sir

Submitted By: Vikhyat Bansal CB.EN.U4AIE21076

1. Write a java code to create an array of strings {“Implement”, “a”, “java”,”code”}. Write a function to display the array elements.

CODE:

```
public class arrassignment {


    public static void main(String[] args) {
        String[] array = {"Implement","a","Java","code"};

        for (String element: array){
            System.out.println(element);
        }
    }
}
```

```
[Running] cd "d:\" && javac arrassignment.java && java arrassignment
Implement
a
Java
code

[Done] exited with code=0 in 0.482 seconds
```

Explanation:

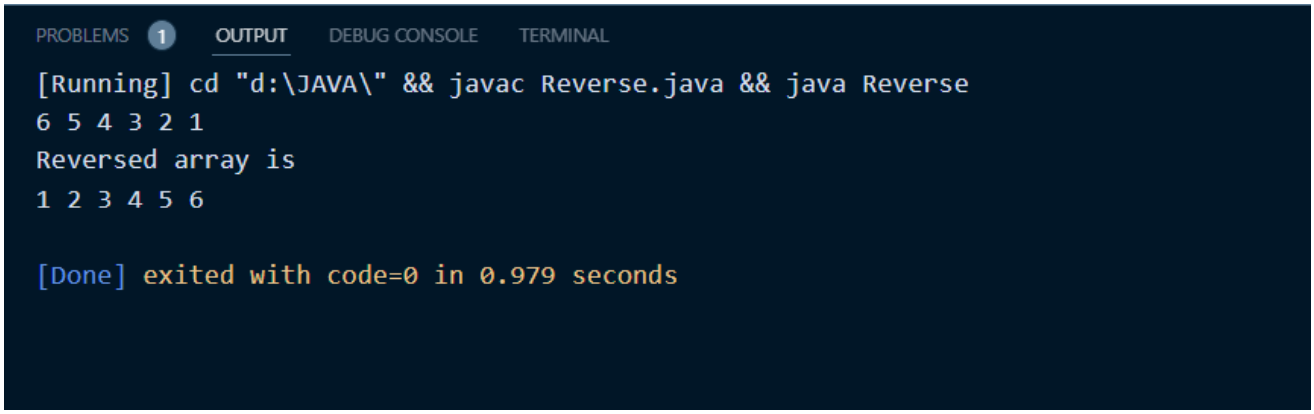
- A string array is being created using { String[] array } and input is given. Now, that array is created.
- We use for-each loop to display all the array elements in order of input { for (String element: array)}
- Using Print line statement the function element is being called which will provide us with the output {System.out.println(element);}.

2. Write a function to reverse the array and display it.

CODE:

```java
// Iterative java program to reverse an
// array
public class GFG {

   /* Function to reverse arr[] from
    start to end*/
    static void rvereseArray(int arr[],
                    int start, int end)
    {
        int temp;

        while (start < end)
        {
            temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }
    }

    /* Utility that prints out an
    array on a line */
    static void printArray(int arr[],
                           int size)
    {
        for (int i = 0; i < size; i++)
             System.out.print(arr[i] + " ");

         System.out.println();
    }

    public static void main(String args[]) {

        int arr[] = {1, 2, 3, 4, 5, 6};
        printArray(arr, 6);
        rvereseArray(arr, 0, 5);
        System.out.print("Reversed array is \n");
        printArray(arr, 6);

    }
}
```

OUTPUT:



Explanation:

- Take input the size of the array and the elements of the array.
- Consider a function reverse which takes the parameters-the array (say arr) and the size of the array(say n).
- In this way, all the elements of the array arr[] are placed reversely in the old array.
- Further, we can iterate through the old array from the beginning and print the elements of the array.
- We swap the value of each element from beginning of the array. Element 1 is swapped with n, 2 with n-1 and so on.

3. Write a function to concatenate any two array elements and display the output.

CODE A:

```
public class Q2 {

    public static void main(String args[]) {
    String stringArray[] = {"Implement ", "Java"};
     StringBuffer sb = new StringBuffer();
     for(int i = 0; i < stringArray.length; i++) {
      sb.append(stringArray[i]);
  }
    String str = sb.toString();
    System.out.println(str);
    }
    }
```

OUTPUT A:

```
[Running] cd "d:\" && javac Q2.java && java Q2
Implement Java

[Done] exited with code=0 in 0.538 seconds
```

Explanation A:

- String buffer is same as string except string buffer is mutable i.e. changeable.
- The append () method concatenates the given argument with this String.
- The **toString()** method of **StringBuffer** class can be used to convert StringBuffer content to a String. This method returns a String object that represents the contents of StringBuffer.

CODE B:

```java
import java.util.*;


class Q2{

// Function to find the integer value
// obtained by joining array elements
// together
static int ConcatenateArr(int[] arr, int N)
{

    // Stores the resulting integer value
    int ans = arr[0];

    // Traverse the array arr[]
    for(int i = 1; i < N; i++)
    {

        // Stores the count of digits of arr[i] in int l
        int l = (int)Math.floor(Math.log10(arr[i]) + 1);

        // Update ans
        ans = ans * (int)Math.pow(10, l);

        // Increment ans by arr[i]
        ans += arr[i];
    }

    // Returns the ans
    return ans;
}
public static void main(String args[])
{

    // Input
    int arr[] = { 1, 23 };
    int N = arr.length;

    // Function call
    System.out.println(ConcatenateArr(arr, N));
}
```

```
[Running] cd "d:\" && javac Q2.java && java Q2
123

[Done] exited with code=0 in 0.492 seconds
```

Explanation B:

- A function is defined to find the output of concatenation. Input is taken in form of an array of two element and an integer which shows length of the array.
- Considering **X** and **Y** as the two integer values to be joined. And also considering the length of the integer **Y** as **L**.
- Then two integers **X** and **Y** can be joined together as following: **$X \times 10^{L} + Y$.**
- Initialize a variable, say **ans** as **0,** to store the resulting value.
- Traverse the array **arr[]** using the variable **i,** and then in each iteration multiply **ans** by **10** to the power of the count of the digit in the integer **arr[i]** and increment **ans** by **arr[i].**
- Finally, after the above step, print the answer obtained in **ans**.

4. Write a function to replace the array element "java" with "JAVA" and display the array elements.

Code:

```
public class Replace
{
    public static void main(String[] args){
      String arr[] = {"Implement","a","java","program"};
      arr[2] = "JAVA";//updating the array element
      for (String str : arr)
{
System.out.println(str);
}
    }
  }
```

OUTPUT:

```
[Running] cd "d:\" && javac Replace.java && java Replace
Implement
a
JAVA
program

[Done] exited with code=0 in 0.518 seconds
```

Explanation:

- An array of 4 element is created and one of the element is being replaced. To do so, we just simply use array indexing and change the element at a particular indexing accordingly.

5. Write a function to insert empty space between array elements and display the output.

CODE 1: In case of integer array (normal spacing)

```
class Space

{

    public static void main(String args[])
    {
        int arr[] = m1();

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

    }

    public static int[] m1()
    {
        // returning  array
        return new int[]{1 ,2 ,3};
    }
}
```

CODE 2: In case of integer array (inserting empty spaces)

```
class Space
{
    // Driver method
    public static void main(String args[])
    {
        int arr[] = m1();

        for (int i = 0; i < arr.length; i++)
        // Comparing with the previous code, extra spacing is
decided by spacing between quotation marks
            System.out.print(arr[i]+"  ");

    }

    public static int[] m1()
    {
        // returning  array
        return new int[]{1,2,3};
    }
}
```

OUTPUT 1:

```
[Running] cd "d:\" && javac Space.java && java Space
123
[Done] exited with code=0 in 0.511 seconds
```

OUTPUT 2:

```
[Running] cd "d:\" && javac Space.java && java Space
1   2   3
[Done] exited with code=0 in 0.493 seconds
```

Explanation:
Returning array using methods. Space is determined by the gap between the quotation marks in System.out.println statement.

6. Receive input from user and replace the initial array with another 4 new words. Display the output.

CODE: For string element array

```
import java.util.Scanner;

public class Replace {
    public static void main(String[] args) {
        printStringArray(replaceInitial(new String[]{"Rice", "Land",
"Water","Fire", "Song", "CSGO", "Halo"}));
    }


        private static String [] replaceInitial(String[] array) {
        //Get user input for replacing with 4 new words
        Scanner ab = new Scanner(System.in);
        //Asking for input 4 times from user
        System.out.println("Enter the first word:");
        array[0] = ab.next();
        System.out.println("Enter the second word:");
        array[1] = ab.next();
        System.out.println("Enter the third word:");
        array[2] = ab.next();
        System.out.println("Enter the fourth word:");
        array[3] = ab.next();


        return array;


    }
    private static void printStringArray(String [] stringArray) {
        for (String s : stringArray){
            System.out.print(s);
        }
    }

}
```

OUTPUT:

```
PS D:\JAVA> java .\Replace.java
Enter the first word:
Hi
Enter the second word:
how
Enter the third word:
are
Enter the fourth word:
you
HihowareyouSongCSGOHalo
```

Explanation:

Here replace replaceInitial method accepts a String array as input.
Scanner is used to get user input.
We have taken input from user 4 different times and placed that input words in array.
We can also add space between array elements using Q5 method.

CODE: For integer element array

```
import java.util.Scanner;

public class Arrayinput
{
public static void main(String[] args)
{
    int[] oldarr = {1, 2, 3, 4};

        for (int element: oldarr) {
            System.out.println(element);
        }
int n;
Scanner sc=new Scanner(System.in);
System.out.print("Enter the number of elements you want to store:
");
//reading the number of elements from the that we want to enter
n=sc.nextInt();

System.out.println("Enter the elements of the array: ");
for(int i=0; i<n; i++)
{
//reading array elements from the user
oldarr[i]=sc.nextInt();
}
System.out.println("New Array elements are: ");
// accessing array elements using the for loop
for (int i=0; i<n; i++)
{
System.out.println(oldarr[i]);
}
}
}
```

OUTPUT:

```
PS D:\JAVA> java .\Arrayinput.java
1
2
3
4
Enter the number of elements you want to store: 4
Enter the new elements of the array:
10
30
50
70
Replaced Array elements are:
10
30
50
70
```

```
PS D:\JAVA> java .\Arrayinput.java
1
2
3
4
5
Enter the number of elements you want to store: 4
Enter the new elements of the array:
10
20
30
40
Replaced Array elements are:
10
20
30
40
```

Explanation:

The `Scanner` class is used to get user input, and it is found in the `java.util` package.

nextInt() method

This Java Scanner class method is used to scan the next token of the input as an int.

7. Define an array which has ‘n’ elements with unique numbers (no duplicates).

a. Take two index locations ‘i’ and ‘j’ (i != j) and perform a subtraction operation

b. Display the output of the operation if it’s a positive number

c. Display an output if its negative numbers

CODE:

```
import java.util.Scanner;
public class Sub
{
public static void main(String[] args)
{
int n;
Scanner sc=new Scanner(System.in);
System.out.print("Enter the number of elements you want to store:
");
//reading the number of elements from the that we want to enter
n=sc.nextInt();
//creates an array in the memory of length 20
int[] array = new int[20];
System.out.println("Enter the elements of the array: ");
for(int i=0; i<n; i++)
{
//reading array elements from the user
array[i]=sc.nextInt();
}
subtract(array);

}

private static void subtract(int[] array) {
    // Scanner to take input from user
    Scanner sub = new Scanner(System.in);

    {
        // Ask user to input index of the element
        System.out.println("Enter index of first Element:" );
        int firstIndex = sub.nextInt();
        System.out.println("Enter index of second Element:");
        int secondIndex = sub.nextInt();
        // Subtract the elements
```

```
        int subtraction = array[firstIndex] - array[secondIndex];
        // Now we will create a loop where subtraction must be
greater than 0, display the output
        if(subtraction > 0){
            System.out.println("The result of subtraction is a
positive number:" + array[firstIndex] + "-" + array[secondIndex] +
"=" + subtraction);
        } else {
            System.out.println("The result of subtraction is a
negative number:" + array[firstIndex] + "-" + array[secondIndex] +
"=" + subtraction);
        }
    }
}
}
```

OUTPUT:

b.)

```
PS D:\JAVA> java .\Sub.java
Enter the number of elements you want to store: 6
Enter the elements of the array:
69
99
21
15
79
35
Enter index of first Element:
4
Enter index of second Element:
5
The result of subtraction is a positive number:79-35=44
```

c.)

```
PS D:\JAVA> java .\Sub.java
Enter the number of elements you want to store: 6
Enter the elements of the array:
69
99
21
15
79
35
Enter index of first Element:
0
Enter index of second Element:
1
The result of subtraction is a negative number:69-99=-30
```

Explanation:

We have created an input array which can take up to 20 elements. If more than 20 elements are given as an input it will throw an exception ArrayoutofBoundException.

After input array is created, we will take input from user of two different indexes on which subtraction is performed.

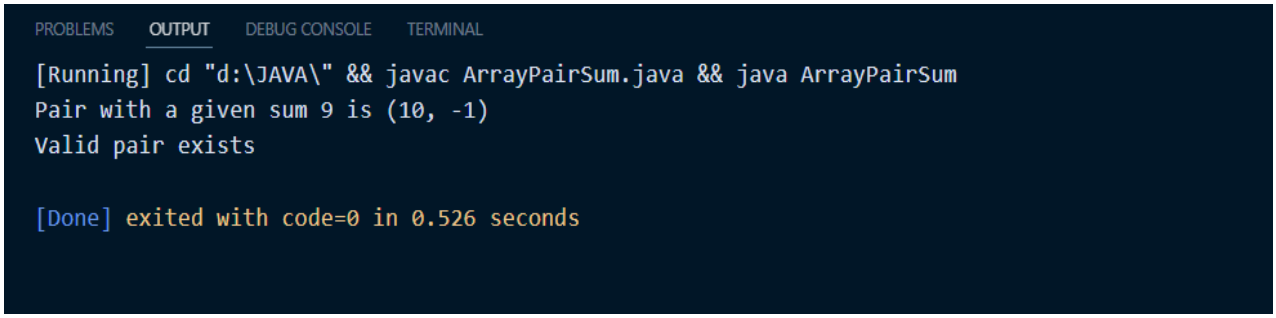
Finally, we will create an if-else condition where output is displayed whether the number after subtraction is positive or negative.

8. Write a java code to find the pair of array elements whose sum will be equal to a given number.

CODE:

```
class ArrayPairSum{


    // Function to find and print pair
    static boolean chkPair(int A[], int size, int x) {
        for (int i = 0; i < (size - 1); i++) {
            for (int j = (i + 1); j < size; j++) {
                if (A[i] + A[j] == x) {
                    System.out.println("Pair with a given sum " + x
+
                                       " is (" + A[i] + ", " + A[j]
+ ")");


                    return true;
                }
            }
        }

        return false;
    }

    public static void main(String [] args) {

        int A[] = {10, -1, 4, -3, 9};
        int x = 9;
        int size = A.length;

        if (chkPair(A, size, x)) {
            System.out.println("Valid pair exists");
        }
        else {
            System.out.println("No valid pair exists for " + x );
        }
    }
  }
```

OUTPUT:



For x = 20;

```
[Running] cd "d:\JAVA\" && javac ArrayPairSum.java && java ArrayPairSum
No valid pair exists for 20

[Done] exited with code=0 in 0.53 seconds
```

Explanation:

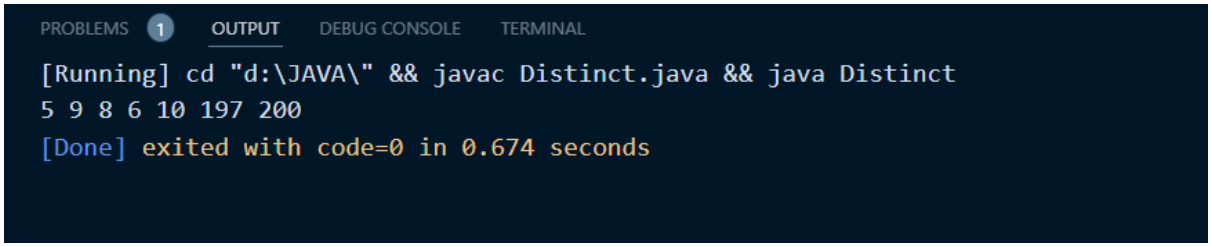
Above code will iterate over the array and will check all the possible sum pair and will print the pair if it exists.

9. Create an array with duplicate elements. Write a code to display unique subarray elements in the same order as it is in the array.

CODE:

```
class Distinct {

    static void printDistinct(int arr[], int n)
    {
        // Pick all elements one by one
        for (int i = 0; i < n; i++)
        {
            // Check if the picked element
            // is already printed
            int j;
            for (j = 0; j < i; j++)
            if (arr[i] == arr[j])
                break;

            // If not printed earlier,
            // then print it
            if (i == j)
            System.out.print( arr[i] + " ");
        }
    }




    public static void main (String[] args)
    {
        int arr[] = {5, 9, 8, 9, 6, 10, 197, 200, 9};
        int n = arr.length;
        printDistinct(arr, n);

    }}
```

OUTPUT:



Explanation:

A **Simple Solution** is to use two nested loops. The outer loop picks an element one by one starting from the leftmost element. The inner loop checks if the element is present on left side of it. If present, then ignores the element, else prints the element. Following is the implementation of the simple algorithm.

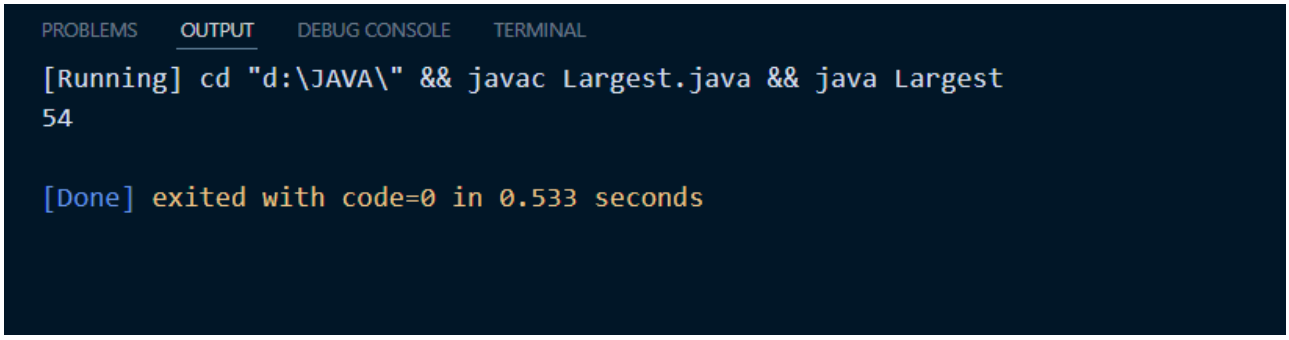
10. Create an array with 15 integer elements (unique elements). Find the subarray of length 3 (consecutive 3 elements), whose sum is largest. Display the subarray.

CODE:

```
import java.util.Arrays;

class Largest
{

    // Function to find the largest sum
    // subarray such that it contains K
    // consecutive elements
public static Integer maximumSum(int[] A, int N, int K)
{
    //Sorting the array so that getting consecutive value gets easy
    Arrays.sort(A);
    // Stores sum of subarray having
    // K consecutive elements
    int curr_sum = 0;

    // Stores the maximum sum among all
    // subarrays of size K having
    // consecutive elements
    int max_sum = Integer.MIN_VALUE;

    // Traverse the array
    for (int i = 0; i < N - K + 1; i++) {

        // Store K elements of one
        // subarray at a time
        int[] dupl_arr = Arrays.copyOfRange(A, i, i + K);

        // Checks if elements in subarray
        // are consecutive or not
        Boolean flag = true;

        // Traverse the k elements
        for (int j = 1; j < K; j++) {

           // Checking if consecutive or not, if not then, break
```

```
            if (dupl_arr[j] - dupl_arr[j - 1]
                != 1) {
                flag = false;
                break;
            }
        }

        // If flag is true update the
        // maximum sum
        if (flag) {
            // Stores the sum of elements
            // of the current subarray
            curr_sum = 0;

            for(int x = 0; x < dupl_arr.length; x++){
                curr_sum += dupl_arr[x];
            }

            // Update the max_sum
            max_sum = Math.max(max_sum,
                        curr_sum);

            // Reset curr_sum
            curr_sum = 0;
        }
    }

    // Return the result
    return max_sum;
}
       public static void main(String args[]) {
        int[] arr = { 11, 21, 13, 9, 8, 18, 10, 15, 1, 3, 2, 17, 14,
19, 71};       // Number of different elements:15
        int K = 3;
        int N = arr.length;
        System.out.println(maximumSum(arr, N, K));
    }

}
```

OUTPUT:



Explanation:

The java.util.Arrays class contains various methods for manipulating arrays such as sorting and searching.
Array.copyofRange - This method creates a copy of elements, within a specified range of the original array.

Boolean Flag is used for condition whether something is true or false. In this case we will check if elements in sub array are consecutive or not

3. Link to upload: (both batch use the same link)

https://amritavishwavidyapeetham-my.sharepoint.com/:f:/g/personal/s_sachinkumar_cb_amrita_edu/Emd4iNFONAFInEnLKioLHEMBm1hjvpn7KWITypy1_Q9_Ew

4. Submission date and time: 12-04-2022, 5PM

THANK YOU