Discrete Structures

Lab (2)

Name	ID
Ranime Ahmed Elsayed Shehata.	21010531
Rowan Gamal Ahmed Elkhouly.	21010539

➢Problem Statement:

PART (1) : Power Set:

Given a set represented as array list of distinct strings, you have to generate the power set from the set. You have to implement the requirement in two ways:

1) Recursive approach

2) Iterative approach

You are not allowed to use built in Set data structure or any data structure similar to it.

PART (2) : Logical Expressions Solver:

Design and implement a robust parser capable of handling logical expressions. The parser should support a simplified syntax for propositional logic, including the parentheses for grouping and the following operations

- negation (~)

- conjunction (^)

- disjunction (v)

- implication (>)

Your program should take, as input, the expression and prompt the user for the boolean values of each variable in the expression. It should then evaluate the expression based on the given boolean values and output the result. Make sure to follow the precedence rules of the operators that were discussed in class.

You are required use following interfaces:

```
interface Expression {

String getRepresentation();

void setRepresentation(String representation);

} interface LogicalExpressionSolver {

 boolean evaluateExpression(Expression expression);

}
```

PART (3) : Inference Engine :

Design and implement inference engine that can apply inference rules to logical expressions. The engine should take logical expressions as input, identify applicable inference rules, and generate the corresponding output based on the rules' logic. The supported inference rules as follows:

- Modus ponens. Given expressions of the form "P > Q" and "P", the rule allows inferring "Q"

- Modus tollens. Given expressions of the form "P > Q" and "~Q", the rule allows inferring "~P"

- Hypothetical syllogism. Given expressions of the form "P > Q" and "Q > R", the rule allows inferring "P > R"

- Disjunctive syllogism. Given expressions of the form "P v Q" and "~P", the rule allows inferring "Q"

- Resolution. Given expressions of the form "P v Q" and "~P v R", the rule allows inferring "Q v R"

Your program should take two expressions as input and output the result of the inference process along with the applied rule. Assume that the inputs are simple expressions without parentheses and that each expression has at most one binary operation.

You are required to use following interfaces:

```
interface InferenceRule {

boolean matches(Expression exp1, Expression exp2);

Expression apply(Expression exp1, Expression exp2);

}

interface InferenceEngine {

 void addRule(InferenceRule rule);

void addExpression(Expression exp);

Expression applyRules();

}
```

➢ Used data Structures:

PART (1) : Power Set:

- ArrayList of type String to store the elements of the Universe.

PART (2) : Logical Expressions Solver:

- Stack for operators.
- Stack for operands.
- ArrayList of type Character for literals.
- ArrayList of typr Boolean for values of literals.

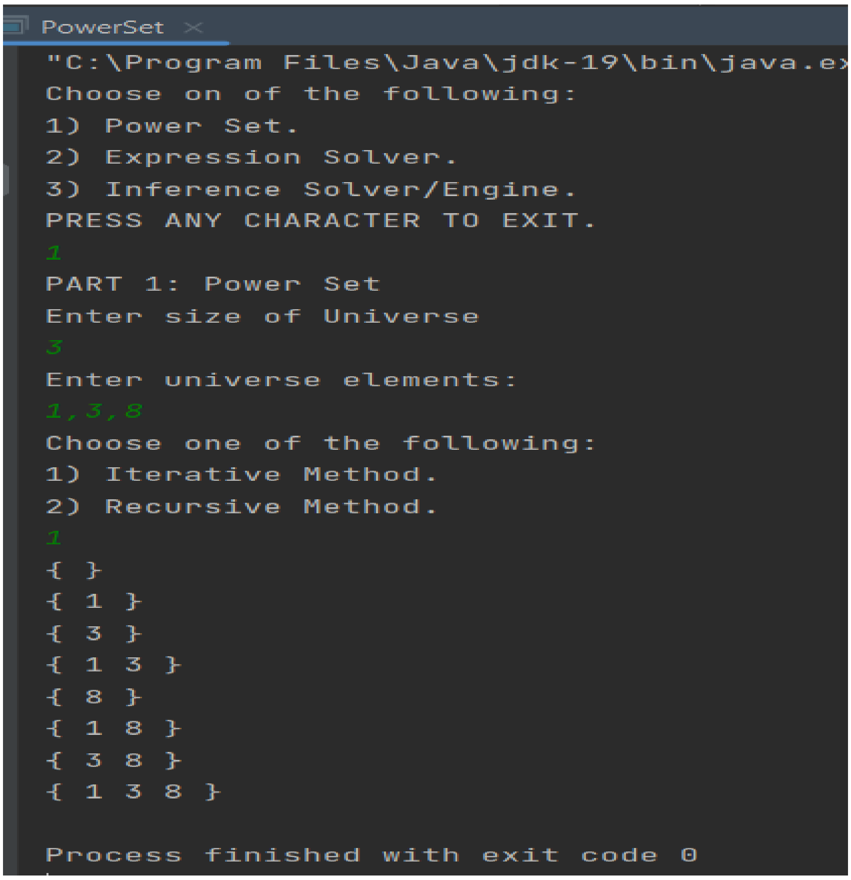
PART (3) : Inference Engine :

- ArrayList of type Expression for expressions.
- ArrayList of type InferenceRule for inference rules.

➢ Sample runs and different test cases:

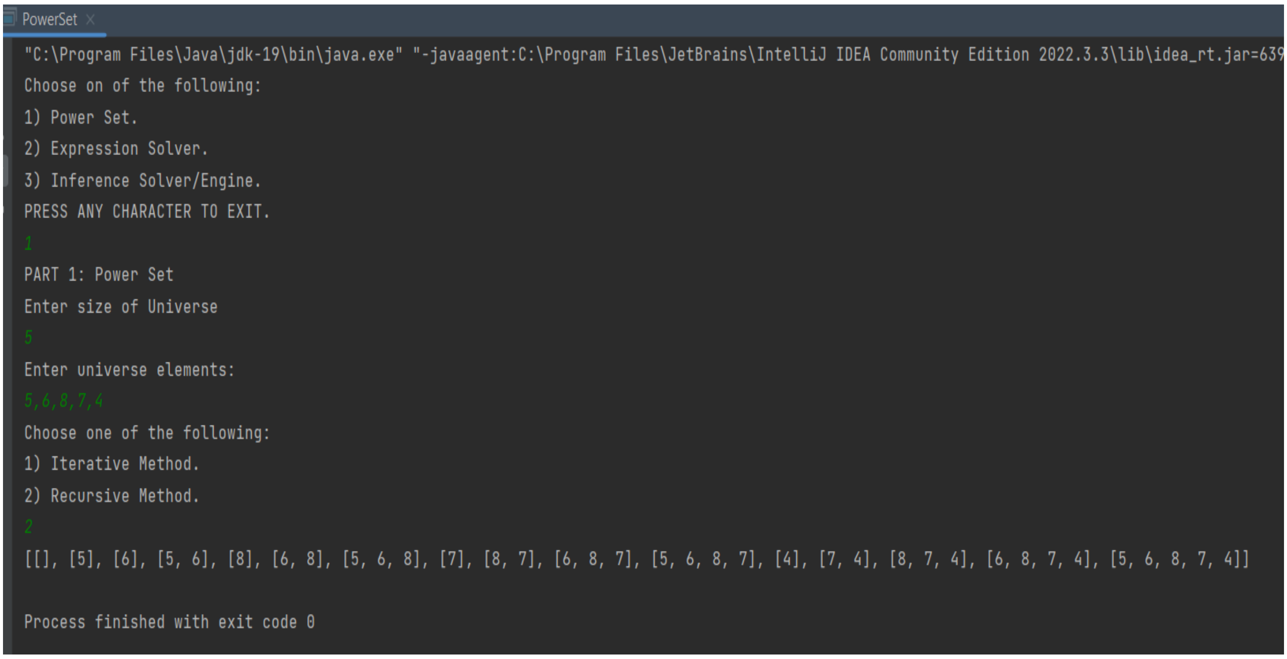
PART (1) : Power Set:

- **Iterative Method:**



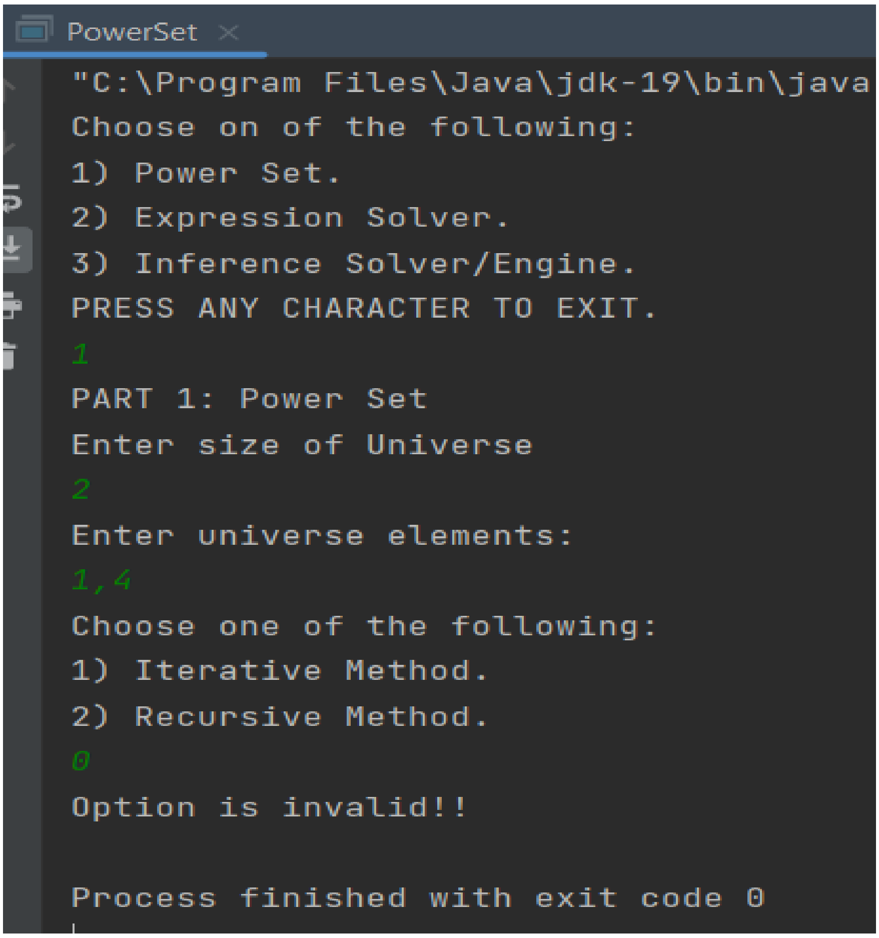
```
"C:\Program Files\Java\jdk-19\bin\java.ex
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
1
PART 1: Power Set
Enter size of Universe
3
Enter universe elements:
1,3,8
Choose one of the following:
1) Iterative Method.
2) Recursive Method.
1
{ }
{ 1 }
{ 3 }
{ 1 3 }
{ 8 }
{ 1 8 }
{ 3 8 }
{ 1 3 8 }

Process finished with exit code 0
```

- **Recursive Method:**



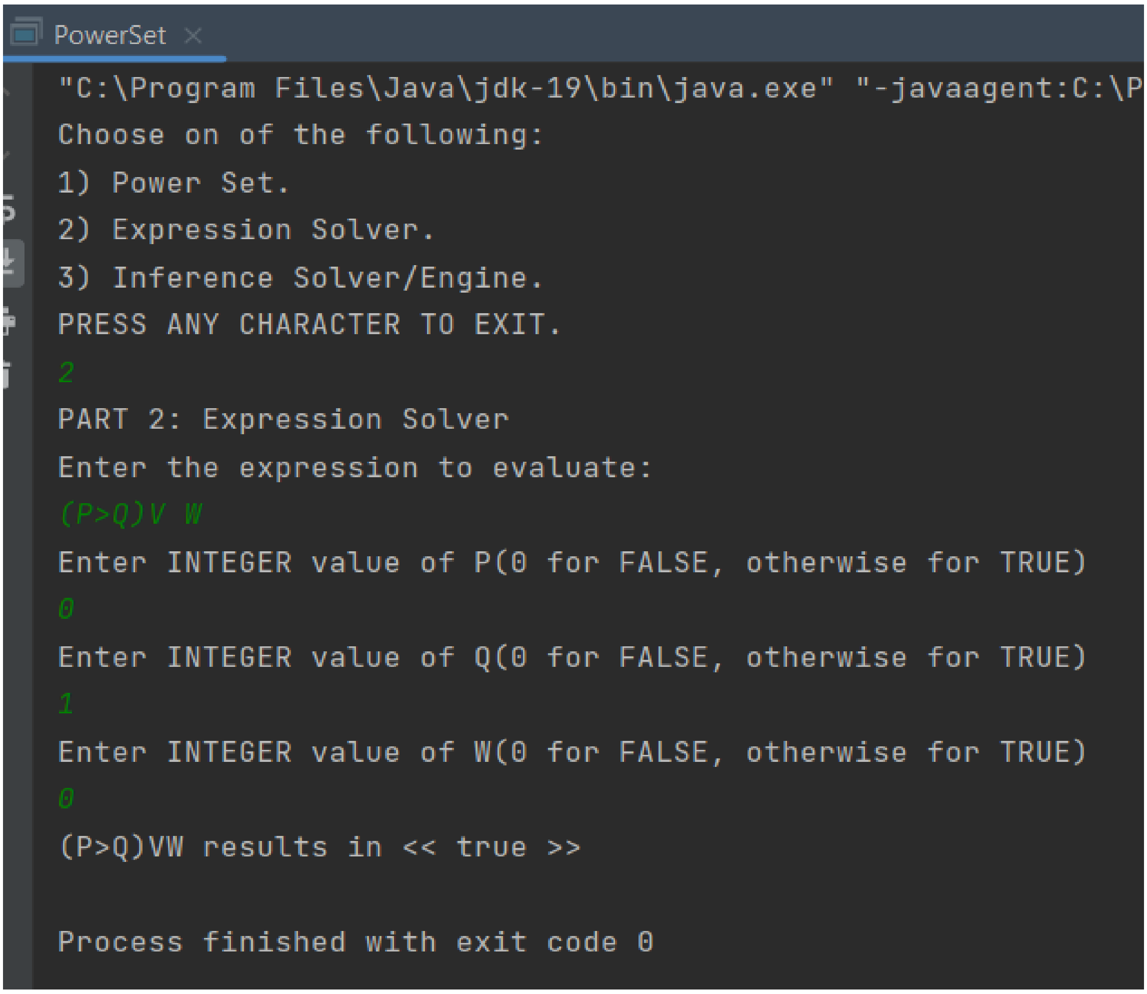
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.3\lib\idea_rt.jar=639
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
1
PART 1: Power Set
Enter size of Universe
5
Enter universe elements:
5,6,8,7,4
Choose one of the following:
1) Iterative Method.
2) Recursive Method.
2
[[], [5], [6], [5, 6], [8], [6, 8], [5, 6, 8], [7], [8, 7], [6, 8, 7], [5, 6, 8, 7], [4], [7, 4], [8, 7, 4], [6, 8, 7, 4], [5, 6, 8, 7, 4]]

Process finished with exit code 0
```

- **Choosing invalid Option:**

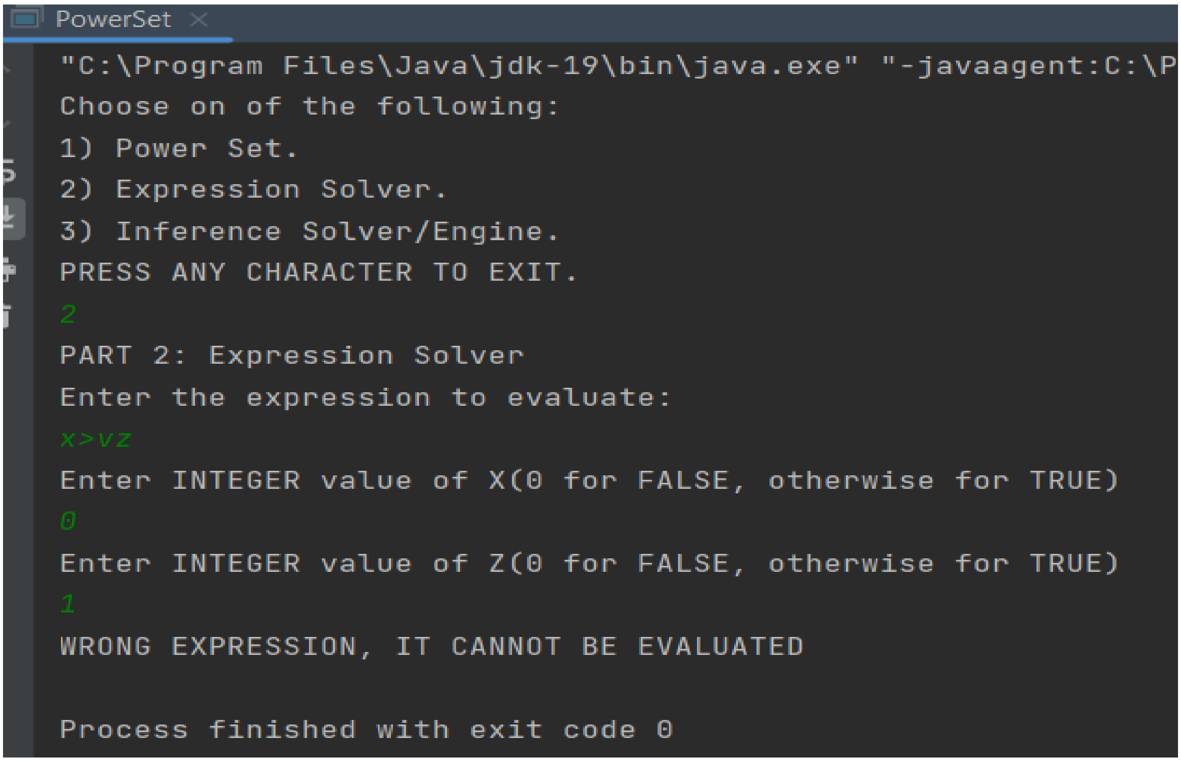


```
"C:\Program Files\Java\jdk-19\bin\java
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
1
PART 1: Power Set
Enter size of Universe
2
Enter universe elements:
1,4
Choose one of the following:
1) Iterative Method.
2) Recursive Method.
0
Option is invalid!!

Process finished with exit code 0
```

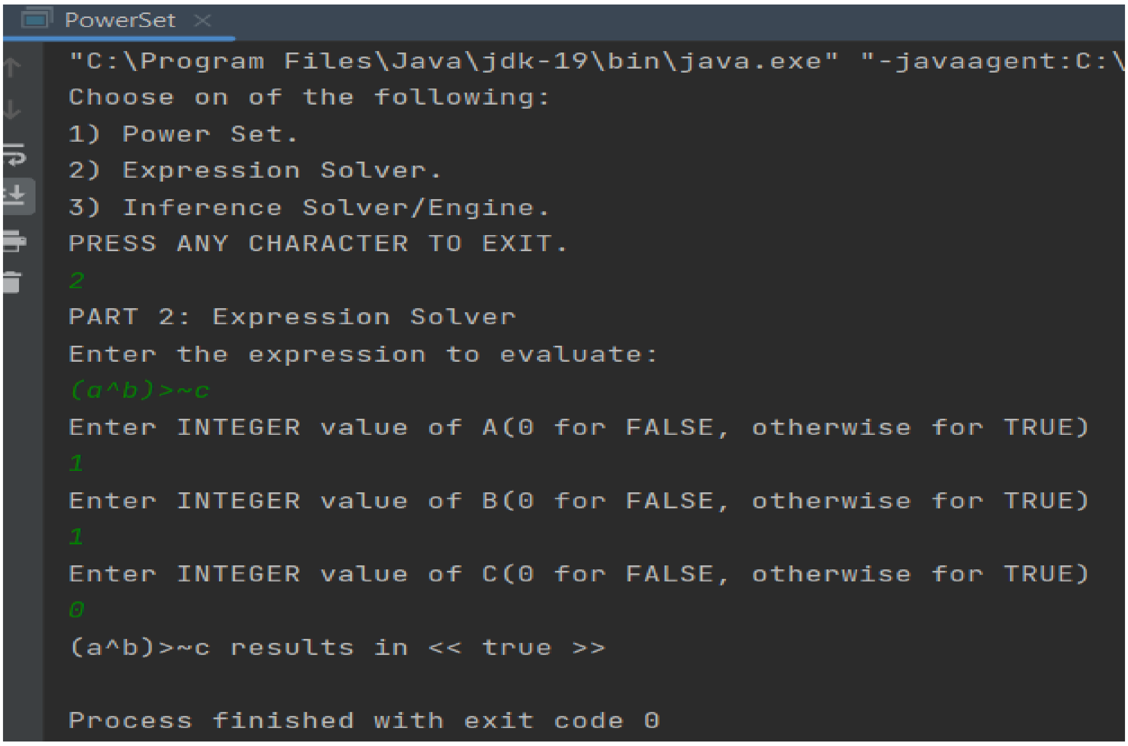
PART (2) : Logical Expressions Solver:



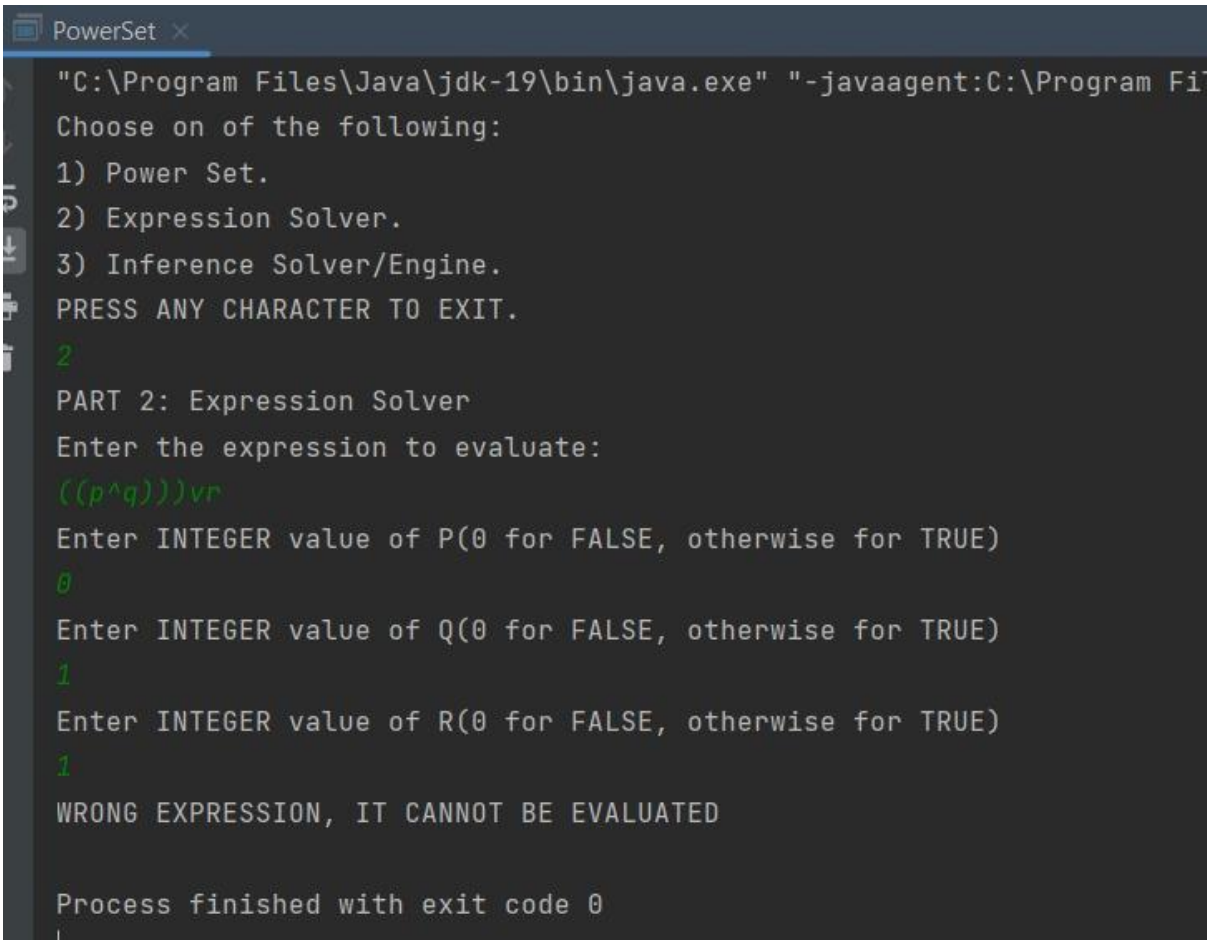
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\P
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
(P>Q)V W
Enter INTEGER value of P(0 for FALSE, otherwise for TRUE)
0
Enter INTEGER value of Q(0 for FALSE, otherwise for TRUE)
1
Enter INTEGER value of W(0 for FALSE, otherwise for TRUE)
0
(P>Q)VW results in << true >>

Process finished with exit code 0
```



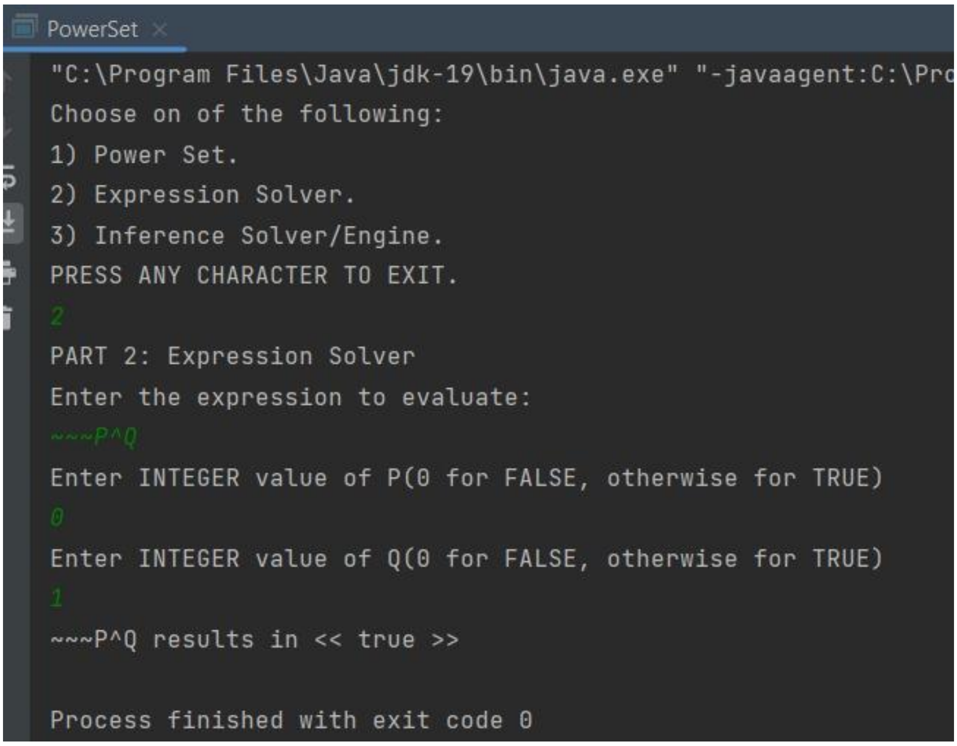
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\P
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
x>vz
Enter INTEGER value of X(0 for FALSE, otherwise for TRUE)
0
Enter INTEGER value of Z(0 for FALSE, otherwise for TRUE)
1
WRONG EXPRESSION, IT CANNOT BE EVALUATED

Process finished with exit code 0
```



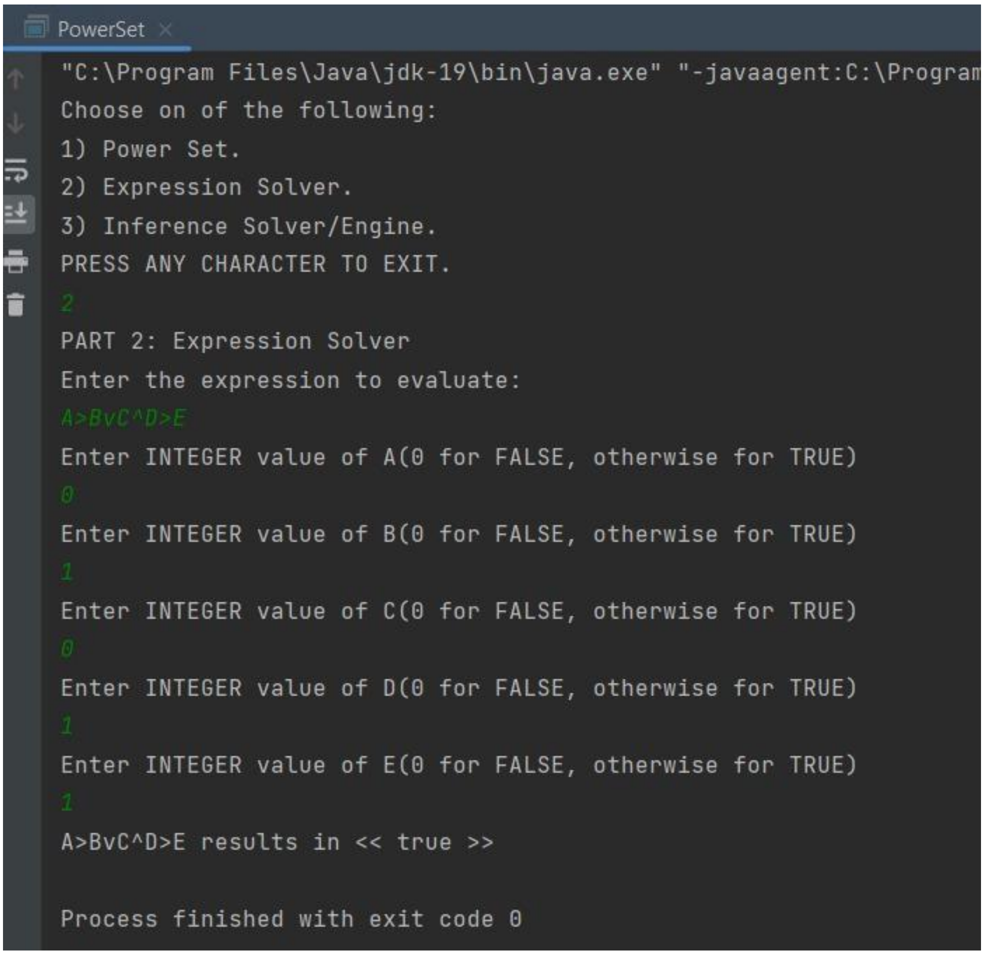
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
(a^b)>~c
Enter INTEGER value of A(0 for FALSE, otherwise for TRUE)
1
Enter INTEGER value of B(0 for FALSE, otherwise for TRUE)
1
Enter INTEGER value of C(0 for FALSE, otherwise for TRUE)
0
(a^b)>~c results in << true >>

Process finished with exit code 0
```



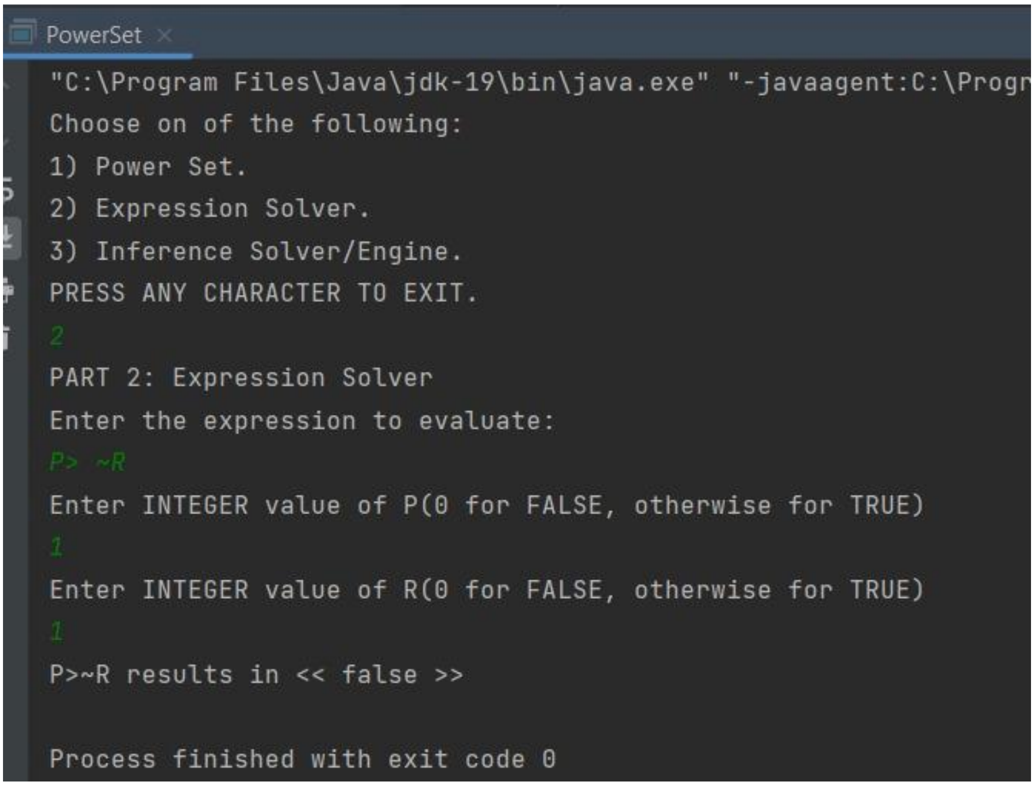
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Fi
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
((p^q)))vr
Enter INTEGER value of P(0 for FALSE, otherwise for TRUE)
0
Enter INTEGER value of Q(0 for FALSE, otherwise for TRUE)
1
Enter INTEGER value of R(0 for FALSE, otherwise for TRUE)
1
WRONG EXPRESSION, IT CANNOT BE EVALUATED

Process finished with exit code 0
```



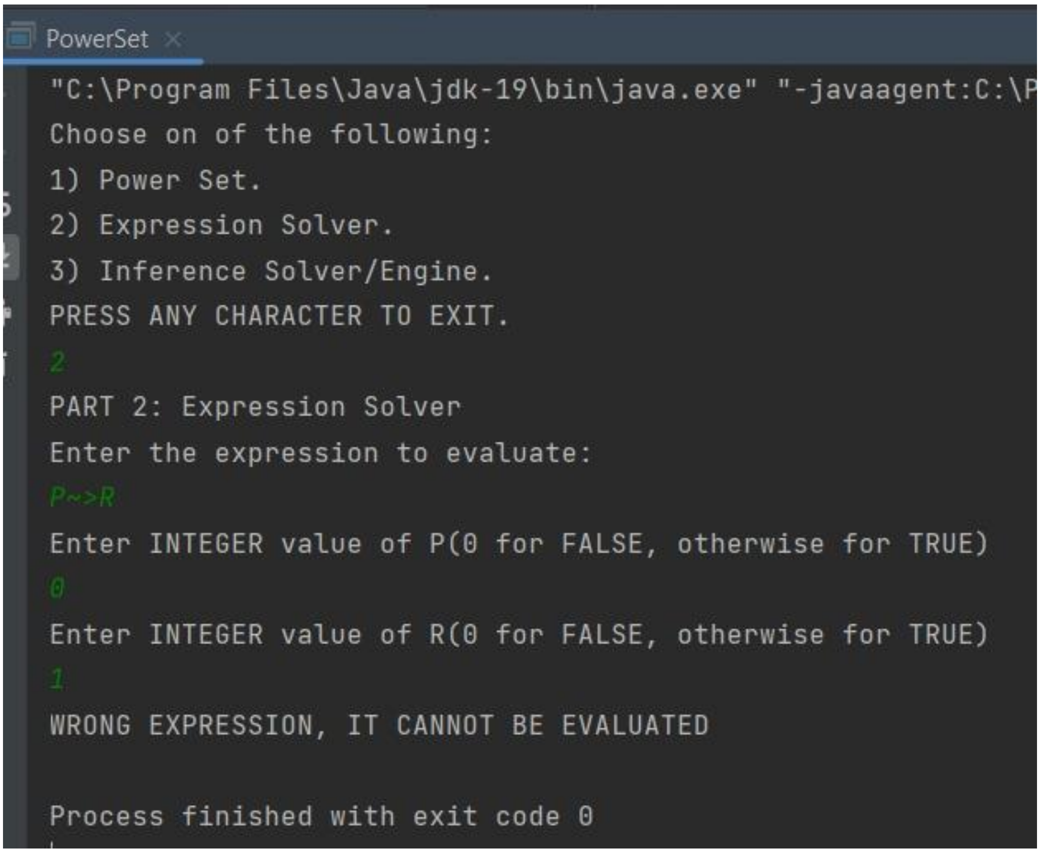
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Pro
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
~~~P^Q
Enter INTEGER value of P(0 for FALSE, otherwise for TRUE)
0
Enter INTEGER value of Q(0 for FALSE, otherwise for TRUE)
1
~~~P^Q results in << true >>

Process finished with exit code 0
```



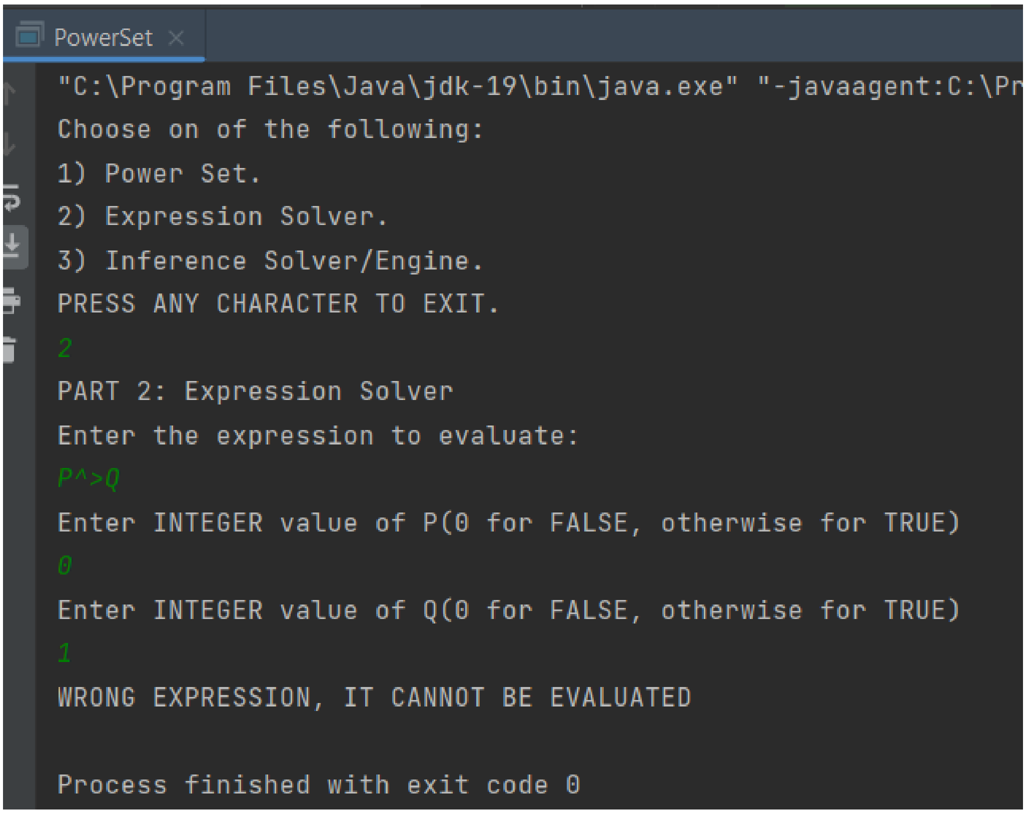
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
A>BvC^D>E
Enter INTEGER value of A(0 for FALSE, otherwise for TRUE)
0
Enter INTEGER value of B(0 for FALSE, otherwise for TRUE)
1
Enter INTEGER value of C(0 for FALSE, otherwise for TRUE)
0
Enter INTEGER value of D(0 for FALSE, otherwise for TRUE)
1
Enter INTEGER value of E(0 for FALSE, otherwise for TRUE)
1
A>BvC^D>E results in << true >>

Process finished with exit code 0
```



```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Progr
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
P> ~R
Enter INTEGER value of P(0 for FALSE, otherwise for TRUE)
1
Enter INTEGER value of R(0 for FALSE, otherwise for TRUE)
1
P>~R results in << false >>

Process finished with exit code 0
```

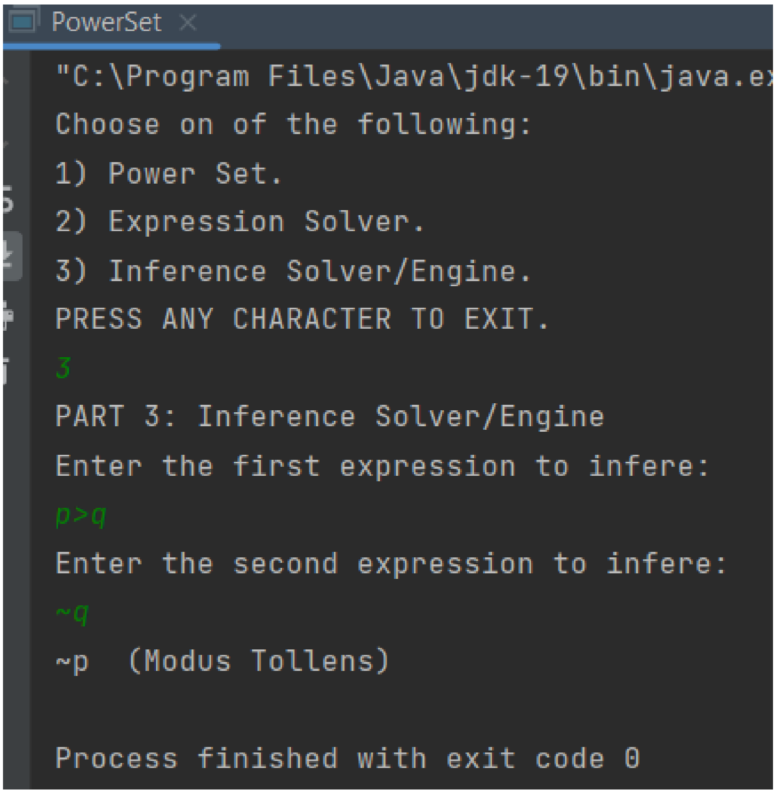


```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\P
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
P~>R
Enter INTEGER value of P(0 for FALSE, otherwise for TRUE)
0
Enter INTEGER value of R(0 for FALSE, otherwise for TRUE)
1
WRONG EXPRESSION, IT CANNOT BE EVALUATED

Process finished with exit code 0
```

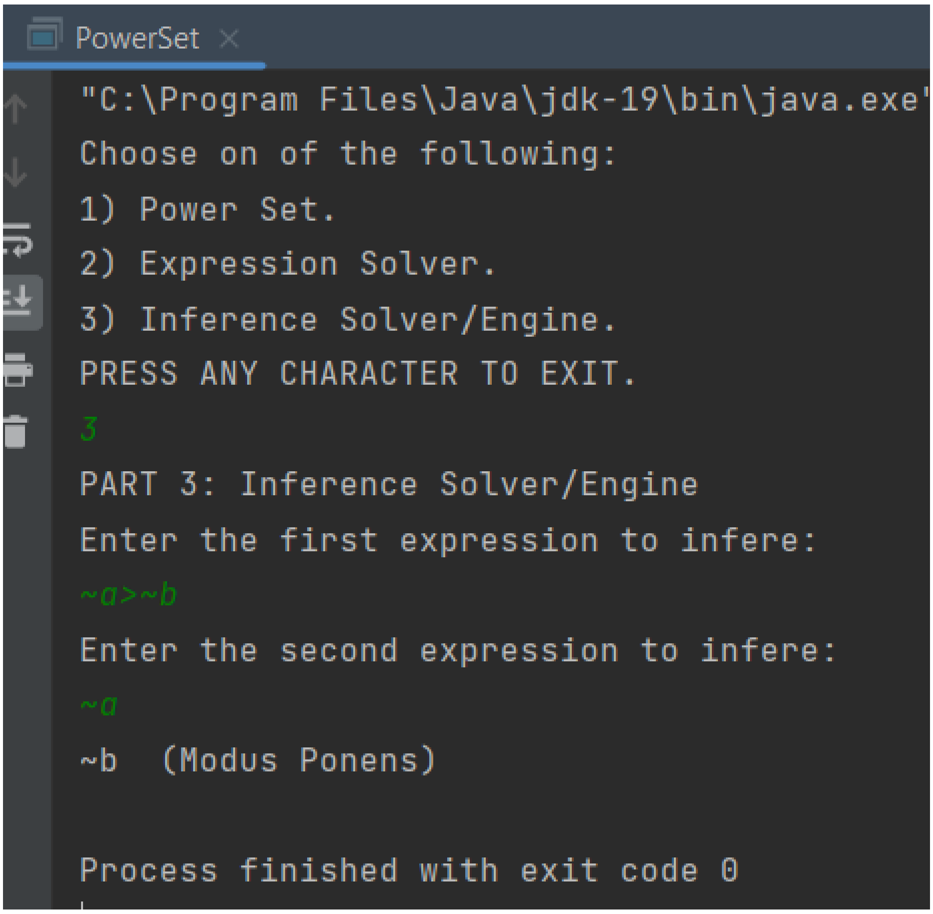


```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Pr
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
2
PART 2: Expression Solver
Enter the expression to evaluate:
P^>Q
Enter INTEGER value of P(0 for FALSE, otherwise for TRUE)
0
Enter INTEGER value of Q(0 for FALSE, otherwise for TRUE)
1
WRONG EXPRESSION, IT CANNOT BE EVALUATED

Process finished with exit code 0
```

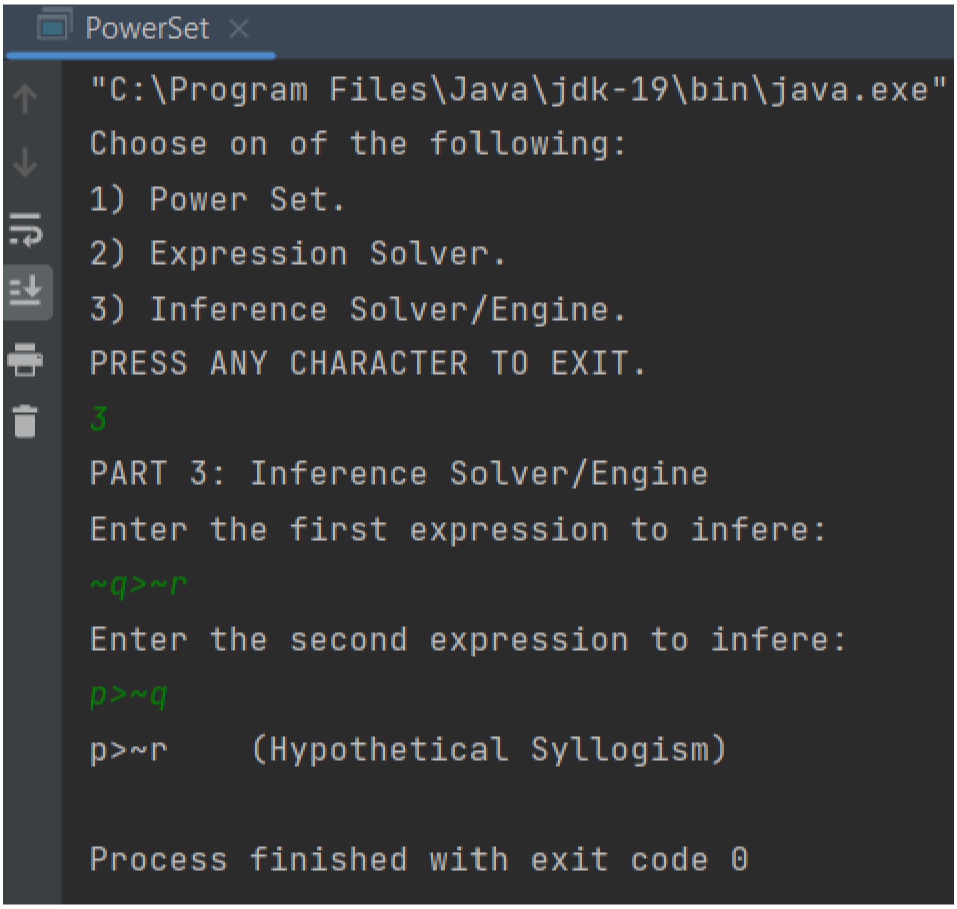
PART (3) : Inference Engine :



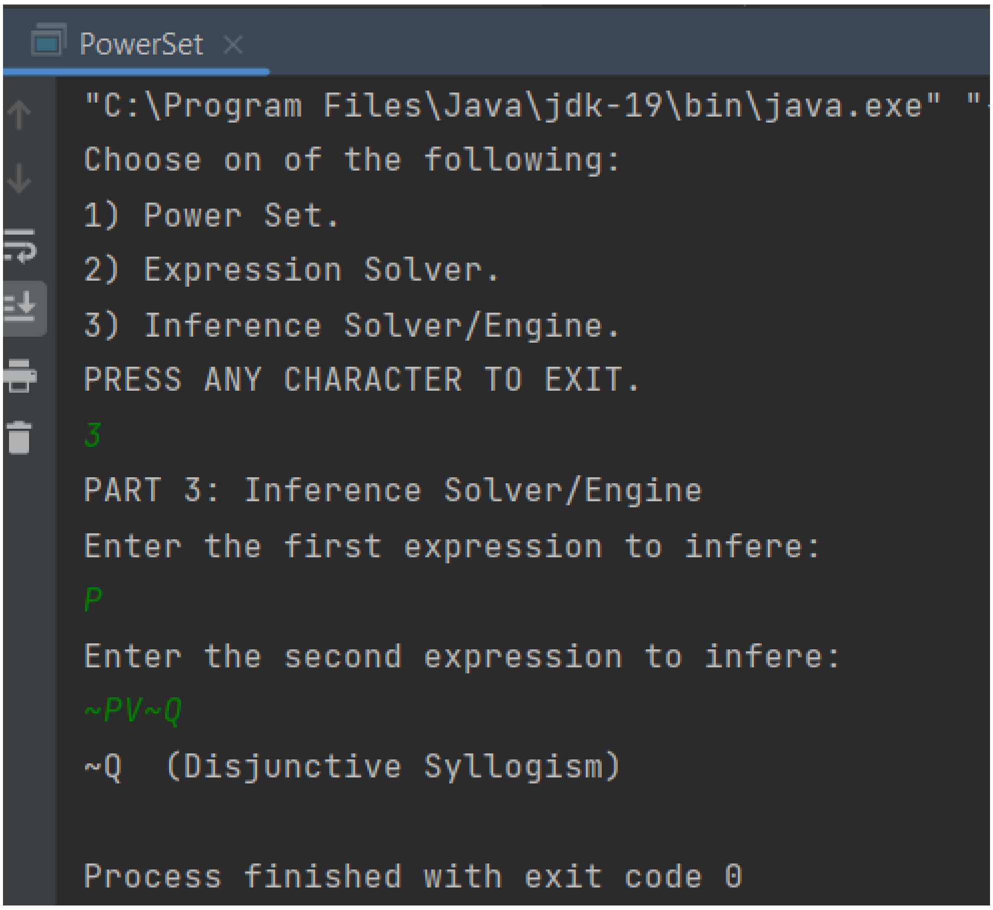
```
"C:\Program Files\Java\jdk-19\bin\java.ex
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
3
PART 3: Inference Solver/Engine
Enter the first expression to infere:
p>q
Enter the second expression to infere:
~q
~p  (Modus Tollens)

Process finished with exit code 0
```



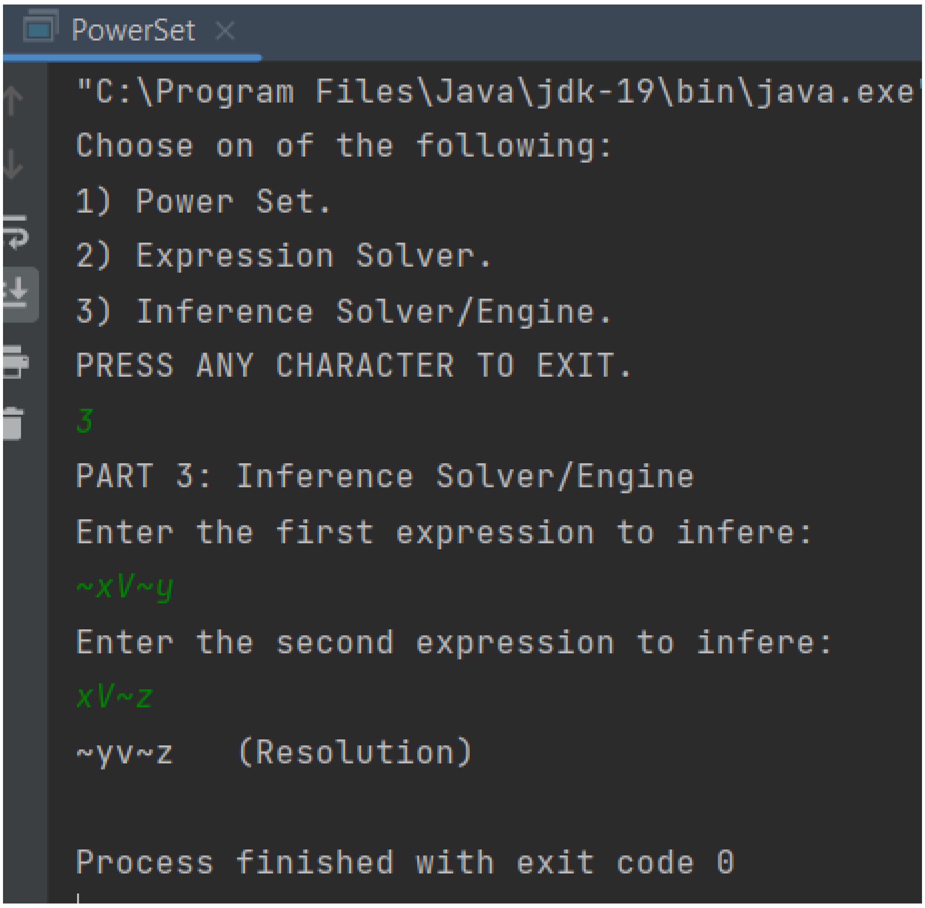
```
"C:\Program Files\Java\jdk-19\bin\java.exe'
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
3
PART 3: Inference Solver/Engine
Enter the first expression to infere:
~a>~b
Enter the second expression to infere:
~a
~b  (Modus Ponens)

Process finished with exit code 0
```



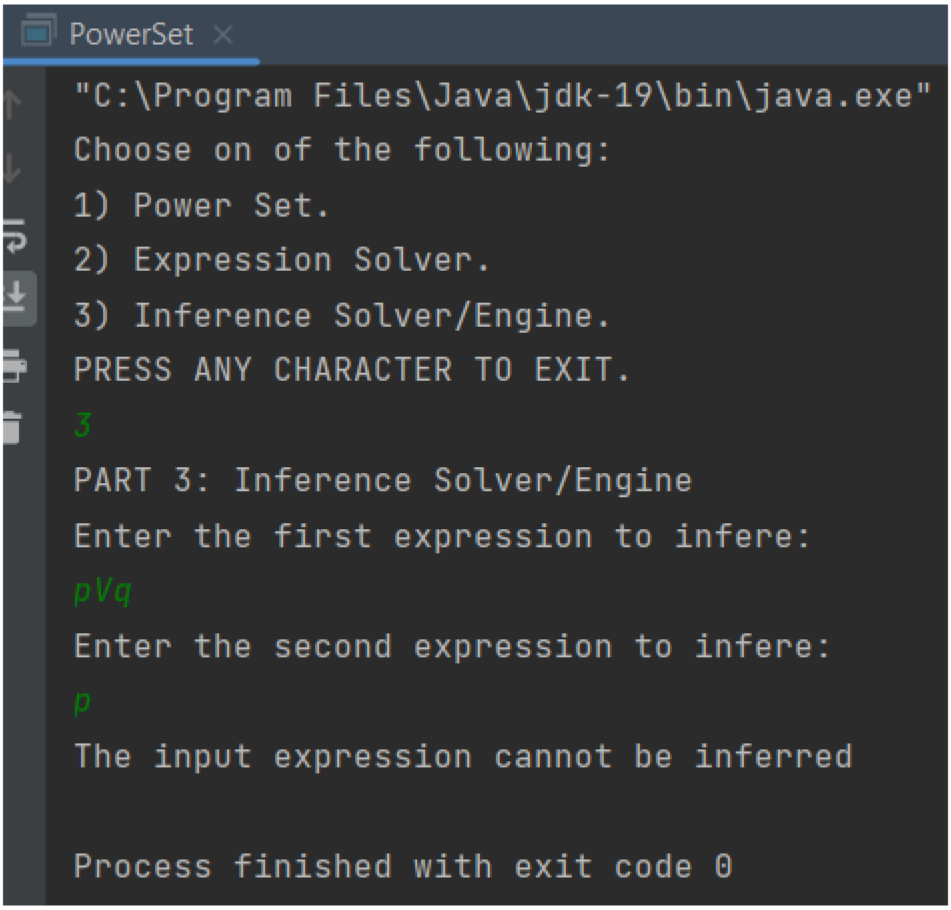
```
"C:\Program Files\Java\jdk-19\bin\java.exe"
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
3
PART 3: Inference Solver/Engine
Enter the first expression to infere:
~q>~r
Enter the second expression to infere:
p>~q
p>~r    (Hypothetical Syllogism)

Process finished with exit code 0
```



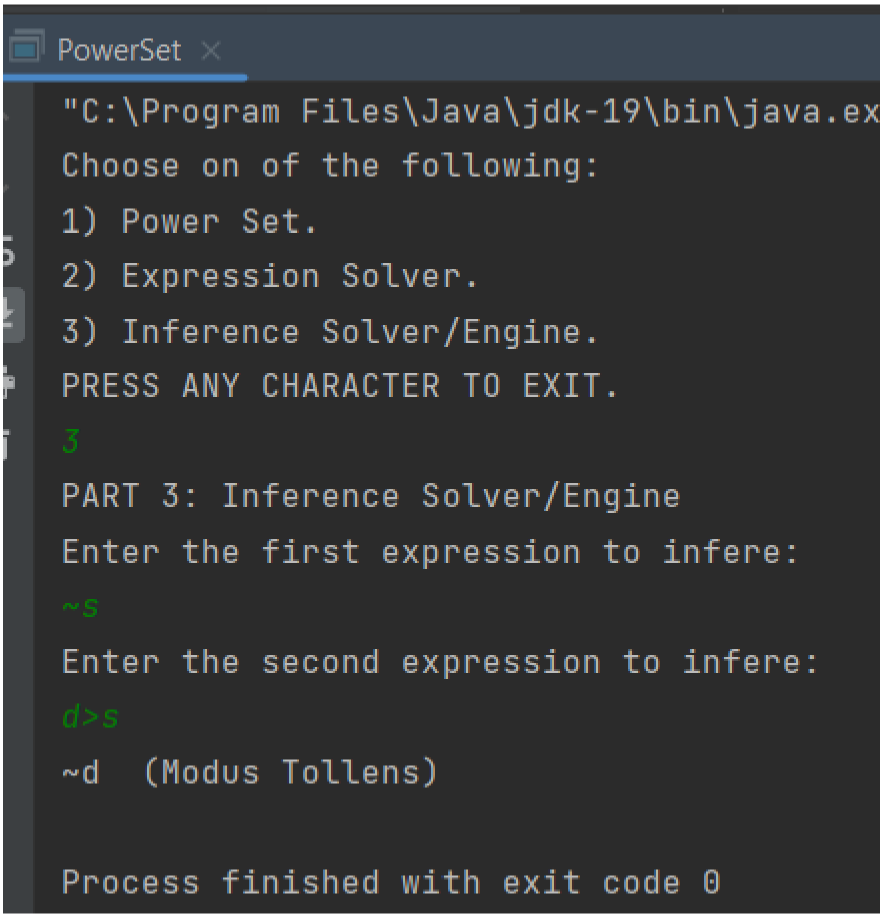
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
3
PART 3: Inference Solver/Engine
Enter the first expression to infere:
P
Enter the second expression to infere:
~PV~Q
~Q  (Disjunctive Syllogism)

Process finished with exit code 0
```



```
"C:\Program Files\Java\jdk-19\bin\java.exe
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
3
PART 3: Inference Solver/Engine
Enter the first expression to infere:
~xV~y
Enter the second expression to infere:
xV~z
~yv~z   (Resolution)

Process finished with exit code 0
```



```
"C:\Program Files\Java\jdk-19\bin\java.exe"
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
3
PART 3: Inference Solver/Engine
Enter the first expression to infere:
pVq
Enter the second expression to infere:
p
The input expression cannot be inferred

Process finished with exit code 0
```



```
"C:\Program Files\Java\jdk-19\bin\java.ex
Choose on of the following:
1) Power Set.
2) Expression Solver.
3) Inference Solver/Engine.
PRESS ANY CHARACTER TO EXIT.
3
PART 3: Inference Solver/Engine
Enter the first expression to infere:
~s
Enter the second expression to infere:
d>s
~d  (Modus Tollens)

Process finished with exit code 0
```