SOURCE CODE

```
import java.util.Scanner;
public class studentmark {
public static void main(String[] args) {
int n, total = 0;
float percentage;
Scanner s = new Scanner(System.in);
System.out.print("Enter no. of subject:");
n = s.nextInt();
int marks[] = new int[n];
System.out.println("Enter marks out of 100:");
for(int i = 0; i < n; i++) {
marks[i] = s.nextInt();
if(marks[i]>100) {
System.out.println("Enter valid marks");
break;   }
else {
total = total + marks[i];   } }
percentage = total / n;
System.out.println("Total mark:"+total);
System.out.println("Percentage:"+percentage);   } }
```

OUTPUT

Lab cycle:2 Date:03/04/2023
Experiment no:1

TOTAL MARKS OBTAINED BY STUDENT

AIM: Write a program which accepts the mark of a student into one dimensional array from the keyboard .Calculate and display total marks obtained by the student.

ALGORITHM:

Step 1: Start

Step 2: Create a class called studentmark.

Step 3: Ask the user to input the no.of subjects(n)

Step 4: Initialize a variable total to 0

Step 5: Create an array called marks of size n

Step 6: Ask the user to enter the marks of each subject and store them in the mark array

Step 7: For each mark entered check if it is greater than 100 if it is display an error message “Enter valid marks” and break out of the loop

Step 8: If the mark is less then or equal to 100 , add it to the total

Step 9: Calculate the percentage by dividing the total by n and store it in the variable percentage and display the total mark and percentage

Step 10 : End

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Arrays;
import java.util.Scanner;
public class StringSorter {
public static void main(String[] args) {
Scanner input = new Scanner(System.in)
System.out.print("Enter the number of strings: ");
int n = input.nextInt()
String[] words = new String[n];
System.out.println("Enter the strings:");
for (int i = 0; i < n; i++) {
words[i] = input.next(); }
Arrays.sort(words);
System.out.println("Sorted strings:");
for (String word : words)  {
System.out.println(word);
} } }
```

OUTPUT

Lab cycle:2 Date:10/04/2023
Experiment no:2

SORT STRINGS

AIM: Write a program to sort strings.

ALGORITHM:

Step 1: Start

Step 2: Create a class called StringSorter

Step 3: Create a scanner object to read user input

Step 4: Ask the user to enter the number of strings they want to sort and store in the variable 'n'

Step 5: Declare an array of strings called words with size n

Step 6: Ask the user to enter each string and strore them in the 'words' array using a loop

Step 7: Sort the 'words' array in alphabetical order using Array.sort(words)

Step 8: Print the sorted strings to the console using a for each loop

Step 9:End

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Arrays;
import java.util.Scanner;
public class CharacterSort {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter a string: ");
String input = scanner.nextLine();
char[] chars = input.toCharArray();
Arrays.sort(chars);
String sorted = new String(chars);
System.out.println("Sorted string: " + sorted);
}
}
```

OUTPUT

Lab cycle:2 Date:10/04/2023
Experiment no:3

SORT CHARACTERS FROM A STRING

AIM: Write a program to sort strings.

ALGORITHM:

Step 1: Start

Step 2: Create a class called CharacterSort

Step 3: Create a scanner object to read user input from the user

Step 4: Prompt the user to enter a string

Step 5: Read the input string using the nextLine method of the scanner object and store it in a string variable called input

Step 6: Convert the input string to a character array using the toCharArray method of the string class and store it in a char array named chars

Step 7: Sort the chars array using the sort method

Step 8: Convert the sorted chars array back to a string using input and store it in a string variable named sorted

Step 9: Print the sorted string to the console

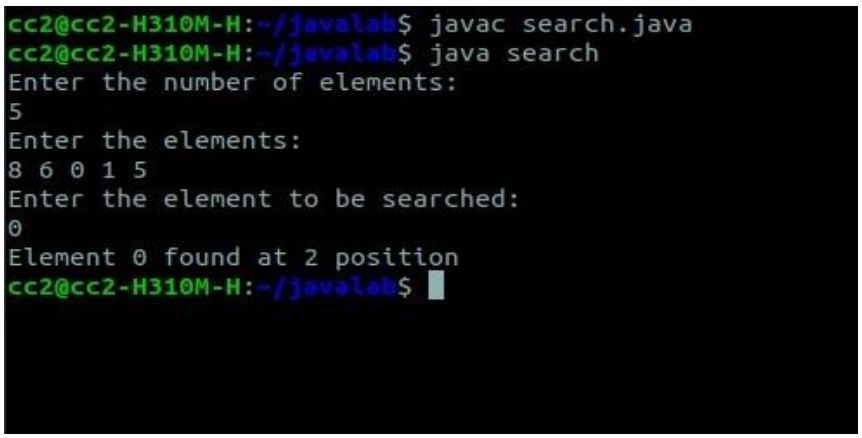
Step 10: End

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.*;
class search {
public static void main(String args[]) {
Scanner sc = new Scanner(System.in);
int i,n,search,flag=0;
System.out.println("Enter the number of elements:") ;
 n = sc.nextInt();
int[] a = new int[n];
System.out.println("Enter the elements:") ;
for(i=0;i<n;i++) {
 a[i] = sc.nextInt();  }
System.out.println("Enter the element to be searched:");
search = sc.nextInt();
for(i=0;i<n;i++) {
 if(a[i]==search) {
 System.out.println("Element "+search+" found at "+i+" position");
 flag=1;
 break; }   }
 if(flag==0) {
 System.out.println("Element "+search+" not found");   } } }
```

OUTPUT



Lab cycle:2 Date:10/04/2023
Experiment no:4

SEARCH AN ELEMENT

AIM: Write a program to search an element in an array.

ALGORITHM:

Step 1: Start

Step 2: Create a class called search

Step 3: Create a scanner object sc to read input from the user

Step 4: Declare and intilize variables i,n,search and flag

Step 5: Prompt the user to enter thr no.of elements and read the input in variable n

Step 6: Create an integer array ‘a’ of size n.

Step 7: Prompt the user to enter the elemets of the array and read the input into the array ‘a’

Step 8: Prompt the user to enter the element to be searched and read the input into ‘search’

Step 9: Use a for loop to iterate over each element of the array

a. Check if the current element is equal to the search element

b. If a match is found then print a message showing that the element is found at current position

c. Set flag to 1 to indicate that the element was found. Break the loop

Step 10: Check the value of flag:

a. If flag is still 0, no match was found, so print a message showing that the element was not found

Step 11:End

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
public class Stringmanipulation{
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a string: ");
String str = scanner.nextLine();
System.out.print("Enter a string to concatenate: ");
String strToConcat = scanner.nextLine();
String concatenatedStr = str.concat(strToConcat);
System.out.println("Concatenated string: " + concatenatedStr);
System.out.print("Enter a character to replace: ");
char oldChar = scanner.nextLine().charAt(0);
System.out.print("Enter a character to replace with: ");
char newChar = scanner.nextLine().charAt(0);
String replacedStr = str.replace(oldChar, newChar);
System.out.println("Replaced string: " + replacedStr);
char[] charArray = str.toCharArray();
System.out.print("Character array: ");
 for (char c : charArray) {
    System.out.print(c + " ");
     }
 System.out.println();
  int length = str.length();
  System.out.println("Length of string: " + length);
  System.out.print("Enter a substring to search for: ");
  String subStr = scanner.nextLine();
   int index = str.indexOf(subStr);
    if (index == -1) {
       System.out.println("Substring not found.");
```

Lab cycle:2 Date:17/04/2023
Experiment no:5

STRING MANIPULATION

AIM: Write a program to perform string manipulation using bulit-in methods of string class andStringBufferclass.

.

ALGORITHM:

Step 1: Start

Step 2: Define a class StringManipulation

Step 3: Prompt the user to enter a string and read the input

Step 4: Prompt the user to enter another string for concatenate with original string and read the input

Step 5 :()method is used for concatenation and then display the result

Step 6: Prompt the user to enter a character to replace it and read the input

Step 7: replace() method is used to replace all the occurrence of old character with new one and display it

Step 8: Convert thre original string into character array using 'toCharArray' method and display each character of array

Step 9: length() method is used to find the length of the string

Step 10:Prompt the user to enter a substring to search for in thr original string

Step 11: indexOf() method used to find the first occurrence of substring in the original string and stores in 'index;

Step 12: If index is -1 then print the substring is not found otherwise print the index

Step 13: toupperCase() method is used to convert the string into upper case and print the result

Step 14: Prompt the user to enter a string for concatenation in StringBuffer usinf append() method and display the result

Step 15:length() method is used to find the length of string in StringBuffer and print the result

Step 16: insert() method insert the substring read from the user at specific position in the StringBuffer

Ste

```
} else {
        System.out.println("Index position of substring: " + index);  }
      String upperCaseStr = str.toUpperCase();
      System.out.println("Uppercase string: " + upperCaseStr);
       StringBuffer stringBuffer = new StringBuffer(str);
       System.out.print("Enter a string to append: ");
       String strToConcat2 = scanner.nextLine();
       stringBuffer.append(strToConcat2);
       System.out.println("Appended string: " + stringBuffer.toString());
       int stringBufferLength = stringBuffer.length();
       System.out.println("Length of StringBuffer: " + stringBufferLength);
       System.out.print("Enter a position to insert substring: ");
       int position = scanner.nextInt();
       scanner.nextLine();
       System.out.print("Enter a substring to insert: ");
       String subStrToInsert = scanner.nextLine();
       stringBuffer.insert(position, subStrToInsert);
       System.out.println("StringBuffer after insertion: " + stringBuffer.toString());
   } }
```

OUTPUT

```
cc2@cc2-H310M-H:~/javalab$ javac Stringmanipulation.java
cc2@cc2-H310M-H:~/javalab$ java Stringmanipulation
Enter a string: hello
Enter a string to concatenate: java
Concatenated string: hellojava
Enter a character to replace: o
Enter a character to replace with: *
Replaced string: hell*
Character array: h e l l o
Length of string: 5
Enter a substring to search for: llo
Index position of substring: 2
Uppercase string: HELLO
Enter a string to append: welcome
Appended string: hellowelcome
Length of StringBuffer: 12
Enter a position to insert substring: 4
Enter a substring to insert: tlow
StringBuffer after insertion: helltlowowelcome
cc2@cc2-H310M-H:~/javalab$
```

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
public class employee  {
   int eNumber;
   String eName;
   double eSalary;
   public void getdetails()  {
      System.out.println("\nEnter the Employee details");
      Scanner sc = new Scanner(System.in);
      System.out.println("Employee number:");
      eNumber=sc.nextInt();
      System.out.println("Name:");
      sc.nextLine();
      eName=sc.nextLine();
      System.out.println("Salary:");
      eSalary=sc.nextDouble();     }
   void display(){
      System.out.println("Empolyee No:"+eNumber);
      System.out.println("Name:"+eName);
      System.out.println("Salary Amount:"+eSalary+"\n");   }
   public static void main(String[] args)    {
      System.out.println("\nEnter the No. of Employee's:");
      Scanner sc1 = new Scanner(System.in);
      int num = sc1.nextInt();
      employee arr[]=new employee[num];
      for(int i =0;i<num;i++){
            arr[i]=new employee();
            arr[i].getdetails();  }
      System.out.println("\nInformations of all the employee's");
      for(int i=0;i<num;i++){
```

Lab cycle:2 Date:17/04/2023
Experiment no:6

EMPLOYEE DETAILS

AIM: Write a program to create a class for employee having attributes eNumber , eName , eSalary. .Read'n' employee information and search for an employee given number using the concept of array of objects.

ALGORITHM:

Step 1: Start

Step 2: Define the 'employee; class with instance variables eNumber,eName and eSalary

Step 3: Define the 'getdetails()' method to get the employee details from the user

Step 4: Define the 'display()' method to display the employee details

Step 5: In the main method:

a. Prompt the user to enter the no.of employees

b. Read the input and store it in the variable num

c. Create an array of employee objects with the size equal to num

d. Use a for loop to iterate num times:

- Create a new employee object
- Call the getdetails() method for employee to get their details
- Store the employee object in the array

e. Print the message to display the information of all employees

f. Use a for loop to iterate through the array

- Call the display() methid for each employee to display their details

g. Prompt the user to enter the employee number to get the details of a specific employee

h. Read the input and store it in the variable num2

i. Use a for loop to iterate through the array:

- Check if the employee number of the current employee matches num2
- If there is a match,print the message to display the employee details and call the dispay method fot that employee.

Step 6: Stop

```
        arr[i].display();    }
    boolean state = false;
    System.out.println("\nEnter the Employee Number to get details of a employee:");
    int num2= sc1.nextInt();
    for(int i=0;i<num;i++){
       if(arr[i].eNumber==num2){
      System.out.println("\nEmployee details");
       arr[i].display(); } } } }
```

OUTPUT

```
cc2@cc2-H310M-H:~/javalab$ javac employee.java
cc2@cc2-H310M-H:~/javalab$ java employee

Enter the No. of Employee's:
3

Enter the Employee details
Employee number:
1
Name:
Arya
Salary:
45000

Enter the Employee details
Employee number:
2
Name:
Varsha
Salary:
50000

Enter the Employee details
Employee number:
3
Name:
Anju
Salary:
40000
```

```
Enter the Employee Number to get details of a employee:
2

Employee details
Empolyee No:2
Name:Varsha
Salary Amount:50000.0

cc2@cc2-H310M-H:~/javalab$
```

RESULT: The program has been executed successfully and output obtained.