

ELEMENTS OF COMPUTING SYSTEMS - 2

TEAM MEMBERS

BATCH-A TEAM-7	
GAJULA SRI VATSANKA	CB.EN.U4AIE.21010
GUNNAM HIMAMSH	CB.EN.U4AIE.21014
M.PRASANNA TEJA	CB.EN.U4AIE.21035
VIKHYAT BANSAL	CB.EN.U4AIE.21076

1.

Write a Jack program to execute the following equation

If $n \leq 10$, $y = \sum n$

Else $y = \sum n^2$

CODE LOGIC

- ✗ We will take a variable n whose value we will take as input from user.
- ✗ If n is less than or equal to 10 then it will give summation of n from 0 to 10
- ✗ If n is more than 10 then output will be given as summation of n^2 from 11 to till number.

NOTE:

Value of n will be calculated till when $n = 45$ as soon as value of n is taken as $n = 46$. It will start the count in negative as 16 bit number integer range is from -32768 to +32767.

```

class Main {
    function void main() {
        var int n;
        var int y;
        var int i;
        //var int sum;

        Let n = Keyboard.readInt("Enter the value of n:");
        do Output.println();

        Let y = 0;
        if ((n<10) | (n=10)) {

            Let i = 0;
            Let y = 0;

            while (i>-1 & i<n) {
                Let i = i+1;
                Let y = y+i;
            }

            do Output.printString("The value of y is:");
            do Output.printInt(y);
            do Output.println();
        }
    }
}

```

JACK CODE

```

else {
    Let i = 0;
    Let y = 0;

    while (i>-1 & i<n) {
        Let i = i+1;
        Let y = y+(i*i);
    }

    do Output.printString("The value of y is:");
    do Output.printInt(y);

}
return;
}

```



JACK COMPILATION



The image shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with folders like HelloWorld, MinMax, OS, Prime, Summation, and Timer. The Summation folder is expanded, showing Main.jack and Main.vm. The code editor displays the content of Main.jack, which is a Jack class with a main function. The terminal shows the command to compile the program using JackCompiler.bat.

```
Summation > Main.jack

1  class Main {
2      function void main() {
3          var int n;
4          var int y;
5          var int i;
6          //var int sum;
7
8          let n = Keyboard.readInt("Enter the value of n:");
9          do Output.println();
10
11         let y = 0;
12         if ((n<10) | (n=10)) {
13
14
15             let i = 0;
16             let y = 0;
17
18             while (i>-1 & i<n) {
19                 let i = i+1;
20                 let y = y+i;
21             }
22         }
23     }
24 }
```

PROBLEMS COMMENTS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> .\JackCompiler.bat .\Summation\Main.jack
Compiling "C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools\Summation\Main.jack"
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> 
```

OUTPUT

File View Run Help



Program

185	call	String.appendChar 2
186	push	constant 105
187	call	String.appendChar 2
188	push	constant 115
189	call	String.appendChar 2
190	push	constant 58
191	call	String.appendChar 2
192	call	Output.printString 1
193	pop	temp 0
194	push	local 1
195	call	Output.printInt 1
196	pop	temp 0
	label	Main.main\$IF_END0
197	push	constant 0
198	return	

Stack

--

Call Stack

--

Static

0	0
1	0
2	0
3	0
4	0

Local

Argument

This

That

--	--

Temp

0	0
1	0

Enter the value of n:5
The value of y is:15



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

When n = 5

OUTPUT

File View Run Help

Animate: View: Format:

Slow Fast

Program

185	call	String.appendChar 2
186	push	constant 105
187	call	String.appendChar 2
188	push	constant 115
189	call	String.appendChar 2
190	push	constant 58
191	call	String.appendChar 2
192	call	Output.printString 1
193	pop	temp 0
194	push	local 1
195	call	Output.printInt 1
196	pop	temp 0
	label	Main.main\$IF_END0
197	push	constant 0
198	return	

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

This

That

Temp

0	0
1	0

Enter the value of n:15
The value of y is:1240

Stack

Call Stack

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

When n = 15

OUTPUT

File View Run Help

Print Step Back Step Forward Breakpoint Toggle Slow Fast Animate: No animation View: Screen Format: Decimal

Program

185	call	String.appendChar 2
186	push	constant 105
187	call	String.appendChar 2
188	push	constant 115
189	call	String.appendChar 2
190	push	constant 58
191	call	String.appendChar 2
192	call	Output.println 1
193	pop	temp 0
194	push	local 1
195	call	Output.println 1
196	pop	temp 0
	label	Main.main\$IF_END0
197	push	constant 0
198	return	

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

This

That

Temp

0	0
1	0

Enter the value of n:50
The value of y is:-22611

Stack

Call Stack

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

When n = 50

2. Write a Jack program which counts up to 30 sec and automatically resets to 0.

CODE LOGIC

- x We will write a jack code which will count from 0 to 30 with a delay of 1000 millisecond (1 second).
- x When counter hits 30 the counter will reset and again start from 0.

JACK CODE

Timer > Main.jack

```
1  class Main {
2      function void main() {
3          var int a, b, c;
4          var int x, y, z;
5          let x = 0;
6          let a = 1;
7          let b = 0;
8          do Output.printString("Counter Started: ");
9          do Output.println();
10         while (x<31) {
11             // do Output.moveCursor(a,b);
12             do Output.printInt(x);
13             do Output.println();
14             do Sys.wait(1000);
15             let x = x + 1;
16
17             if (x = 22) {
18                 do Screen.clearScreen();
19             }
20         }
21     }
22     do Screen.clearScreen();
23     let b = 0;
24     let a = 0;
25     do Output.moveCursor(a,b);
26 }
```

Timer > Main.jack

```
26     while (x>30) {
27         let x = 0;
28         do Output.printString("Counter reset: ");
29         do Output.println();
30         while (x<31) {
31             // do Output.moveCursor(a,b);
32             do Output.printInt(x);
33             do Output.println();
34             do Sys.wait(1000);
35             let x = x + 1;
36             if (x = 22) {
37                 do Screen.clearScreen();
38             }
39         }
40
41         if (x>30) {
42             do Screen.clearScreen();
43             let b = 0;
44             let a = 0;
45             do Output.moveCursor(a,b);
46         }
47     }
48 }
49 }
50 }
51 return;
52 }
53 }
```

JACK COMPILATION

TOOLS Timer > Main.jack

```
26 while (x>30) {
27     let x = 0;
28     do Output.printString("Counter reset: ");
29     do Output.println();
30     while (x<31) {
31         // do Output.moveCursor(a,b);
32         do Output.printInt(x);
33         do Output.println();
34         do Sys.wait(1000);
35         let x = x + 1;
36         if (x = 22) {
37             do Screen.clearScreen();
38         }
39     }
40     if (x>30) {
41         do Screen.clearScreen();
42         let b = 0;
43     }
```

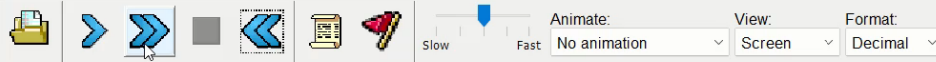
PROBLEMS COMMENTS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> .\JackCompiler.bat .\Summation\Main.jack
Compiling "C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools\Summation\Main.jack"
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> .\JackCompiler.bat .\Timer\Main.jack
Compiling "C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools\Timer\Main.jack"
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> 
```


OUTPUT

Virtual Machine Emulator (2.5) - C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools\Timer\Main.vm

File View Run Help



Program

156	push	constant 0
157	pop	local 1
158	push	constant 0
159	pop	local 0
160	push	local 0
161	push	local 1
162	call	Output.moveCursor 2
163	pop	temp 0
	label	Main.main\$IF_FALSE2
164	goto	Main.main\$WHILE_EXP2
	label	Main.main\$WHILE_END2
165	goto	Main.main\$WHILE_EXP1
	label	Main.main\$WHILE_END1
166	push	constant 0
167	return	

Stack

Call Stack

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

This

That

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

ALTERNATIVE JACK CODE

```

Main - Notepad
File Edit View

class Main {
function void main() {
    var int k;
    let k=1;

    while (true) {
        if (k=31){
            let k=0;
            do Screen.clearScreen();
        }

        do Output.printInt(k);
        do Output.printString(" ");
        do Sys.wait(1000);

        if (k=0) {
            do Sys.halt();
        }

        let k=k+1;
    }

    return;
}
}
```

NEXT WE HAVE TO COMPILE THAT IN COMMAND PROMPT

C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.22000.675]

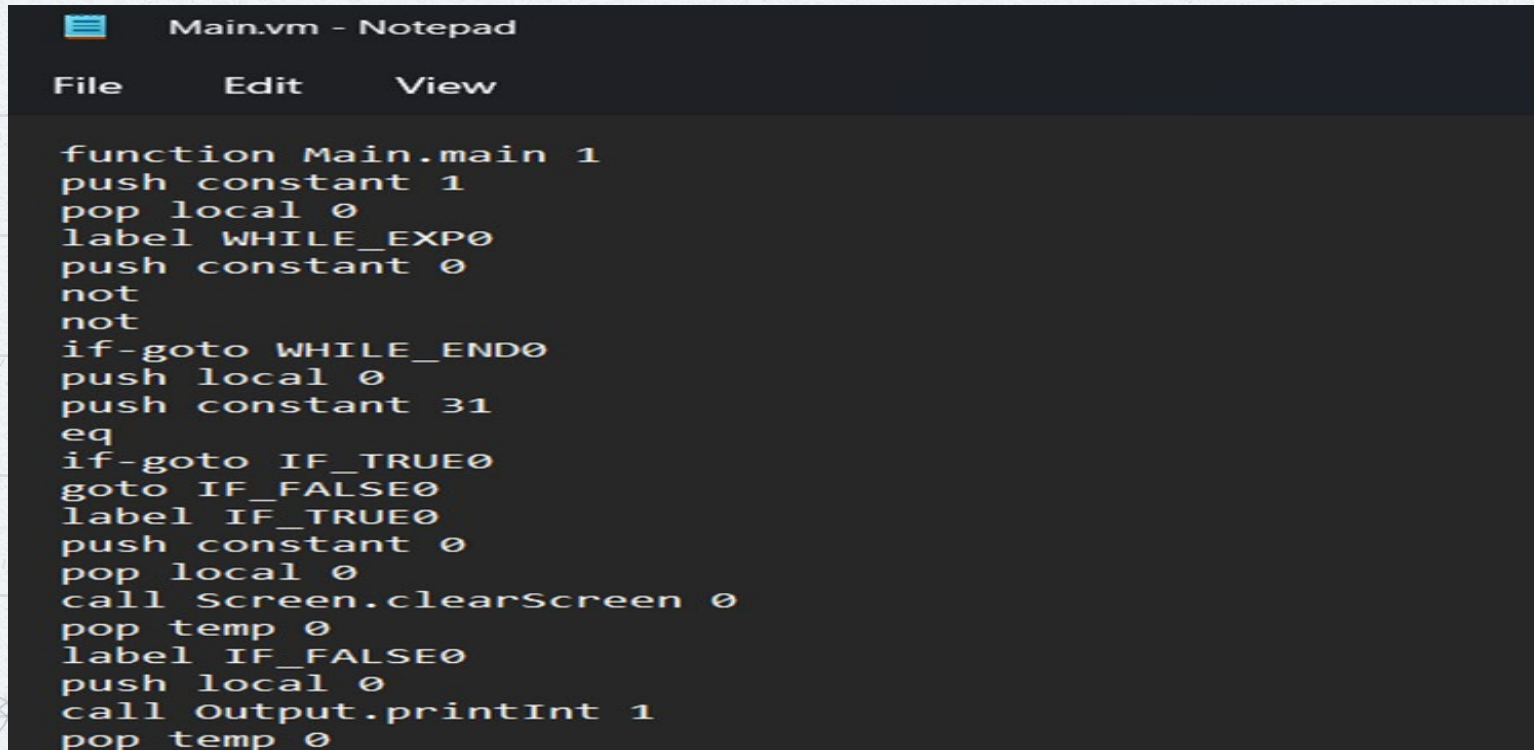
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\Downloads\nand2tetris\nand2tetris\tools>JackCompiler.bat counter

Compiling "C:\Users\DELL\Downloads\nand2tetris\nand2tetris\tools\counter"

C:\Users\DELL\Downloads\nand2tetris\nand2tetris\tools>_

NEXT WE CAN FIND .VM FILE IN FOLDER AS BELOW



```
function Main.main 1
push constant 1
pop local 0
label WHILE_EXP0
push constant 0
not
not
if-goto WHILE_END0
push local 0
push constant 31
eq
if-goto IF_TRUE0
goto IF_FALSE0
label IF_TRUE0
push constant 0
pop local 0
call Screen.clearScreen 0
pop temp 0
label IF_FALSE0
push local 0
call Output.printInt 1
pop temp 0
```

```
push constant 1000
call Sys.wait 1
pop temp 0
push local 0
push constant 0
eq
if-goto IF_TRUE1
goto IF_FALSE1
label IF_TRUE1
call Sys.halt 0
pop temp 0
label IF_FALSE1
push local 0
push constant 1
add
pop local 0
goto WHILE_EXP0
label WHILE_END0
push constant 0
return
```

ALTERNATIVE OUTPUT

Virtual Machine Emulator (2.5) - C:\Users\DELL\Downloads\nand2tetris\nand2tetris\tools\counter

File View Run Help

Animate: No animation View: Screen Format: Decimal

Program

19	push	constant 1
20	call	String.new 1
21	push	constant 32
22	call	String.appendCh...
23	call	Output.printStrin...
24	pop	temp 0
25	push	constant 1000
26	call	Sys.wait 1
27	pop	temp 0
28	push	local 0
29	push	constant 0
30	eq	
31	if-goto	Main.main\$IF_T...
32	goto	Main.main\$IF_F...
	label	Main.main\$IF_T...

Static

0	0
1	0
2	0
3	0
4	0

Local

Argument

0	1000

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Stack

ALTERNATIVE OUTPUT

virtual machine emulator (2.3) - C:\Users\DELL\Downloads\handztests\handztests\tools\counter

File View Run Help

Slow Fast Animate: No animation View: Screen Format: Decimal

Program

19	push	constant 1
20	call	String.new 1
21	push	constant 32
22	call	String.appendCh...
23	call	Output.printStrin...
24	pop	temp 0
25	push	constant 1000
26	call	Sys.wait 1
27	pop	temp 0
28	push	local 0
29	push	constant 0
30	eq	
31	if-goto	Main.main\$IF_T...
32	goto	Main.main\$IF_F...
	label	Main.main\$IF_T...

Static

0	0
1	0
2	0
3	0
4	0

Local

Argument

0	

This

Stack

Information Message

Program Halted

OK

3.

WRITE A JACK PROGRAM TO FIND THE SUM OF PRIME
NUMBERS FROM 50 TO 100.

CODE LOGIC

- x We will take input from user of two number that are ranging from $n=50$ to $n=100$.
- x Then we will check whether number starting from $n = 50$ till $n=100$ contains how many prime numbers and then add those prime numbers.

JACK CODE

Prime > Main.jack

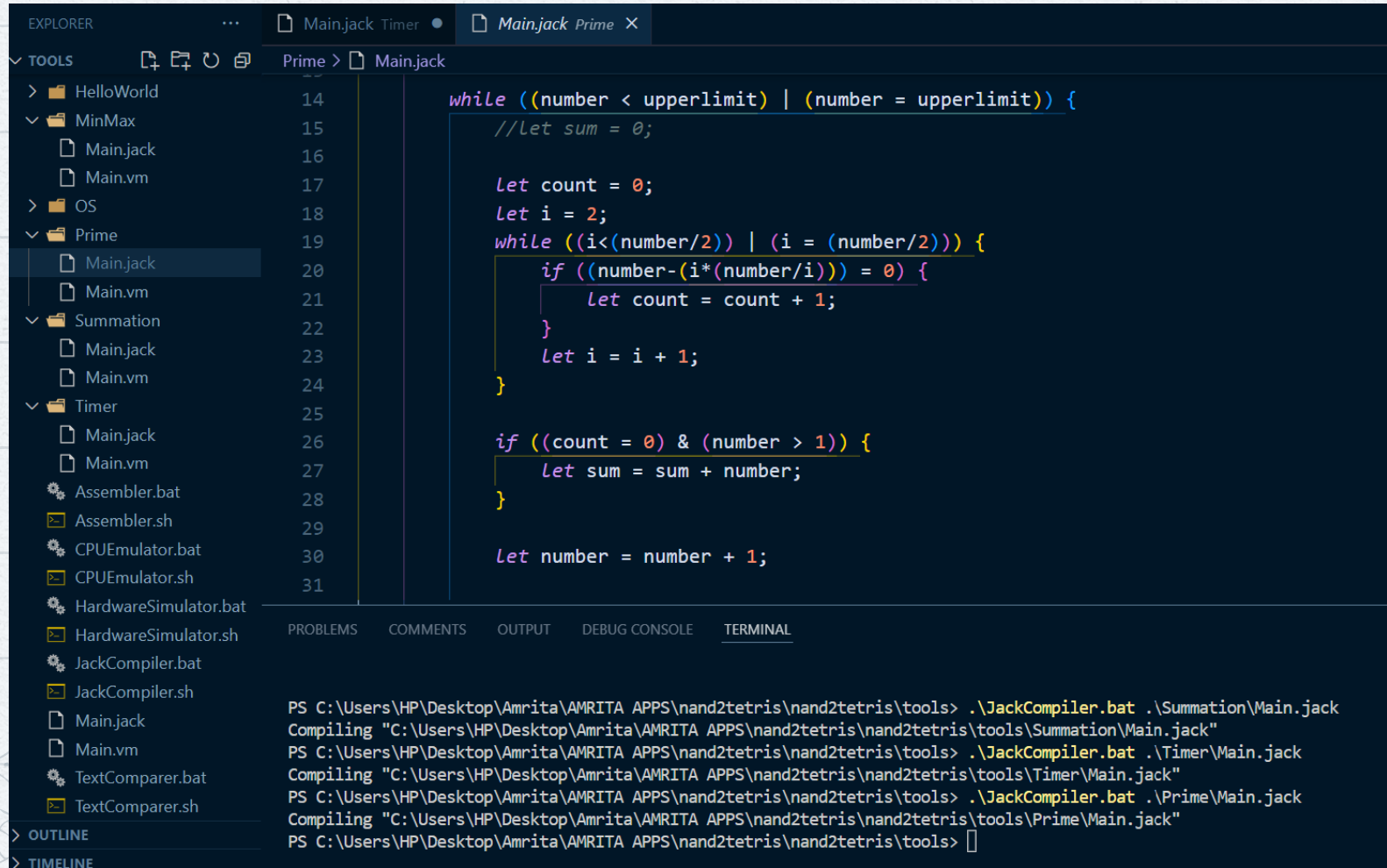
```
1  class Main {
2      function void main () {
3          var int count, sum, number, i;
4          var boolean k;
5          var int upperlimit;
6          var int number; // Starting range of prime number to be checked and keep number less than or equal
                           // to final number
7          let sum = 0;
8          let count = 0;
9
10
11         let number = Keyboard.readInt("Enter the first number: ");
12         let upperlimit = Keyboard.readInt("Enter the second number: ");
13
14         while ((number < upperlimit) | (number = upperlimit)) {
15             //let sum = 0;
16
17             let count = 0;
18             let i = 2;
19             while ((i < (number/2)) | (i = (number/2))) {
20                 if ((number - (i * (number/i))) = 0) {
21                     let count = count + 1;
22                 }
23                 let i = i + 1;
24             }
25
26             if ((count = 0) & (number > 1)) {
27                 let sum = sum + number;
```

JACK CODE

Prime > Main.jack

```
14 while ((number < upperlimit) | (number = upperlimit)) {
15     //let sum = 0;
16
17     let count = 0;
18     let i = 2;
19     while ((i < (number/2)) | (i = (number/2))) {
20         if ((number - (i * (number/i))) = 0) {
21             let count = count + 1;
22         }
23         let i = i + 1;
24     }
25
26     if ((count = 0) & (number > 1)) {
27         let sum = sum + number;
28     }
29
30     let number = number + 1;
31
32 }
33 do Output.println("The sum of the prime numbers between 50 and 100 is: ");
34 do Output.println(sum);
35 return;
36
37 }
38 }
```

JACK COMPILATION



EXPLORER

... Main.jack Timer Main.jack Prime X

TOOLS

Prime Main.jack

```
14 while ((number < upperlimit) | (number = upperlimit)) {
15     //let sum = 0;
16
17     let count = 0;
18     let i = 2;
19     while ((i < (number/2)) | (i = (number/2))) {
20         if ((number - (i * (number/i))) = 0) {
21             let count = count + 1;
22         }
23         let i = i + 1;
24     }
25
26     if ((count = 0) & (number > 1)) {
27         let sum = sum + number;
28     }
29
30     let number = number + 1;
31 }
```

PROBLEMS COMMENTS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> .\JackCompiler.bat .\Summation\Main.jack
Compiling "C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools\Summation\Main.jack"
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> .\JackCompiler.bat .\Timer\Main.jack
Compiling "C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools\Timer\Main.jack"
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> .\JackCompiler.bat .\Prime\Main.jack
Compiling "C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools\Prime\Main.jack"
PS C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools> }
```

OUTLINE

TIMELINE

OUTPUT

Virtual Machine Emulator (2.5) - C:\Users\HP\Desktop\Amrita\AMRITA APPS\nand2tetris\nand2tetris\tools\Prime\Main.vm

File View Run Help



Program

273	push	constant 105
274	call	String.appendChar 2
275	push	constant 115
276	call	String.appendChar 2
277	push	constant 58
278	call	String.appendChar 2
279	push	constant 32
280	call	String.appendChar 2
281	call	Output.printString 1
282	pop	temp 0
283	push	local 1
284	call	Output.printInt 1
285	pop	temp 0
286	push	constant 0
287	return	

Stack

--

Call Stack

--

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

This

That

--	--

Temp

0	0
1	0

Enter the first number: 50
Enter the second number: 100
The sum of the prime numbers between 50 and 100 is: 732

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

ALTERNATIVE JACK CODE

```
class Main {  
  function void main() {  
    var int i;  
    var int sum; //We took 2 variables one for the Loop and another to store the output.  
    Let i = 50;  
    do Output.printString("Prime numbers from 50 to 100: "); //It prints the statement which represent the numbers we stored in the sum  
  
    while (i < 100) {  
      if (~(Main.rem(i, 2) = 0) | (Main.rem(i, 3) = 0) | (Main.rem(i, 5) = 0) | (Main.rem(i, 7) = 0))) {  
        Let sum = sum + i;  
      }  
  
      // We used the while Loop to check the numbers which are prime between 50-100 and we used the condition that remainder should not be 0 when we divide it with 2,3,5,7  
  
      do Output.printInt(i);  
      do Output.printString(","); //We printed all the prime numbers between 50-100  
  
      }  
      Let i = i + 1;  
    }  
  
    do Output.println();  
    do Output.printString("Sum of prime numbers from 50 to 100: ");  
    do Output.printInt(sum); //It prints sum  
    return;  
  }  
  
  function int rem(int dividend, int divisor) {  
    var int quotient;  
  
    Let quotient=dividend/divisor;  
    return dividend - (quotient * divisor); // Since remainder operation is not available we used division formula to find the reminder.  
  }  
}
```

VM SIMULATION

Animate: No animation
 View: Screen
 Format: Decimal

Program

197	call	Output.printInt 1
198	pop	temp 0
199	push	constant 0
200	return	
0	function	Main.mod 1
1	push	argument 0
2	push	argument 1
3	call	Math.divide 2
4	pop	local 0
5	push	argument 0
6	push	local 0
7	push	argument 1
8	call	Math.multiply 2
9	sub	
10	return	

Static

0	0
1	0
2	0
3	0
4	0

Local

Argument

This

That

Temp

0	0
1	0

Stack

Call Stack

Prime numbers from 50 to 100: 53,59,61,67,71,73,79,83,89,97.
Sum of prime numbers from 50 to 100: 732

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.

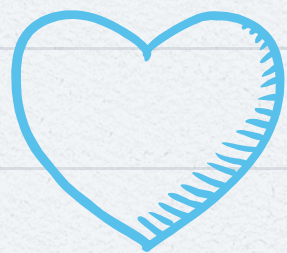
WRITE A BRIEF DESCRIPTION OF THE VARIOUS JACK OS SERVICES

INTRODUCTION

- The Jack language comes with a standard library that may also be viewed as an interface to an underlying operating system.
- The library is a collection of Jack classes, and must be provided in every implementation of the Jack language.

- String:** Implements the String type and basic.

- Sys:** Provides some execution-related services.



THANKS!