AMRITA SCHOOL OF COMPUTING CHENNAI

AMRITA VISHWA VIDYAPEETHAM

22AIE113 ELEMENTS OF COMPUTING SYSTEMS-2

ASSESSMENT - 3

DONE BY:

S.PRAVEEN KUMAR

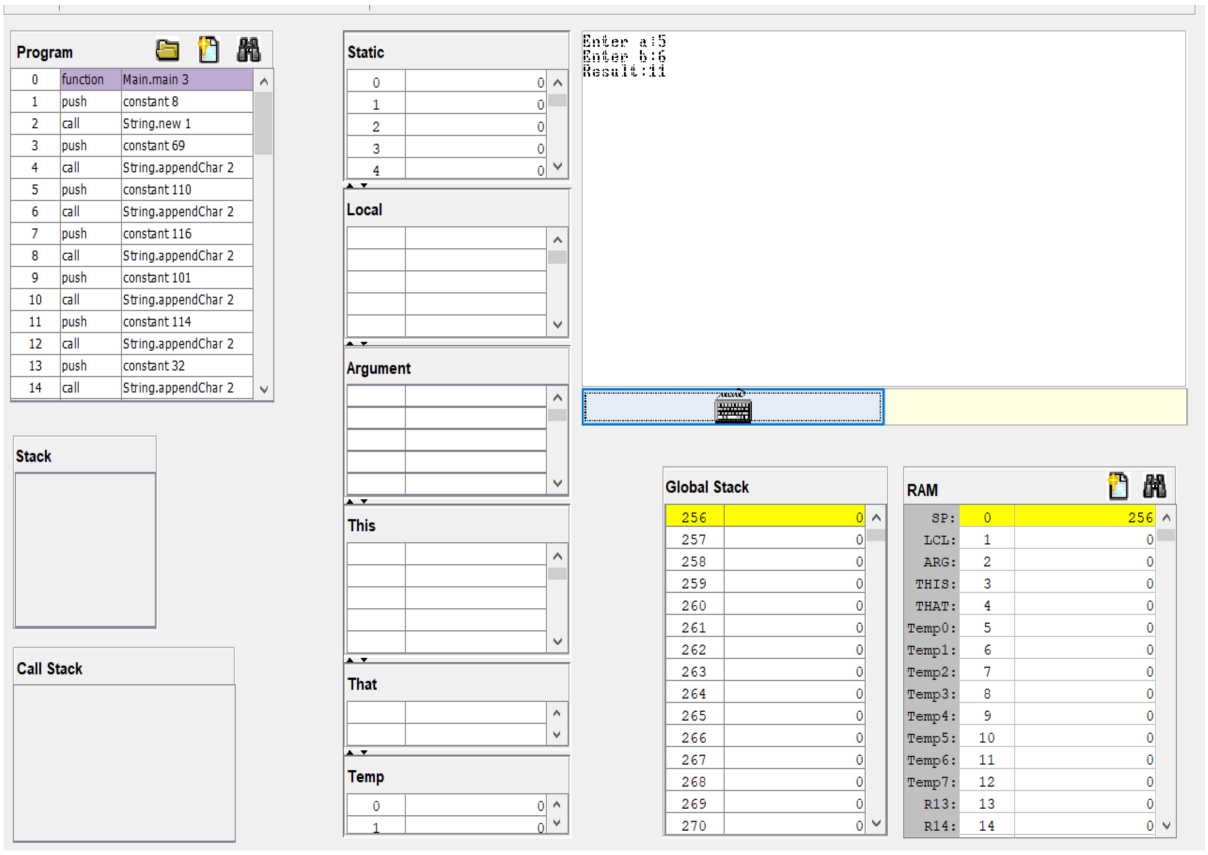
CH.EN.U4AIE22048

1.Write a program in JACK to evaluate c=a+b

CODE:

```
class Main{
    function void  main() {
        var int a;
        var int b;
        var int c;
        let a = Keyboard.readInt("Enter a:");
        let b = Keyboard.readInt("Enter b:");
        let c = a + b;
        do Output.printInt(c);
        return;
    }
}
```

OUTPUT:



Program

0	function	Main.main 3
1	push	constant 8
2	call	String.new 1
3	push	constant 69
4	call	String.appendChar 2
5	push	constant 110
6	call	String.appendChar 2
7	push	constant 116
8	call	String.appendChar 2
9	push	constant 101
10	call	String.appendChar 2
11	push	constant 114
12	call	String.appendChar 2
13	push	constant 32
14	call	String.appendChar 2

Static

0	0
1	0
2	0
3	0
4	0

Temp

0	0
1	0

```
Enter a:5
Enter b:6
Result:11
```

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

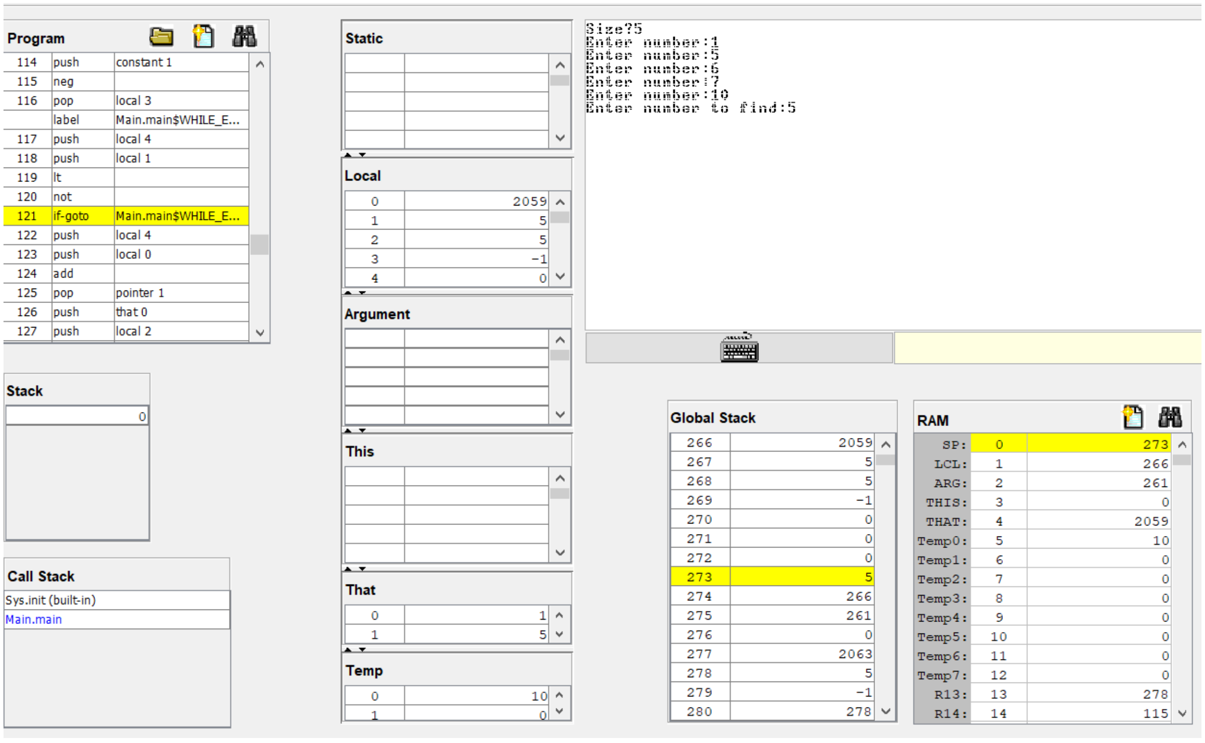
SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

2.Write a Program in JACK to perform linear search.

CODE:

```
class Main{
    function void main(){
        var Array arr;
        var int len;
        var int key;
        var int i;
        var int b ;
        let len = Keyboard.readInt("Enter the len : ");
        let arr = Array.new(len);
        let i = 0;
        while(~(i=len)){
            let arr[i] = Keyboard.readInt("Enter element : ");
            let i = i + 1;
        }

        let key = Keyboard.readInt("Enter the element to search : ");
        let i = 0;
        let b = 0;
        while(~(i = len) & (b = 0)){
         if(arr[i] = key){
            let b = 1;
         }
         let i = i + 1;
        }
        if(b = 0){
            do Output.printString("not found");
        }else{
            do Output.printString("found in index  ");
            do Output.printInt(i);
        }
        return;
    }
}
```

OUTPUT:



Program

114	push	constant 1
115	neg	
116	pop	local 3
	label	Main.main$WHILE_E...
117	push	local 4
118	push	local 1
119	lt	
120	not	
121	if-goto	Main.main$WHILE_E...
122	push	local 4
123	push	local 0
124	add	
125	pop	pointer 1
126	push	that 0
127	push	local 2

Stack

0

Call Stack

Sys.init (built-in)
Main.main

Static

Local

0	2059
1	5
2	5
3	-1
4	0

Argument

This

That

0	1
1	5

Temp

0	10
1	0

```
Size?5
Enter number:1
Enter number:5
Enter number:6
Enter number:7
Enter number:10
Enter number to find:5
```

Global Stack

266	2059
267	5
268	5
269	-1
270	0
271	0
272	0
273	5
274	266
275	261
276	0
277	2063
278	5
279	-1
280	278

RAM

SP:	0	273
LCL:	1	266
ARG:	2	261
THIS:	3	0
THAT:	4	2059
Temp0:	5	10
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	278
R14:	14	115

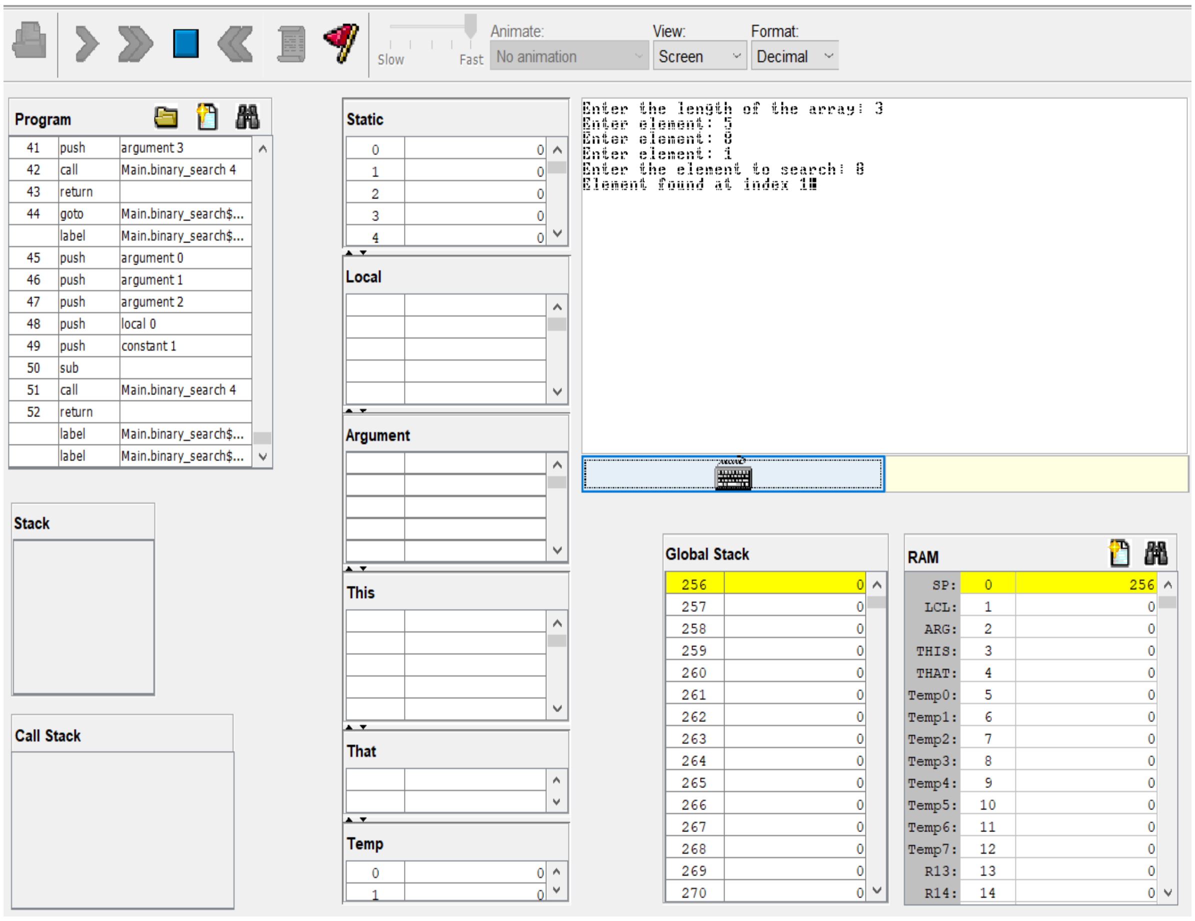
S.Praveen Kumar ch.en.u4aie22048

3.Write a Program in JACK to perform binary search.

CODE:

```
class Main {
    function void main() {
        var Array arr;
        var int len;
        var int key;
        var int i;
        var int result;
        let len = Keyboard.readInt("Enter the length of the array: ");
        let arr = Array.new(len);
        let i = 0;
        while (i < len) {
            let arr[i] = Keyboard.readInt("Enter element: ");
            let i = i + 1;
        }
        let key = Keyboard.readInt("Enter the element to search: ");

        let result = Main.binary_search(arr, key, 0,len-1);
        if (result = -1) {
            do Output.printString("Element not found.\n");
        } else {
            do Output.printString("Element found at index ");
            do Output.printInt(result);
            do Output.printString("\n");
        }
        return;
    }
    function int binary_search(Array arr, int key,int low,int high){
        var int mid;
        if(low > high){
            return -1;
        }
        let mid = (low + high) / 2;
        if (arr[mid] = key) {
            return mid;
        } else {
            if (arr[mid] < key) {
            return Main.binary_search(arr, key, mid + 1, high);
            } else {
            return Main.binary_search(arr, key, low, mid - 1);
            }
        }
    }
}
```

OUTPUT:



Animate: No animation View: Screen Format: Decimal

```
Enter the length of the array: 3
Enter element: 5
Enter element: 8
Enter element: 1
Enter the element to search: 8
Element found at index 1
```

Program

41	push	argument 3
42	call	Main.binary_search 4
43	return	
44	goto	Main.binary_search$...
	label	Main.binary_search$...
45	push	argument 0
46	push	argument 1
47	push	argument 2
48	push	local 0
49	push	constant 1
50	sub	
51	call	Main.binary_search 4
52	return	
	label	Main.binary_search$...
	label	Main.binary_search$...

Static

0	0
1	0
2	0
3	0
4	0

Temp

0	0
1	0

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

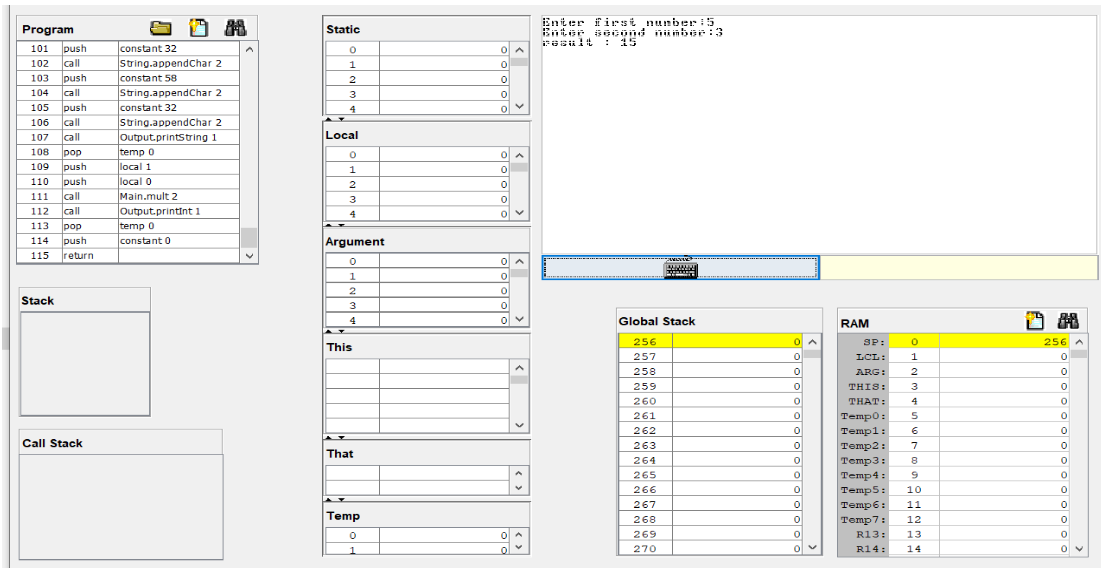
SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

4.Write a recursive program for multiplication.

CODE:

```
class Main {
    function void main () {
        var int a;
        var int b;
        var int c;

        let a  = Keyboard.readInt("Enter a :");
        let b  = Keyboard.readInt("Enter b :");
        let c  = Main.mul(a,b);

        do Output.printString("The product of both = ");
        do Output.printInt(c);

        return;
    }

    function int mul(int a, int b){
        if(b=0){
            return 0;
        }else{
            return a + Main.mul(a,b-1);
        }
    }
}
```

OUTPUT:



Program

```
101  push    constant 32
102  call    String.appendChar 2
103  push    constant 58
104  call    String.appendChar 2
105  push    constant 32
106  call    String.appendChar 2
107  call    Output.printString 1
108  pop     temp 0
109  push    local 1
110  push    local 0
111  call    Main.mult 2
112  call    Output.printInt 1
113  pop     temp 0
114  push    constant 0
115  return
```

```
Enter first number:5
Enter second number:3
result : 15
```

Static

0	0
1	0
2	0
3	0
4	0

Local

0	0
1	0
2	0
3	0
4	0

Argument

0	0
1	0
2	0
3	0
4	0

Temp

0	0
1	0

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

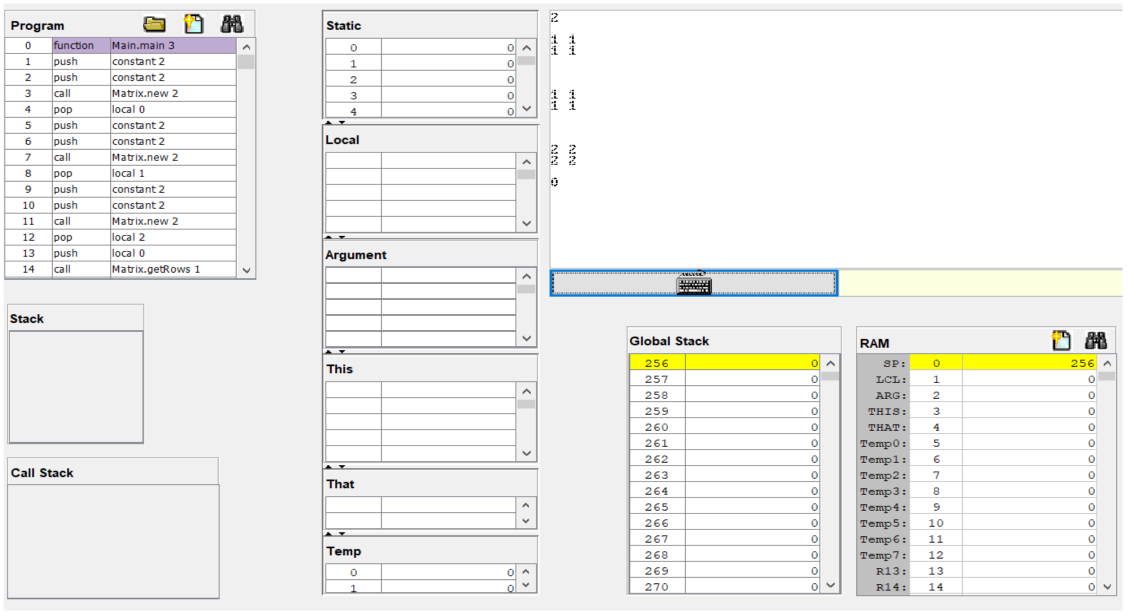
S.Praveen Kumar ch.en.u4aie22048

5.Write a Program in JACK to perform Matrix Addition.

CODE:

```
class Main {

    function void main() {
        var Matrix a;
        var Matrix b;
        var Matrix c;
        let a = Matrix.new(2, 2);
        let b = Matrix.new(2, 2);
        let c = Matrix.new(2, 2);
        do Output.printInt(a.getRows());
        do a.ones();
        do a.display();
        do b.ones();
        do b.display();
        do a.add(b);
        do a.display();
        do Output.printInt(c.index(0));
        return;
    }
}
```

OUTPUT:



Program

0	function	Main.main 3
1	push	constant 2
2	push	constant 2
3	call	Matrix.new 2
4	pop	local 0
5	push	constant 2
6	push	constant 2
7	call	Matrix.new 2
8	pop	local 1
9	push	constant 2
10	push	constant 2
11	call	Matrix.new 2
12	pop	local 2
13	push	local 0
14	call	Matrix.getRows 1

Static

0	0
1	0
2	0
3	0
4	0

Temp

0	0
1	0

```
2

1 1
1 1


1 1
1 1


2 2
2 2

0
```

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

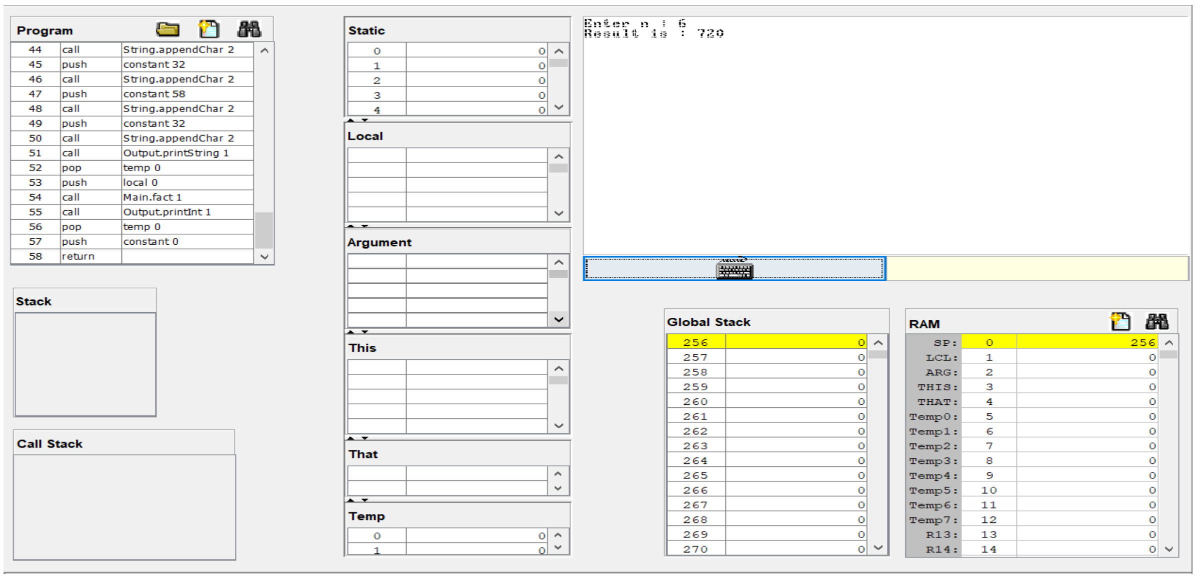
SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

6.Write a Program in JACK to perform factorial of a number using recursion.

CODE:

```
class Main {
    function void main() {
        var int a;
        var int res;

        let a = Keyboard.readInt("Enter the n-th Factorial number: ");
        let res = Main.factorial(a);
        do Output.printString("The n-th Factorial number is ");
        do Output.printInt(res);
        return;
    }

    function int factorial(int n) {
        if (n = 1) {
            return 1;
        }
        return n * Main.factorial(n - 1);
    }
}
```

OUTPUT:



```
Enter n : 6
Result is : 720
```

Program

44	call	String.appendChar 2
45	push	constant 32
46	call	String.appendChar 2
47	push	constant 58
48	call	String.appendChar 2
49	push	constant 32
50	call	String.appendChar 2
51	call	Output.printString 1
52	pop	temp 0
53	push	local 0
54	call	Main.fact 1
55	call	Output.printInt 1
56	pop	temp 0
57	push	constant 0
58	return	

Static

0	0
1	0
2	0
3	0
4	0

Temp

0	0
1	0

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

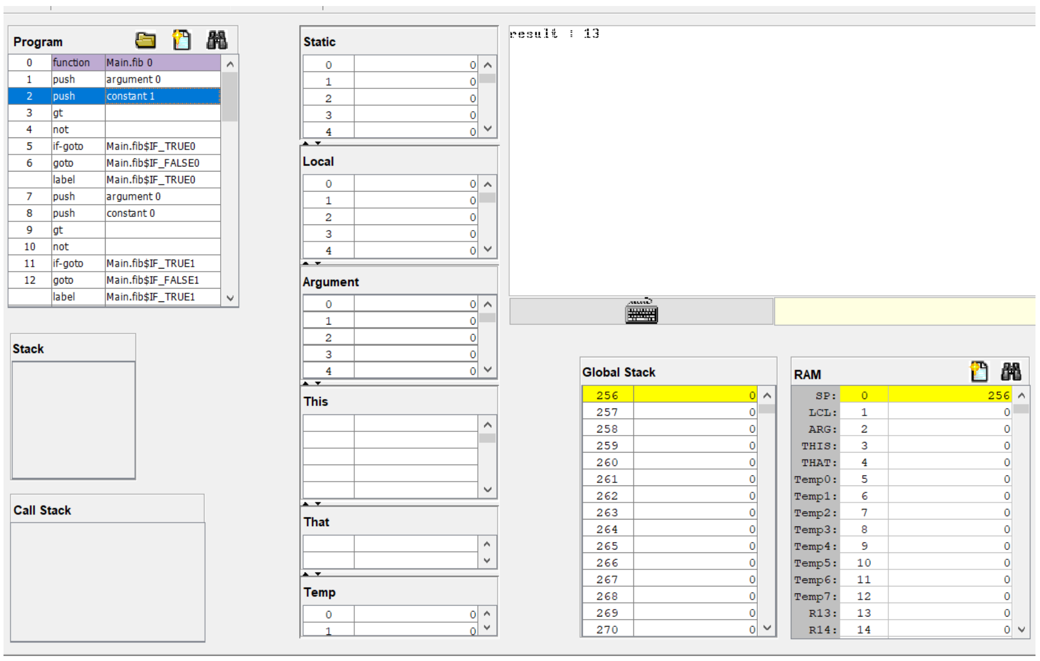
SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

7.Write a Program in JACK to generate Fibonacci series using recursion.

CODE:

```
class Main {
    function void main() {
        var int a;
        var int res;

        let a = Keyboard.readInt("Enter the n-th Fibonacci number: ");
        let res = Main.fib(a);
        do Output.printString("The n-th Fibonacci number is ");
        do Output.printInt(res);
        return;
    }

    function int fib(int n) {
        if (n = 0) {
            return 0;
        }
        if (n = 1) {
            return 1;
        }
        return Main.fib(n - 1) + Main.fib(n - 2);
    }
}
```

OUTPUT:



```
result : 13
```

Program

0	function	Main.fib 0
1	push	argument 0
2	push	constant 1
3	gt	
4	not	
5	if-goto	Main.fib$IF_TRUE0
6	goto	Main.fib$IF_FALSE0
	label	Main.fib$IF_TRUE0
7	push	argument 0
8	push	constant 0
9	gt	
10	not	
11	if-goto	Main.fib$IF_TRUE1
12	goto	Main.fib$IF_FALSE1
	label	Main.fib$IF_TRUE1

Static

0	0
1	0
2	0
3	0
4	0

Local

0	0
1	0
2	0
3	0
4	0

Argument

0	0
1	0
2	0
3	0
4	0

Temp

0	0
1	0

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

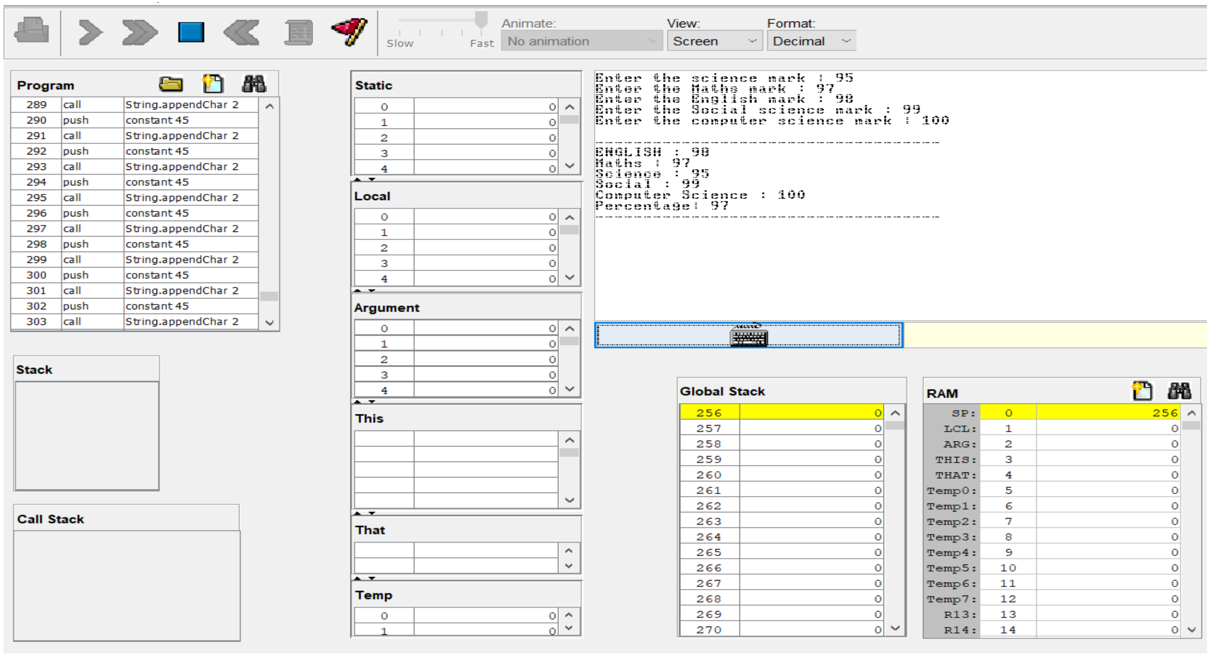
S.Praveen Kumar ch.en.u4aie22048

8.Write a program in JACK which generates report card for "n" students.

CODE:

```
class Main {

    function void main() {
        var Student st;
        var int sci;
        var int sc;
        var int eng;
        var int maths;
        var int cs;

        let sci = Keyboard.readInt("Enter the science mark : ");
        let maths = Keyboard.readInt("Enter the Maths mark : ");
        let eng = Keyboard.readInt("Enter the English mark : ");
        let sc = Keyboard.readInt("Enter the Social science mark : ");
        let cs = Keyboard.readInt("Enter the computer science mark : ");

        let st = Student.new(maths, sci, eng, cs,sc);
        do st.display();

        return;
    }
}
```

OUTPUT:



Animate: No animation
View: Screen
Format: Decimal

Program

289	call	String.appendChar 2
290	push	constant 45
291	call	String.appendChar 2
292	push	constant 45
293	call	String.appendChar 2
294	push	constant 45
295	call	String.appendChar 2
296	push	constant 45
297	call	String.appendChar 2
298	push	constant 45
299	call	String.appendChar 2
300	push	constant 45
301	call	String.appendChar 2
302	push	constant 45
303	call	String.appendChar 2

```
Enter the science mark : 95
Enter the Maths mark : 97
Enter the English mark : 98
Enter the Social science mark : 99
Enter the computer science mark : 100

-----------------------------------------
ENGLISH : 98
Maths : 97
Science : 95
Social : 99
Computer Science : 100
Percentage: 97
-----------------------------------------
```

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

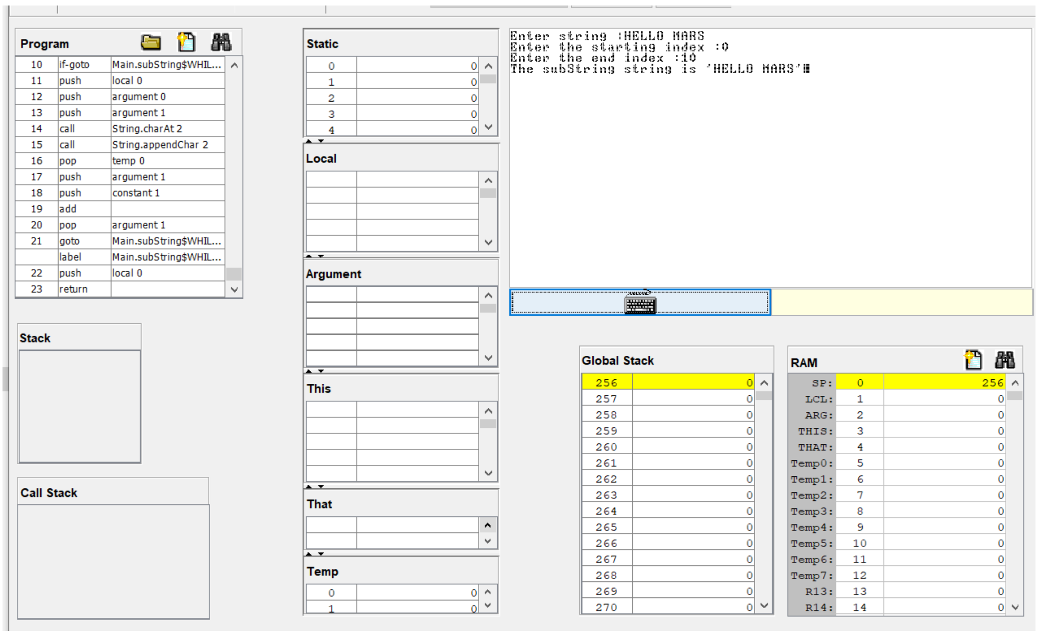
SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

9.Write a jack program which extracts substring from a given sentence/string.

CODE:

```
class Main {
    function void main () {
        var String string;
        var int startingIndex;
        var int endingIndex;
        var String subString;
        let string = Keyboard.readLine("Enter string :");
        let startingIndex = Keyboard.readInt("Enter the starting index :");
        let endingIndex = Keyboard.readInt("Enter the end index :");
        let subString = Main.subString(string, startingIndex, endingIndex);
        do Output.printString("The subString string is '");
        do Output.printString(subString);
        do Output.printString("'\n");
        return;
    }
    function String subString (String s, int start, int end ) {
        var String result;
        let result = String.new(end - start);
        while(end>start){
            do result.appendChar(s.charAt(start));
            let start = start + 1;
        }
        return result;
    }
}
```

OUTPUT:



Program

10	if-goto	Main.subString$WHIL...
11	push	local 0
12	push	argument 0
13	push	argument 1
14	call	String.charAt 2
15	call	String.appendChar 2
16	pop	temp 0
17	push	argument 1
18	push	constant 1
19	add	
20	pop	argument 1
21	goto	Main.subString$WHIL...
	label	Main.subString$WHIL...
22	push	local 0
23	return	

```
Enter string :HELLO MARS
Enter the starting index :0
Enter the end index :10
The subString string is 'HELLO MARS'
```

Static

0	0
1	0
2	0
3	0
4	0

Temp

0	0
1	0

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

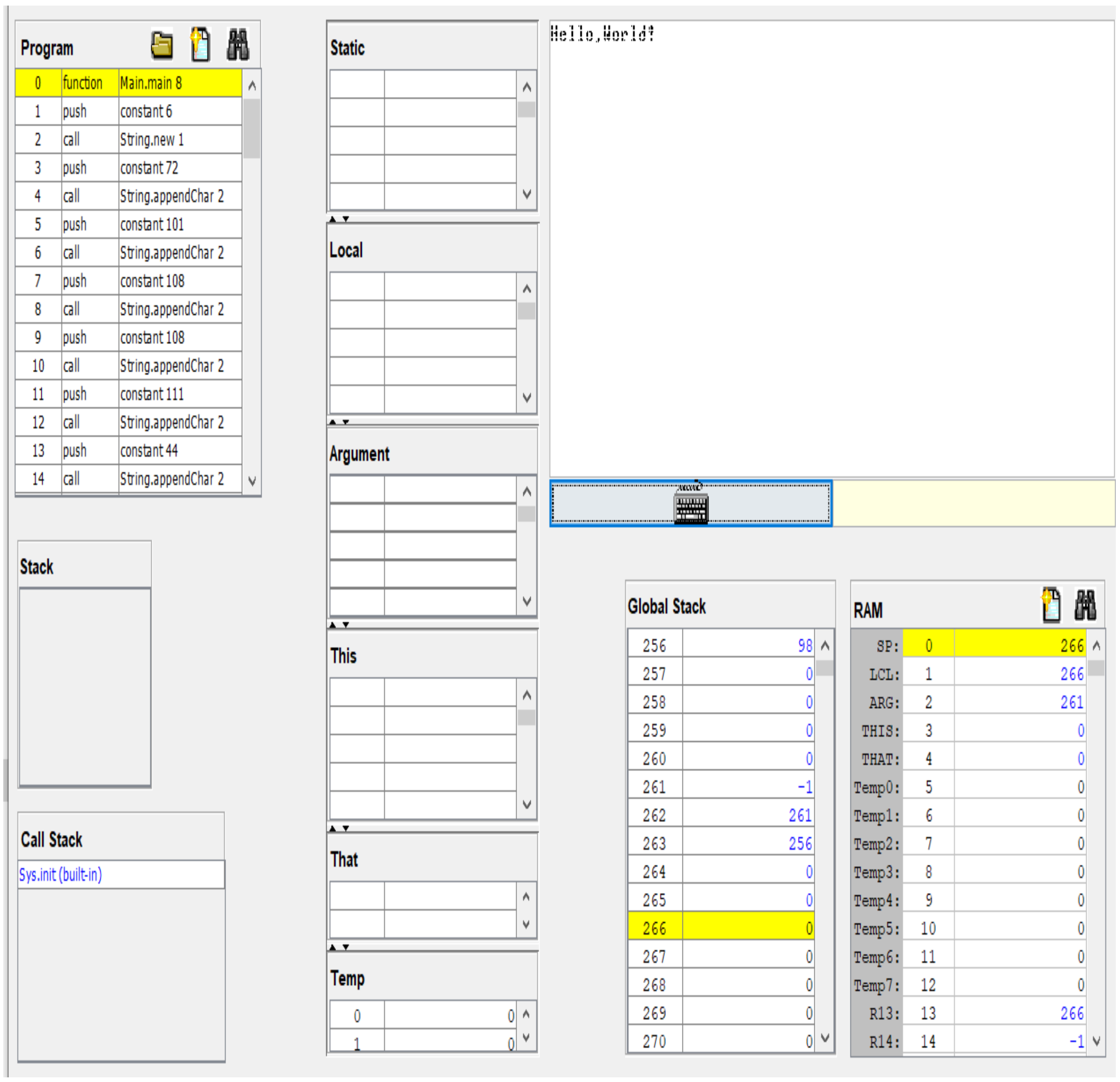
S.Praveen Kumar ch.en.u4aie22048

10.Write a jack program which concatenates 2 strings.

CODE:

```
class Main {
    function void main () {
        var String string1;
        var String string2;
        var String concatedString;

        let string1 = Keyboard.readLine("Enter string1 :");
        let string2 = Keyboard.readLine("Enter string2 :");

        let concatedString = Main.conact(string1, string2);
        do Output.printString("The concated string is '");
        do Output.printString(concatedString);
        do Output.printString("'\n");
        return;
    }

    function String conact (String a, String b) {
        var int len_b;
        var int len_a;
        var int i;
        var String result;

        let len_b = b.length();
        let len_a = a.length();
        let result = String.new(len_a+len_b);

        let  i = 0;
        while(i<len_a){
            do result.appendChar(a.charAt(i));
            let i = i + 1;
        }
        let  i = 0;
        while(i<len_b){
            do result.appendChar(b.charAt(i));
            let i = i + 1;
        }

        return result;
    }
}
```

OUTPUT:



Program

0	function	Main.main 8
1	push	constant 6
2	call	String.new 1
3	push	constant 72
4	call	String.appendChar 2
5	push	constant 101
6	call	String.appendChar 2
7	push	constant 108
8	call	String.appendChar 2
9	push	constant 108
10	call	String.appendChar 2
11	push	constant 111
12	call	String.appendChar 2
13	push	constant 44
14	call	String.appendChar 2

Stack

Call Stack

Sys.init (built-in)

Static

Local

Argument

This

That

Temp

0	0
1	0

```
Hello,World!
```

Global Stack

256	98
257	0
258	0
259	0
260	0
261	-1
262	261
263	256
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

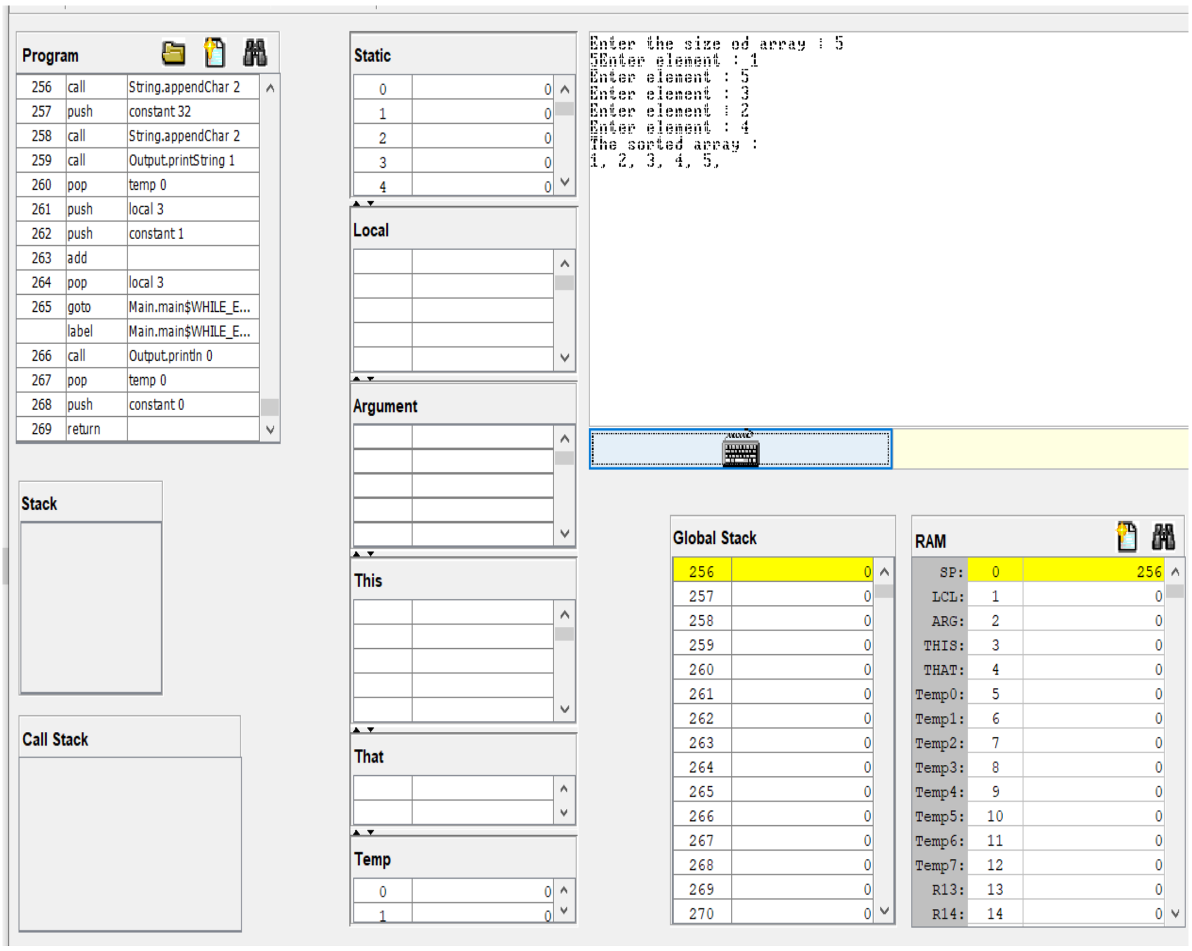
SP:	0	266
LCL:	1	266
ARG:	2	261
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	266
R14:	14	-1

11.Write a Program in JACK to perform MERGE sort using recursion.

CODE:

```
class Main {
    function void main () {
        var int len;
        var Array arr;
        var int temp;
        var int i;
        var int j;
        let i = 0;
        let j = 0;
        let len = Keyboard.readInt("Enter the size od array : ");
        let arr = Array.new(len);
        do Output.printInt(len);
        while(~(i=len)){
            let arr[i] = Keyboard.readInt("Enter element : ");
            let i = i + 1;
        }
        let i = 0;
        while(i<(len-1)){
            let j = 0;
            while(j<(len - i - 1)){
                if(arr[j] > arr[j+1]){
                    let temp = arr[j];
                    let arr[j] = arr[j+1];
                    let arr[j+1] = temp;
                }
                let j = j + 1;
            }
            let i = i + 1;
        }
        let i = 0;
        do Output.printString("The sorted array : ");
        do Output.println();
        while(i<len){
            do Output.printInt(arr[i]);
            do Output.printString(", ");
            let i = i + 1;
        }
        do Output.println();
        return;
    }
}
```

OUTPUT:



Program

256	call	String.appendChar 2
257	push	constant 32
258	call	String.appendChar 2
259	call	Output.printString 1
260	pop	temp 0
261	push	local 3
262	push	constant 1
263	add	
264	pop	local 3
265	goto	Main.main$WHILE_E...
	label	Main.main$WHILE_E...
266	call	Output.println 0
267	pop	temp 0
268	push	constant 0
269	return	

Static

0	0
1	0
2	0
3	0
4	0

Temp

0	0
1	0

```
Enter the size od array : 5
5Enter element : 1
Enter element : 5
Enter element : 3
Enter element : 2
Enter element : 4
The sorted array :
1, 2, 3, 4, 5,
```

Global Stack

256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM

SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0