Experiment 1

Second smallest element in the array

Date Of Submission: 26-08-2020

Aim: Write a Java program to find the second smallest element in an array

Concepts Used: Class, Arrays

Algorithm:

1. Step 1: Start
2. Step 2: pos = 0 //Position of the smallest element
3. Step 3: smallest = arr[pos] //Assume the first element is the smallest
4. Step 4: for i from 1 to arraySize-1 do // Find the smallest element
5. Step 1: if arr[pos] > arr[i] then
6. Step 1: pos = i
7. Step 2: endif
8. Step 5: endFor
9. Step 6: if pos == 0 then
10. Step 1: secondSmallest = 1 // Assume that the second smallest number is the second number
11. Step 7: else
12. Step 1: secondSmallest =0;
13. Step 8: endif
14. Step 10: for i from 0 to arraySize-1 do
15. Step 1: if i == pos then
16. Step 1: continue
17. Step 2: endif
18. Step 3: if arr[i]<arr[secondSmallest] then
19. Step 1: secondSmallest = i
20. Step 4: endif
21. Step 11; Endfor
22. step 12: Stop

Result: The program is successfully compiled and the required output is obtained.

Program Code:

```
/*  Java program to find the second smallest element in an array
 *
*/
class Program1{
```

```
    public static void main(String args[]){
        int[] arr = {-34,2,9,34,12,9,-23,1,4,9,0};
        int arrSize=11;
        int smallest,secondSmallest;
        int i,pos=0;

        smallest=arr[0];

        for(i=1;i<11;i++){
            if(arr[i]<smallest){
                pos=i;
            }
        }

        if(pos!=0){
            secondSmallest=arr[0];
        }
        else{
            secondSmallest=arr[arrSize-1];
        }

        for(i=0;i<11;i++){
            if(i==pos)continue;
                if(arr[i]<secondSmallest){
                    secondSmallest=arr[i];
            }
        }

        System.out.println("Second Smallest element is "+secondSmallest);
    }
}
```

Sample Input
-34,2,9,34,12,9,-23,1,4,9,0

Sample Output:
Second Smallest element is -23

Experiment 2

Program to check whether the given number is prime

Date Of Submission: 26-08-2020

Aim: Write a Java program to check whether the given number is prime or not

Concepts Used: Class

Algorithm:

1. Step 1: Start
2. Step 2: read n // the number to be checked
3. Step 3: flag = 0
4. Step 4; for i from 2 to n/2 do
5. Step 1: if n%i == 0 then
6. Step 1: flag = 1
7. Step 2: break
8. Step 2: endif
9. Step 5: endFor
10. Step 6: if flag == 1 then
11. Step 1: print “The number is prime”
12. Step 7: else:
13. Step 1: print “The number is not prime”
14. Step 8: endif
15. Step 9: Stop

Result: The program is successfully compiled and the required output is obtained.

Program Code:

```
/*
 *Program to check whether a given number is prime or not
 *
 */

class Program2{
    public static void main(String[] args){
        int n,i;
        boolean flag=true;

        n=31; //The number to be checked

        for(i=2;i<n/2;i++){
```

```
            if(n%i==0){
                flag=false;
            }

        }

        if(flag)
            System.out.println("The number "+n+" is prime");
        else
            System.out.println("The number "+n+" is not prime");
    }
}
```

Sample input

31

Sample output:

The number 31 is prime

Experiment 3

Multiplication of Matrices

Date Of Submission: 26-08-2020

Aim: Write a Java program to multiply two given matrices

Concepts Used: 2-D Array, Class

Algorithm:

1. Step 1: Start
2. Step 2: if A.colums == B.rows then
3. Step 1: C.colums = B.columns
4. Step 2: C.rows = A.rows
5. Step 3: for i from 0 to A.rows-1 do
6. Step 1: for j from 0 to B.columns-1 do
7. Step 1: C.array[i][j] = 0
8. Step 2: for k from 0 to B.rows do
9. Step 1: C[i][j] += A[i][k]*B[k][j]
10. Step 3: endfor
11. Step 2: endFor
12. Step 4: endfor
13. Step 3:else
14. Step 1: Print “Matrices can’t be multiplied”
15. Step 4: endif
16. Step 5: Stop

Result: The program is successfully compiled and the required output is obtained.

Program Code

```
/*  Program to multiply two given matrices
 */

class Program3{
    public static void main(String[] args){
        int[][] C;
        int a_rows,a_columns,b_rows,b_columns,c_rows,c_columns;
        int i,j,k;

        a_rows=3;
        a_columns=2;
        int A[][] = {{1,2},{4,5},{9,16}};

        b_rows = 2; b_columns=4;
```

```
        int B[][] = {{3,4,5,6},{4,3,1,0}};

        if(a_columns==b_rows){
            c_rows = a_rows;
            c_columns = b_columns;
            C = new int[a_rows][b_columns];

            for(i=0;i<a_rows;i++){
                for(j=0;j<b_columns;j++){
                    C[i][j]=0;
                    for(k=0;k<a_columns;k++){
                        C[i][j]+=(A[i][k]*B[k][j]);
                    }
                }
            }

            System.out.println("Solution Matrix is : ");
            for(i=0;i<a_rows;i++){
                for(j=0;j<b_columns;j++){
                    System.out.print(C[i][j]+" ");
                }
                System.out.println(" ");
            }

        }

        else{
            System.out.println("Matrix cant be multiplied");
        }
    }
}
```

Sample input:

```
Matrix A = 1,2
           4,5
           9,16
Matrix B =  3,4,5,6
            4,3,1,0
```

Sample output:

```
Solution Marix is:
11 10 7 6
32 31 25 24
91 84 61 54
```