For Windows OS and Linux (e.g. Ubuntu,...):

**Step 1:** Download the Algorithms.rar file from the link and extract it

**Step 2:**

+ For Windows: Go to the windows start menu and open the **Command Prompt application**.

+ For Linux: Open **Terminal application**

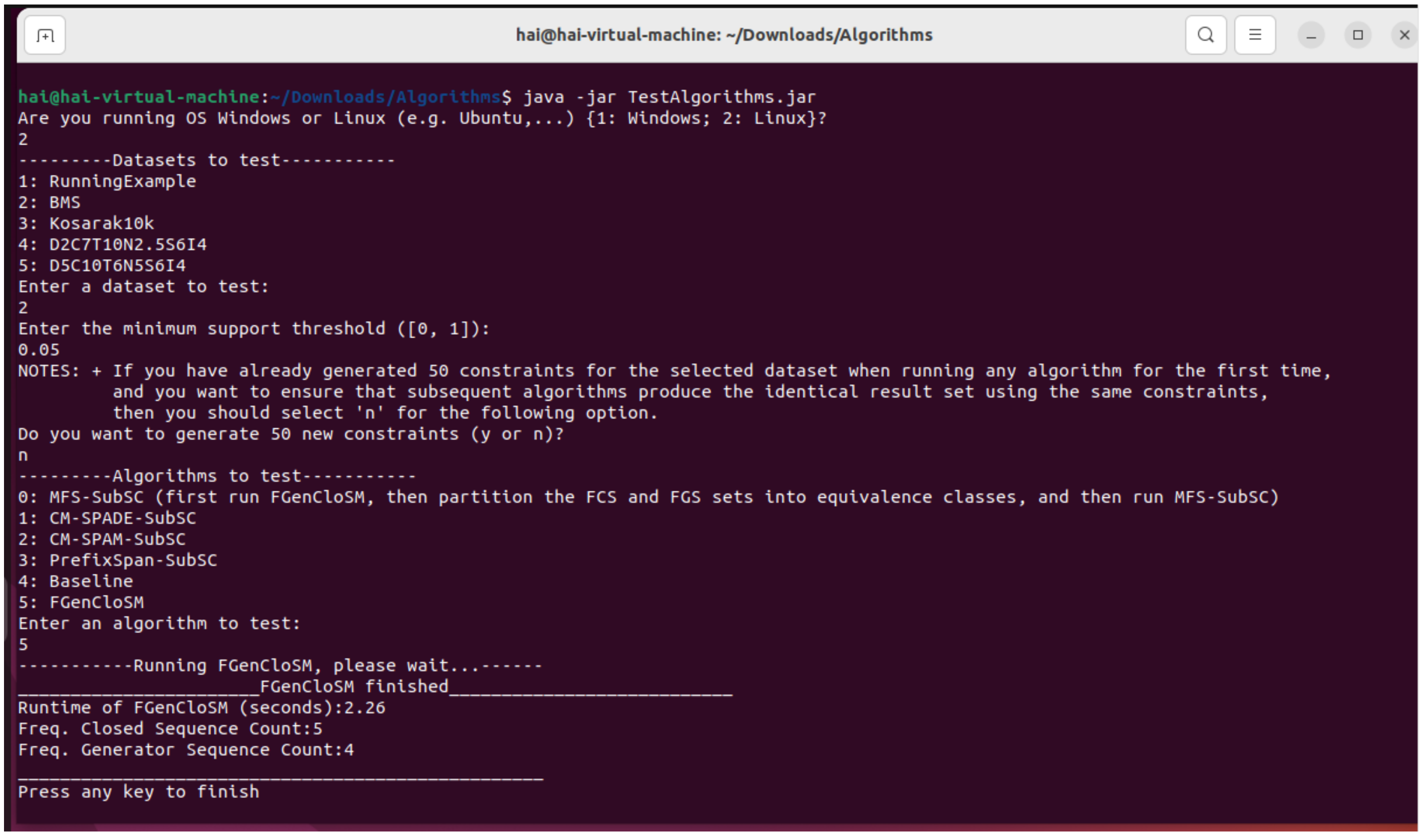
**Step 3:** Go to the extracted Algorithms folder and type the following command:

```
java -jar TestAlgorithms.jar
```

**Notes**: To execute the program with low values of $minsup$ on datasets (e.g. $minsup \leq 0.059\%$ on the BMS dataset...), ensure that the program's heap space (or memory) is a minimum of 12GB.

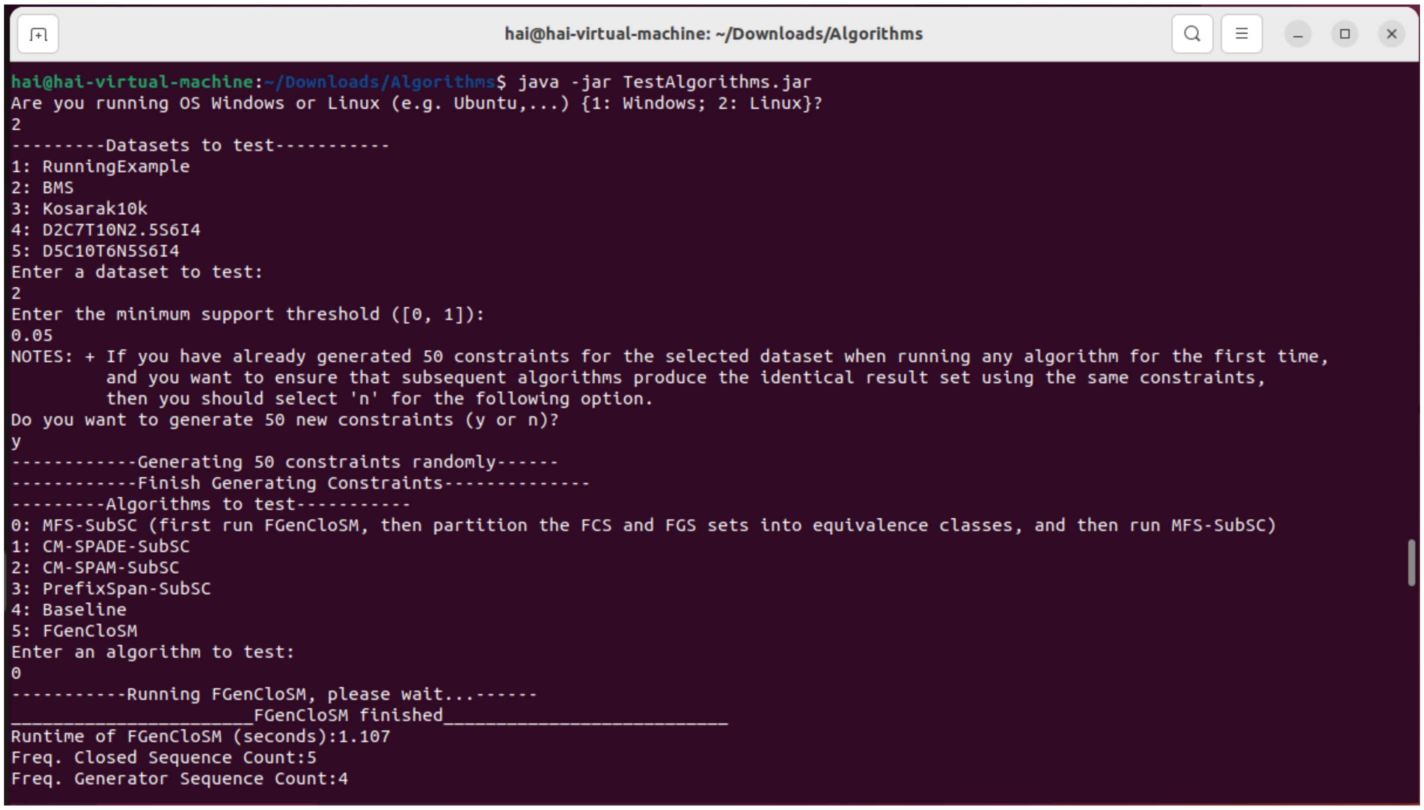
***Below, you'll find examples illustrating how to execute the program.***

1) *The first example demonstrates running the program on Ubuntu 22 as follows:*



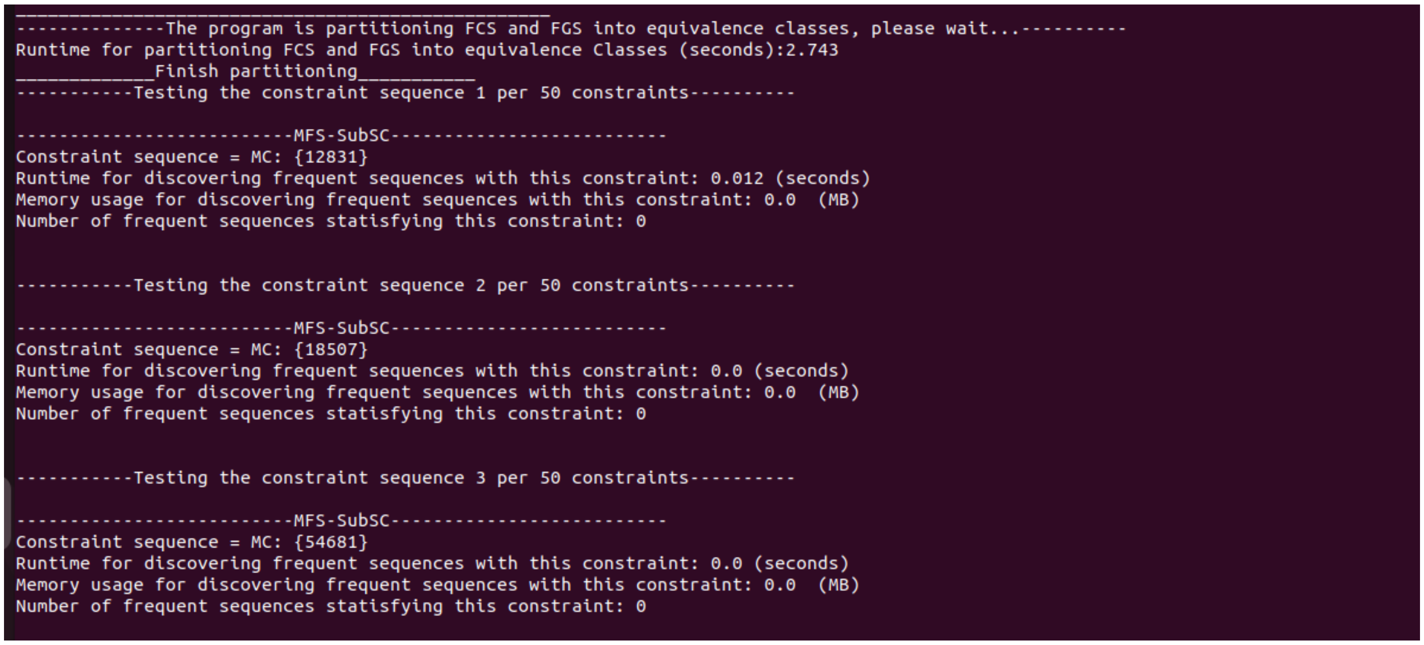
```
hai@hai-virtual-machine:~/Downloads/Algorithms$ java -jar TestAlgorithms.jar
Are you running OS Windows or Linux (e.g. Ubuntu,...) {1: Windows; 2: Linux}?
2
---------Datasets to test----------
1: RunningExample
2: BMS
3: Kosarak10k
4: D2C7T10N2.5S6I4
5: D5C10T6N5S6I4
Enter a dataset to test:
2
Enter the minimum support threshold ([0, 1]):
0.05
NOTES: + If you have already generated 50 constraints for the selected dataset when running any algorithm for the first time,
         and you want to ensure that subsequent algorithms produce the identical result set using the same constraints,
         then you should select 'n' for the following option.
Do you want to generate 50 new constraints (y or n)?
n
---------Algorithms to test----------
0: MFS-SubSC (first run FGenCloSM, then partition the FCS and FGS sets into equivalence classes, and then run MFS-SubSC)
1: CM-SPADE-SubSC
2: CM-SPAM-SubSC
3: PrefixSpan-SubSC
4: Baseline
5: FGenCloSM
Enter an algorithm to test:
5
-----------Running FGenCloSM, please wait...------
_______________________FGenCloSM finished___________________________
Runtime of FGenCloSM (seconds):2.26
Freq. Closed Sequence Count:5
Freq. Generator Sequence Count:4

__________________________________________________
Press any key to finish
```

*2) The second example demonstrates running the program on Ubuntu 22 as follows:*



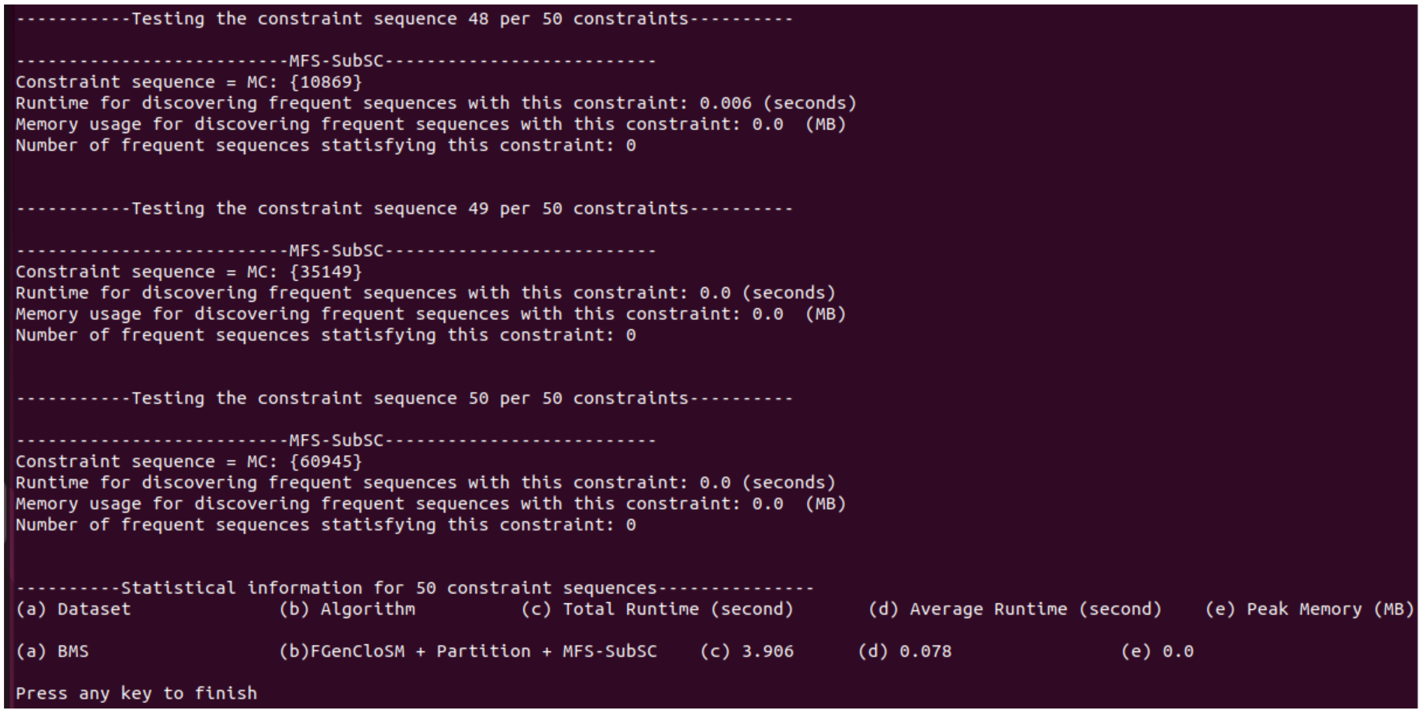
hai@hai-virtual-machine: ~/Downloads/Algorithms

```
hai@hai-virtual-machine:~/Downloads/Algorithms$ java -jar TestAlgorithms.jar
Are you running OS Windows or Linux (e.g. Ubuntu,...) {1: Windows; 2: Linux}?
2
---------Datasets to test----------
1: RunningExample
2: BMS
3: Kosarak10k
4: D2C7T10N2.5S6I4
5: D5C10T6N5S6I4
Enter a dataset to test:
2
Enter the minimum support threshold ([0, 1]):
0.05
NOTES: + If you have already generated 50 constraints for the selected dataset when running any algorithm for the first time,
         and you want to ensure that subsequent algorithms produce the identical result set using the same constraints,
         then you should select 'n' for the following option.
Do you want to generate 50 new constraints (y or n)?
y
------------Generating 50 constraints randomly------
------------Finish Generating Constraints-------------
---------Algorithms to test----------
0: MFS-SubSC (first run FGenCloSM, then partition the FCS and FGS sets into equivalence classes, and then run MFS-SubSC)
1: CM-SPADE-SubSC
2: CM-SPAM-SubSC
3: PrefixSpan-SubSC
4: Baseline
5: FGenCloSM
Enter an algorithm to test:
0
-----------Running FGenCloSM, please wait...------
_______________________FGenCloSM finished___________________________
Runtime of FGenCloSM (seconds):1.107
Freq. Closed Sequence Count:5
Freq. Generator Sequence Count:4
```



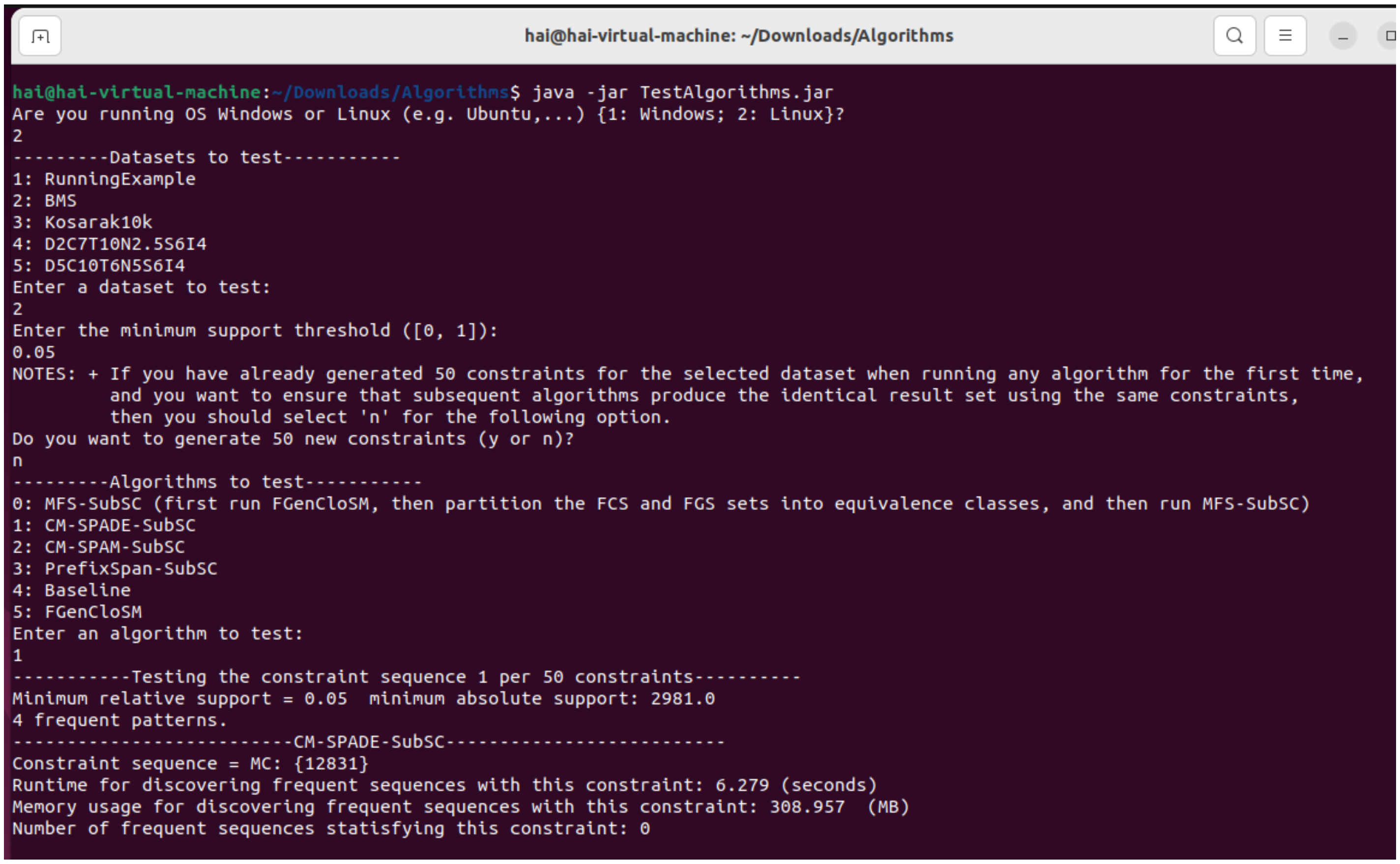
```
__________________________________________________
-------------The program is partitioning FCS and FGS into equivalence classes, please wait...----------
Runtime for partitioning FCS and FGS into equivalence Classes (seconds):2.743
_____________Finish partitioning___________
-----------Testing the constraint sequence 1 per 50 constraints----------

-------------------------MFS-SubSC-------------------------
Constraint sequence = MC: {12831}
Runtime for discovering frequent sequences with this constraint: 0.012 (seconds)
Memory usage for discovering frequent sequences with this constraint: 0.0  (MB)
Number of frequent sequences statisfying this constraint: 0


-----------Testing the constraint sequence 2 per 50 constraints----------

-------------------------MFS-SubSC-------------------------
Constraint sequence = MC: {18507}
Runtime for discovering frequent sequences with this constraint: 0.0 (seconds)
Memory usage for discovering frequent sequences with this constraint: 0.0  (MB)
Number of frequent sequences statisfying this constraint: 0


-----------Testing the constraint sequence 3 per 50 constraints----------

-------------------------MFS-SubSC-------------------------
Constraint sequence = MC: {54681}
Runtime for discovering frequent sequences with this constraint: 0.0 (seconds)
Memory usage for discovering frequent sequences with this constraint: 0.0  (MB)
Number of frequent sequences statisfying this constraint: 0
```

…



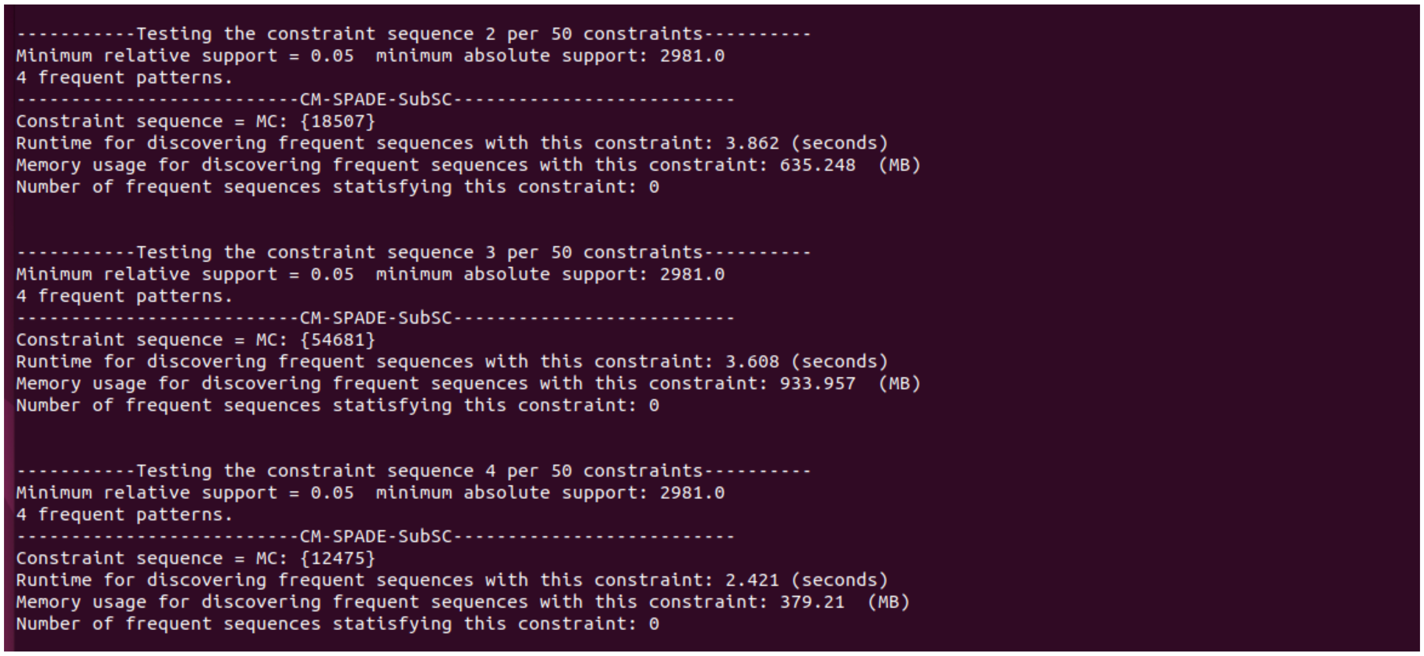
```
-----------Testing the constraint sequence 48 per 50 constraints----------

-------------------------MFS-SubSC-------------------------
Constraint sequence = MC: {10869}
Runtime for discovering frequent sequences with this constraint: 0.006 (seconds)
Memory usage for discovering frequent sequences with this constraint: 0.0  (MB)
Number of frequent sequences satisfying this constraint: 0


-----------Testing the constraint sequence 49 per 50 constraints----------

-------------------------MFS-SubSC-------------------------
Constraint sequence = MC: {35149}
Runtime for discovering frequent sequences with this constraint: 0.0 (seconds)
Memory usage for discovering frequent sequences with this constraint: 0.0  (MB)
Number of frequent sequences satisfying this constraint: 0


-----------Testing the constraint sequence 50 per 50 constraints----------

-------------------------MFS-SubSC-------------------------
Constraint sequence = MC: {60945}
Runtime for discovering frequent sequences with this constraint: 0.0 (seconds)
Memory usage for discovering frequent sequences with this constraint: 0.0  (MB)
Number of frequent sequences satisfying this constraint: 0


----------Statistical information for 50 constraint sequences---------------
(a) Dataset              (b) Algorithm          (c) Total Runtime (second)       (d) Average Runtime (second)    (e) Peak Memory (MB)

(a) BMS                  (b)FGenCloSM + Partition + MFS-SubSC    (c) 3.906      (d) 0.078                (e) 0.0

Press any key to finish
```

*3) The third example demonstrates running the program on Ubuntu 22 as follows:*



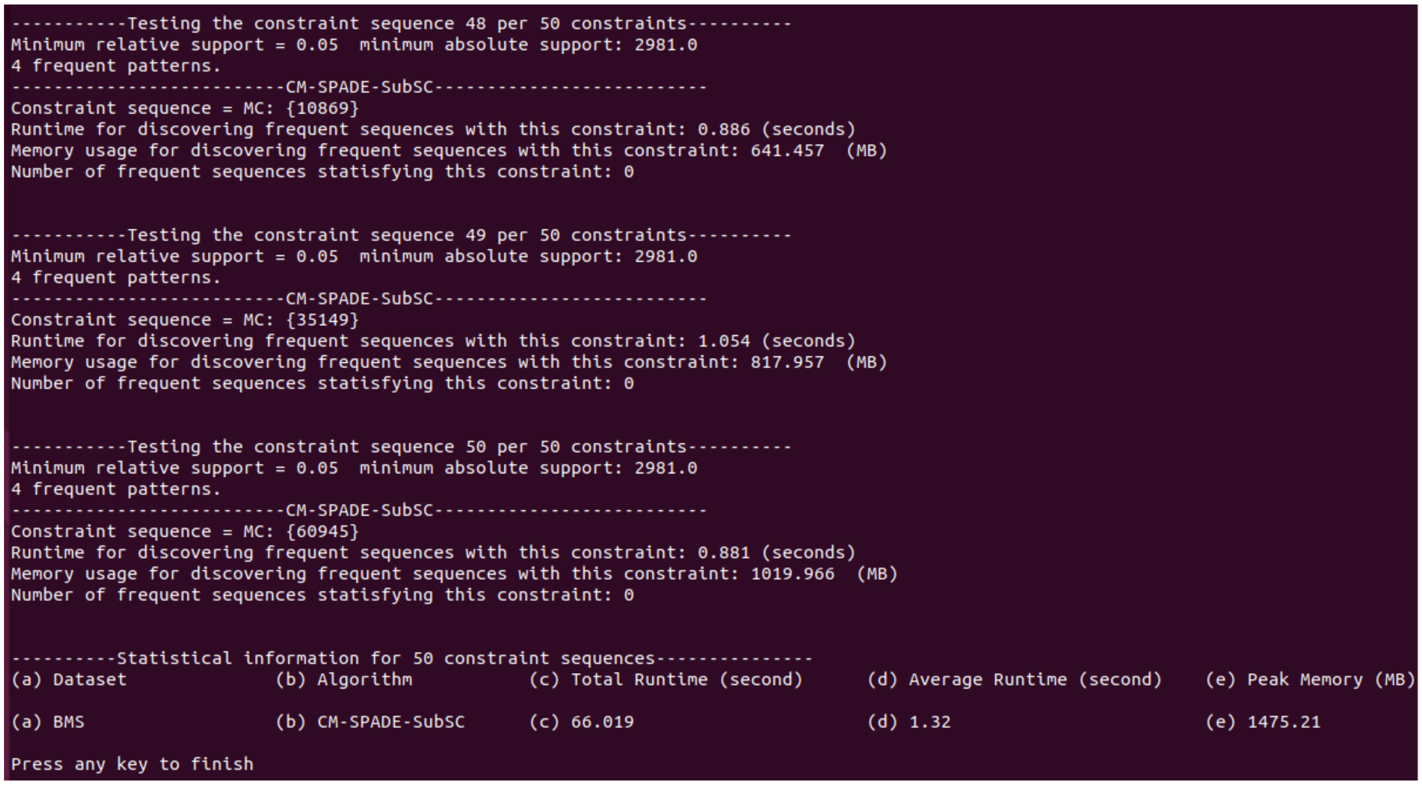
hai@hai-virtual-machine: ~/Downloads/Algorithms

```
hai@hai-virtual-machine:~/Downloads/Algorithms$ java -jar TestAlgorithms.jar
Are you running OS Windows or Linux (e.g. Ubuntu,...) {1: Windows; 2: Linux}?
2
---------Datasets to test-----------
1: RunningExample
2: BMS
3: Kosarak10k
4: D2C7T10N2.5S6I4
5: D5C10T6N5S6I4
Enter a dataset to test:
2
Enter the minimum support threshold ([0, 1]):
0.05
NOTES: + If you have already generated 50 constraints for the selected dataset when running any algorithm for the first time,
         and you want to ensure that subsequent algorithms produce the identical result set using the same constraints,
         then you should select 'n' for the following option.
Do you want to generate 50 new constraints (y or n)?
n
---------Algorithms to test-----------
0: MFS-SubSC (first run FGenCloSM, then partition the FCS and FGS sets into equivalence classes, and then run MFS-SubSC)
1: CM-SPADE-SubSC
2: CM-SPAM-SubSC
3: PrefixSpan-SubSC
4: Baseline
5: FGenCloSM
Enter an algorithm to test:
1
-----------Testing the constraint sequence 1 per 50 constraints----------
Minimum relative support = 0.05  minimum absolute support: 2981.0
4 frequent patterns.
-------------------------CM-SPADE-SubSC-------------------------
Constraint sequence = MC: {12831}
Runtime for discovering frequent sequences with this constraint: 6.279 (seconds)
Memory usage for discovering frequent sequences with this constraint: 308.957  (MB)
Number of frequent sequences satisfying this constraint: 0
```



```
----------Testing the constraint sequence 2 per 50 constraints----------
Minimum relative support = 0.05  minimum absolute support: 2981.0
4 frequent patterns.
-------------------------CM-SPADE-SubSC-------------------------
Constraint sequence = MC: {18507}
Runtime for discovering frequent sequences with this constraint: 3.862 (seconds)
Memory usage for discovering frequent sequences with this constraint: 635.248  (MB)
Number of frequent sequences statisfying this constraint: 0


----------Testing the constraint sequence 3 per 50 constraints----------
Minimum relative support = 0.05  minimum absolute support: 2981.0
4 frequent patterns.
-------------------------CM-SPADE-SubSC-------------------------
Constraint sequence = MC: {54681}
Runtime for discovering frequent sequences with this constraint: 3.608 (seconds)
Memory usage for discovering frequent sequences with this constraint: 933.957  (MB)
Number of frequent sequences statisfying this constraint: 0


----------Testing the constraint sequence 4 per 50 constraints----------
Minimum relative support = 0.05  minimum absolute support: 2981.0
4 frequent patterns.
-------------------------CM-SPADE-SubSC-------------------------
Constraint sequence = MC: {12475}
Runtime for discovering frequent sequences with this constraint: 2.421 (seconds)
Memory usage for discovering frequent sequences with this constraint: 379.21  (MB)
Number of frequent sequences statisfying this constraint: 0
```

...



```
----------Testing the constraint sequence 48 per 50 constraints----------
Minimum relative support = 0.05  minimum absolute support: 2981.0
4 frequent patterns.
-------------------------CM-SPADE-SubSC-------------------------
Constraint sequence = MC: {10869}
Runtime for discovering frequent sequences with this constraint: 0.886 (seconds)
Memory usage for discovering frequent sequences with this constraint: 641.457  (MB)
Number of frequent sequences statisfying this constraint: 0


----------Testing the constraint sequence 49 per 50 constraints----------
Minimum relative support = 0.05  minimum absolute support: 2981.0
4 frequent patterns.
-------------------------CM-SPADE-SubSC-------------------------
Constraint sequence = MC: {35149}
Runtime for discovering frequent sequences with this constraint: 1.054 (seconds)
Memory usage for discovering frequent sequences with this constraint: 817.957  (MB)
Number of frequent sequences statisfying this constraint: 0


----------Testing the constraint sequence 50 per 50 constraints----------
Minimum relative support = 0.05  minimum absolute support: 2981.0
4 frequent patterns.
-------------------------CM-SPADE-SubSC-------------------------
Constraint sequence = MC: {60945}
Runtime for discovering frequent sequences with this constraint: 0.881 (seconds)
Memory usage for discovering frequent sequences with this constraint: 1019.966  (MB)
Number of frequent sequences statisfying this constraint: 0


----------Statistical information for 50 constraint sequences--------------
(a) Dataset             (b) Algorithm          (c) Total Runtime (second)      (d) Average Runtime (second)    (e) Peak Memory (MB)

(a) BMS                 (b) CM-SPADE-SubSC     (c) 66.019                      (d) 1.32                        (e) 1475.21

Press any key to finish
```