Mrudhula Ashok Shenava SUID: 265469976

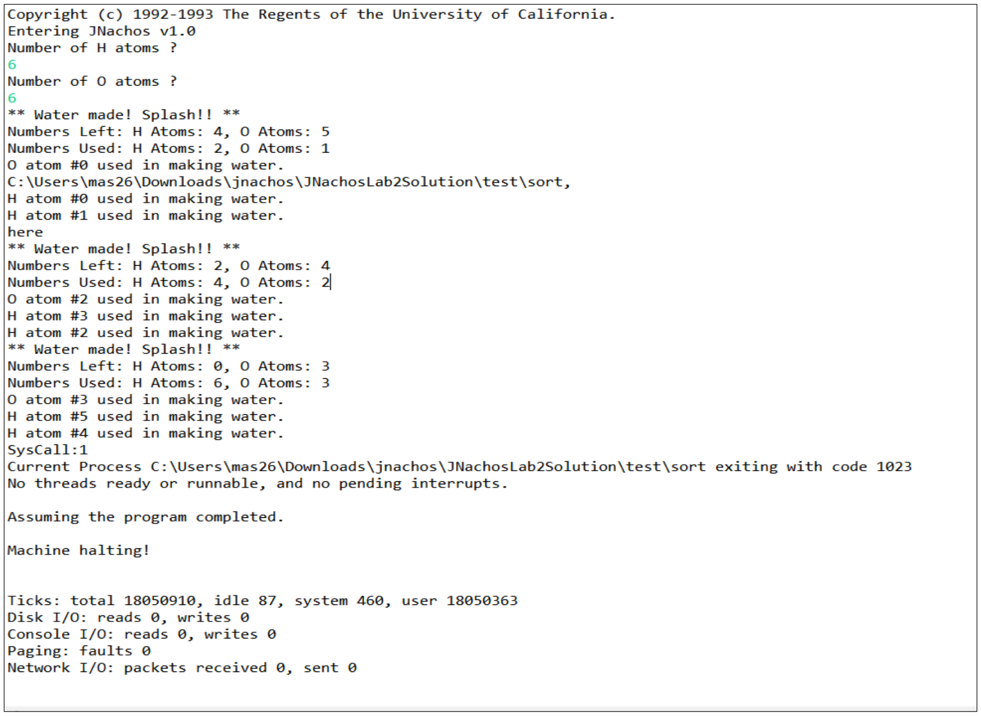
# PRINCIPALS OF OPERATING SYSTEMS
# LAB 3

**1) Run JNachos with Water uncommented. It prompts you for the number of H atoms and number of O atoms. Run the program with several different console Inputs. Explain the output in terms of the input (not the code).**

**Solution:**

OBSERVATION:

Case 1: For H atoms = 6 and O atoms = 6



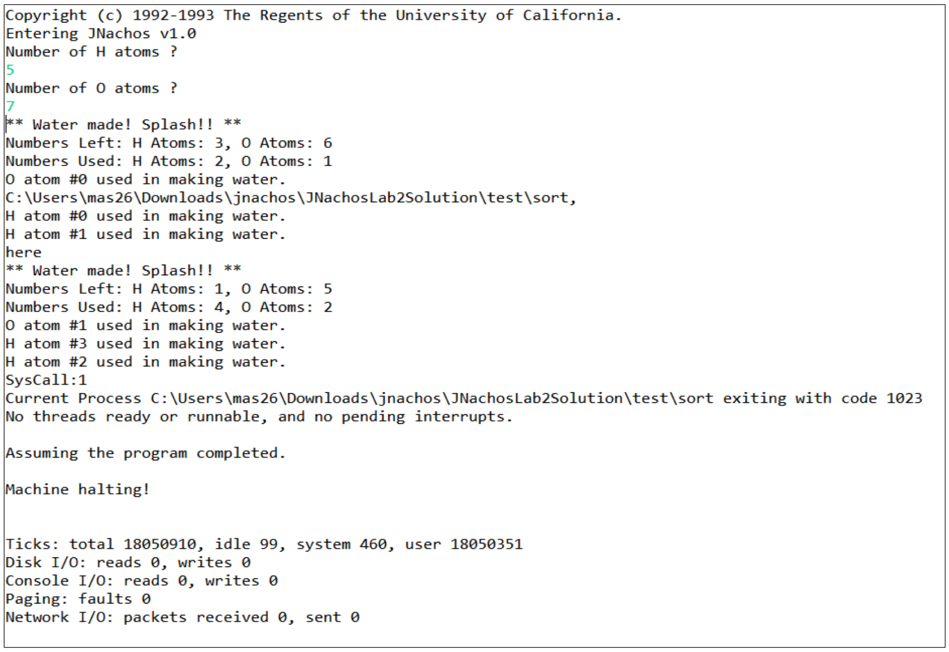
```
Copyright (c) 1992-1993 The Regents of the University of California.
Entering JNachos v1.0
Number of H atoms ?
6
Number of O atoms ?
6
** Water made! Splash!! **
Numbers Left: H Atoms: 4, O Atoms: 5
Numbers Used: H Atoms: 2, O Atoms: 1
O atom #0 used in making water.
C:\Users\mas26\Downloads\jnachos\JNachosLab2Solution\test\sort,
H atom #0 used in making water.
H atom #1 used in making water.
here
** Water made! Splash!! **
Numbers Left: H Atoms: 2, O Atoms: 4
Numbers Used: H Atoms: 4, O Atoms: 2
O atom #2 used in making water.
H atom #3 used in making water.
H atom #2 used in making water.
** Water made! Splash!! **
Numbers Left: H Atoms: 0, O Atoms: 3
Numbers Used: H Atoms: 6, O Atoms: 3
O atom #3 used in making water.
H atom #5 used in making water.
H atom #4 used in making water.
SysCall:1
Current Process C:\Users\mas26\Downloads\jnachos\JNachosLab2Solution\test\sort exiting with code 1023
No threads ready or runnable, and no pending interrupts.

Assuming the program completed.

Machine halting!


Ticks: total 18050910, idle 87, system 460, user 18050363
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
```

EXPLANATION:

In the above case, we have considered H and O to have equal number of atoms that is 6. We see that in the initial water splash 2 H atoms and 1 O atoms is used, where, 4 H atoms and 5 O atoms are remaining. Since there are atoms available to be implemented, further water splash is called in which 2 more H atoms and 1 more of O atom is used. Therefore, now there is 2 more of H atom and 4 of O atoms remaining. Since there are still 2 more of H atoms and 4 of O atoms available, water is made again using up 2 atoms of H and 1 atom of O. Therefore, there are no more of H atoms remaining and there are still 3 more O atoms. But since there are no more of H atoms the program stops implementing with the total number atoms needed to make water.

OBSERVATION:

Case 2: For H atoms = 5 and O atoms = 7



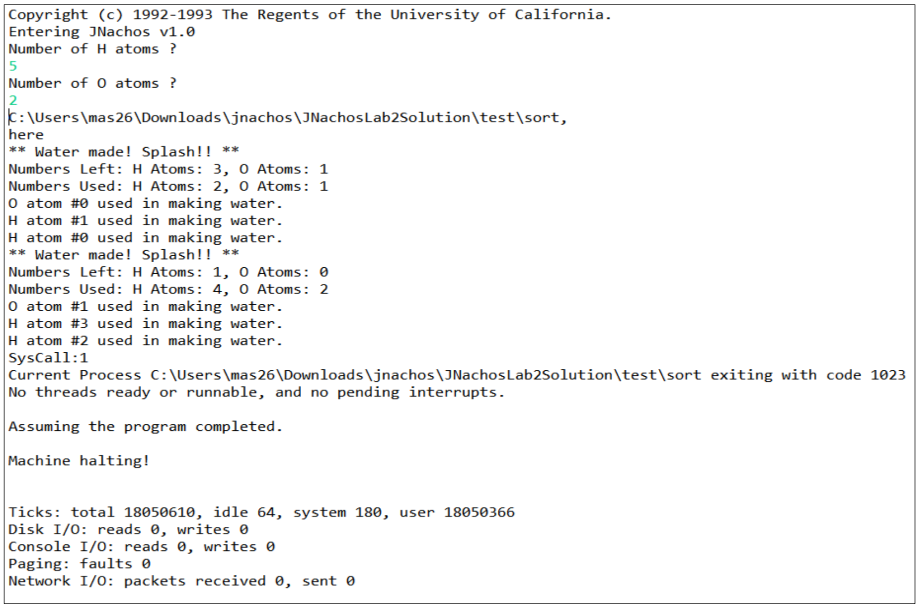
```
Copyright (c) 1992-1993 The Regents of the University of California.
Entering JNachos v1.0
Number of H atoms ?
5
Number of O atoms ?
7
** Water made! Splash!! **
Numbers Left: H Atoms: 3, O Atoms: 6
Numbers Used: H Atoms: 2, O Atoms: 1
O atom #0 used in making water.
C:\Users\mas26\Downloads\jnachos\JNachosLab2Solution\test\sort,
H atom #0 used in making water.
H atom #1 used in making water.
here
** Water made! Splash!! **
Numbers Left: H Atoms: 1, O Atoms: 5
Numbers Used: H Atoms: 4, O Atoms: 2
O atom #1 used in making water.
H atom #3 used in making water.
H atom #2 used in making water.
SysCall:1
Current Process C:\Users\mas26\Downloads\jnachos\JNachosLab2Solution\test\sort exiting with code 1023
No threads ready or runnable, and no pending interrupts.

Assuming the program completed.

Machine halting!


Ticks: total 18050910, idle 99, system 460, user 18050351
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
```

EXPLANATION:

In this case, we take a smaller value for H atom compared to O atom. Let H atom be 5 and O atom be 7. Initially 2 H atoms and 1 O atom is used to make water splash. Now, there are 3 H atoms and 6 O atoms left. Checking the condition, we can further implement as there are more than 2 H atoms and more than 1 O atom that can be used. Hence, another 2 H atoms and 1 O atom is used to make water splash, leaving, 1 H atom and 5 more of O atom. Since, we need 2 H atoms and 1 of O atom to make water splash and in this case, we have only 1 H atom but 6 of O atoms we cannot implement further due to less than 2 H atoms are available.

OBSERVATION:

Case 3: For H atoms = 5 and O atoms = 2



```
Copyright (c) 1992-1993 The Regents of the University of California.
Entering JNachos v1.0
Number of H atoms ?
5
Number of O atoms ?
2
C:\Users\mas26\Downloads\jnachos\JNachosLab2Solution\test\sort,
here
** Water made! Splash!! **
Numbers Left: H Atoms: 3, O Atoms: 1
Numbers Used: H Atoms: 2, O Atoms: 1
O atom #0 used in making water.
H atom #1 used in making water.
H atom #0 used in making water.
** Water made! Splash!! **
Numbers Left: H Atoms: 1, O Atoms: 0
Numbers Used: H Atoms: 4, O Atoms: 2
O atom #1 used in making water.
H atom #3 used in making water.
H atom #2 used in making water.
SysCall:1
Current Process C:\Users\mas26\Downloads\jnachos\JNachosLab2Solution\test\sort exiting with code 1023
No threads ready or runnable, and no pending interrupts.

Assuming the program completed.

Machine halting!


Ticks: total 18050610, idle 64, system 180, user 18050366
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
```

EXPLANATION:

In this case, we take a greater value for H atom compared to O atom. Let H atom be 5 and O atom be 2. Initially 2 H atoms and 1 O atom is used to make water splash. Now, there are 3 H atoms and 1 O atoms left. Checking the condition, we can further implement as there are more than 2 H atoms and at least 1 O atom that can be used. Hence, another 2 H atoms and 1 O atom is used to make water splash, leaving, 1 H atom and no more of O atom. Since, we need 2 H atoms and 1 of O atom to make water splash and in this case, we have only 1 H atom but none of O atoms we cannot implement further as there are no more of O atoms left to be used and there is no more than 2 H atoms.

**2) Run JNachos with the program argument '-rs', where seed is a number. Try with different numbers, is there any difference?**

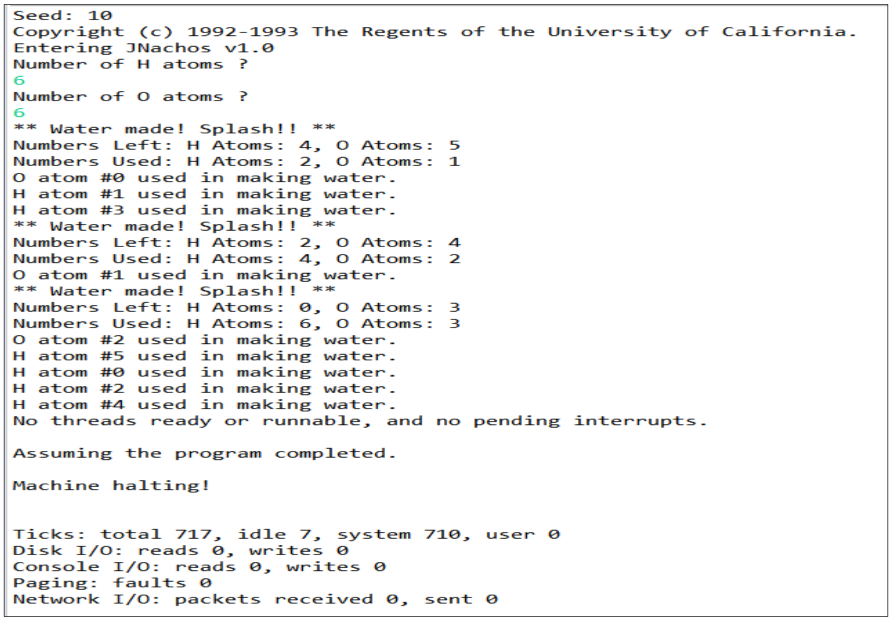
**Solution:**

WITH SEED VALUE

Assigning seed value to be 10

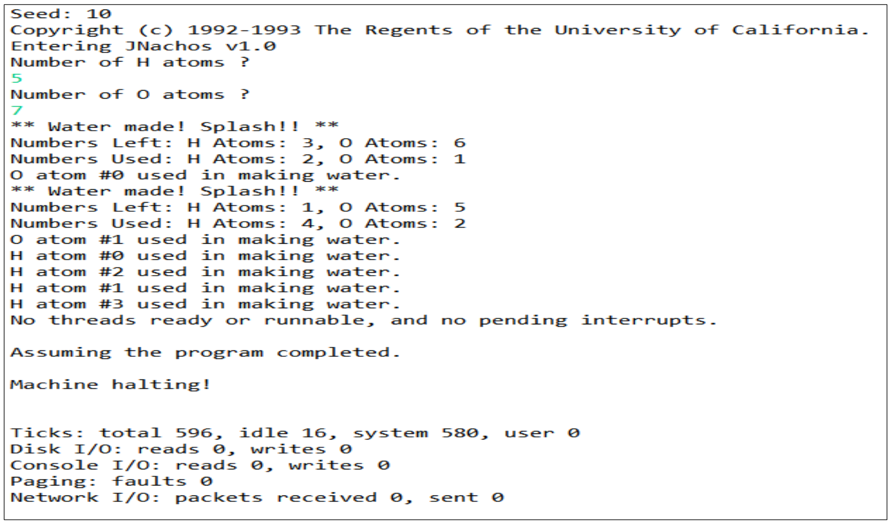
OBSERVATION:

For H atoms = 6 and O atoms = 6



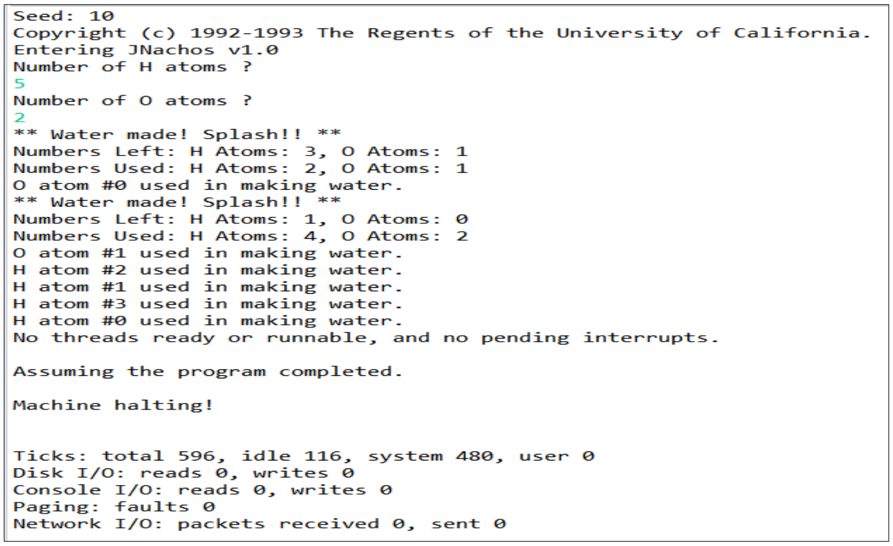
```
Seed: 10
Copyright (c) 1992-1993 The Regents of the University of California.
Entering JNachos v1.0
Number of H atoms ?
6
Number of O atoms ?
6
** Water made! Splash!! **
Numbers Left: H Atoms: 4, O Atoms: 5
Numbers Used: H Atoms: 2, O Atoms: 1
O atom #0 used in making water.
H atom #1 used in making water.
H atom #3 used in making water.
** Water made! Splash!! **
Numbers Left: H Atoms: 2, O Atoms: 4
Numbers Used: H Atoms: 4, O Atoms: 2
O atom #1 used in making water.
** Water made! Splash!! **
Numbers Left: H Atoms: 0, O Atoms: 3
Numbers Used: H Atoms: 6, O Atoms: 3
O atom #2 used in making water.
H atom #5 used in making water.
H atom #0 used in making water.
H atom #2 used in making water.
H atom #4 used in making water.
No threads ready or runnable, and no pending interrupts.

Assuming the program completed.

Machine halting!


Ticks: total 717, idle 7, system 710, user 0
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
```

For H atoms = 5 and O atoms = 7



```
Seed: 10
Copyright (c) 1992-1993 The Regents of the University of California.
Entering JNachos v1.0
Number of H atoms ?
5
Number of O atoms ?
7
** Water made! Splash!! **
Numbers Left: H Atoms: 3, O Atoms: 6
Numbers Used: H Atoms: 2, O Atoms: 1
O atom #0 used in making water.
** Water made! Splash!! **
Numbers Left: H Atoms: 1, O Atoms: 5
Numbers Used: H Atoms: 4, O Atoms: 2
O atom #1 used in making water.
H atom #0 used in making water.
H atom #2 used in making water.
H atom #1 used in making water.
H atom #3 used in making water.
No threads ready or runnable, and no pending interrupts.

Assuming the program completed.

Machine halting!


Ticks: total 596, idle 16, system 580, user 0
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
```

For H atoms = 5 and O atoms = 2



```
Seed: 10
Copyright (c) 1992-1993 The Regents of the University of California.
Entering JNachos v1.0
Number of H atoms ?
5
Number of O atoms ?
2
** Water made! Splash!! **
Numbers Left: H Atoms: 3, O Atoms: 1
Numbers Used: H Atoms: 2, O Atoms: 1
O atom #0 used in making water.
** Water made! Splash!! **
Numbers Left: H Atoms: 1, O Atoms: 0
Numbers Used: H Atoms: 4, O Atoms: 2
O atom #1 used in making water.
H atom #2 used in making water.
H atom #1 used in making water.
H atom #3 used in making water.
H atom #0 used in making water.
No threads ready or runnable, and no pending interrupts.

Assuming the program completed.

Machine halting!


Ticks: total 596, idle 116, system 480, user 0
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
```

EXPLANATION:

I have used the same values for H atom an O atom as to the previous screenshots before we assigned seed value, to take note of the output and differentiate. In arguments we write '-rs <seed>' where the seed value is 10. From the screenshot, after inputting the seed value we see that the implementation of H atom and O atom used to make water splash is in random order, that is, the order in which the atoms are used are not in ascending numerical order instead it all called randomly. Which in case of without seed values the atoms were called in a numerical order.

3) **Trace through the logic of what the -rs flag does. Explain this logic here.**

**Solution:**

In this program '-rs' flag causes Yield to occur at random spots. Yield will return a generator but in this case due to the '-rs <seed #>' the H atoms and O atoms are generated randomly, that is the program implementation happens as per the code logic. But, when the atoms are generated whichever atom is ready at that point for H atom and O atom, that will be placed in output. Hence, atoms are not called in numerical increasing order any atom can be implement and are called in a random order.

4) **There are 5 Semaphore instances, explain the purpose of each instance. For “true” Semaphores explain what the associated count represents. For a mutex explain what critical section each is protecting.**

**Solution:**

Semaphore H - represents the number of H atoms used in the creation of water molecules.
Semaphore O - represents the number of O atoms used in the creation of water molecules.
Semaphore wait - used to ensure that only 2 H atoms and 1 O atom are used to create a water molecule. Puts the H process to sleep while using O and vice-versa.
Semaphore mutex - protects the critical section that uses the H and O atoms to create water molecules.
Semaphore mutex1 - protects the critical section that updates the count of H and O atoms.

5) **Explain the logic of HAtom and OAtom.**

**Solution:**

Under the HAtom and OAtom class, id is created. There are 5 semaphores being implemented. Taking the HAtom class first, the program initially checks for the condition if there are at least 2 H atoms so that the program can be implemented once this condition satisfies and is true. If true the counter is set and the count is taken for H atoms that is 2 H atoms per water splash. There is a wait set up for the second H atom to be made once H atoms are implemented the O atoms wake up for implementation. Next, we take the OAtom class, in this we need only 1 O atom to be implement per water splash. When H atoms are used we set the Hcount to ‘Hcount = Hcount – 2’ and Ocount to ‘Ocount --'.