Principles of Programming Language

Assignment 3

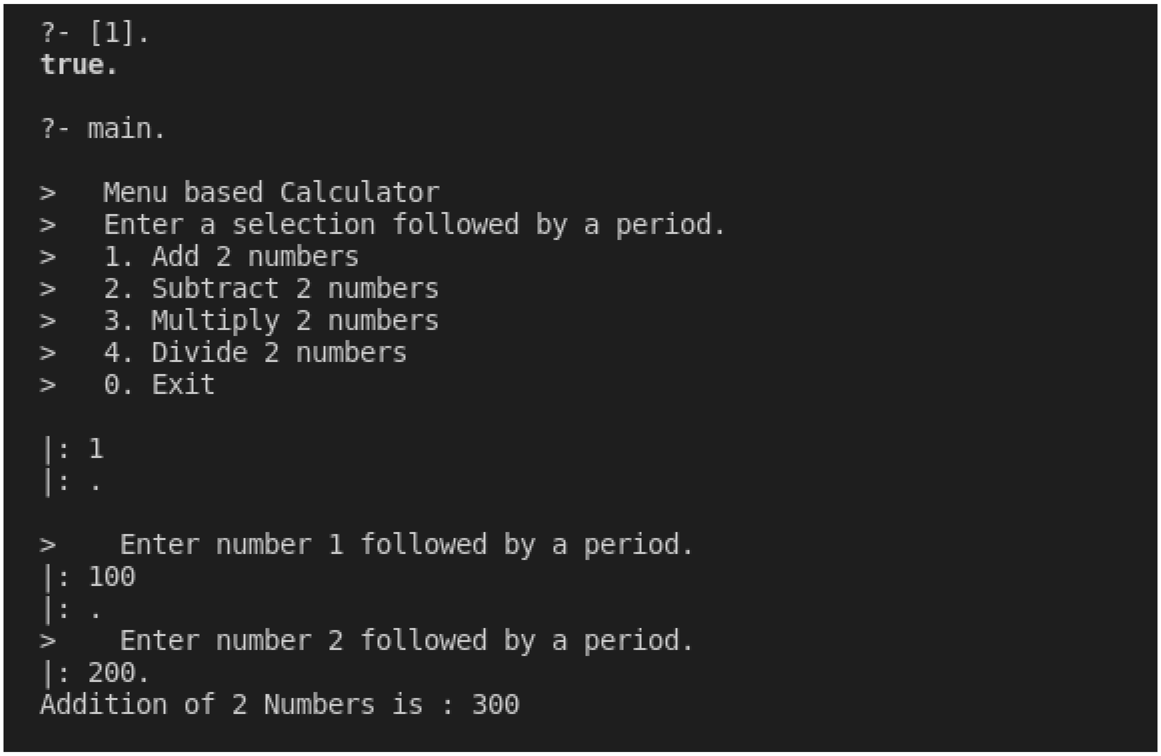
Student Details

Name: Krunal Rank
Adm No: U18CO081

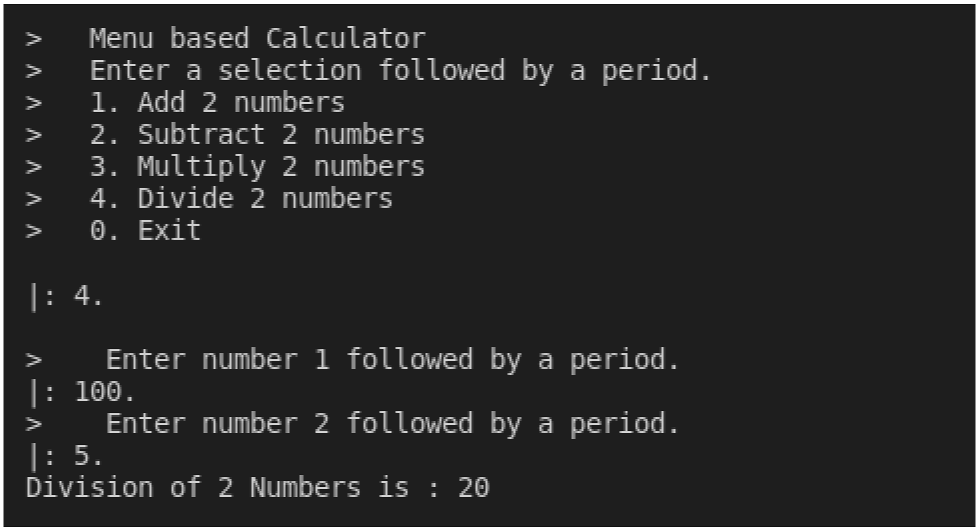
1

```
read_nums(Num1,Num2) :-
   nl,
   write('>    Enter number 1 followed by a period.'),nl,
   read(Num1),
   write('>    Enter number 2 followed by a period.'),nl,
   read(Num2).


run_opt(1) :-
   read_nums(Num1,Num2),
   write('Addition of 2 Numbers is : '),
   Sum is Num1 + Num2,
   write(Sum),nl.
run_opt(2) :-
   read_nums(Num1,Num2),
   write('Subtraction of 2 Numbers is : '),
   Diff is Num1 - Num2,
   write(Diff),nl.
run_opt(3) :-
   read_nums(Num1,Num2),
   write('Multiplication of 2 Numbers is : '),
   Mul is Num1*Num2,
   write(Mul),nl.
run_opt(4) :-
   read_nums(Num1,Num2),
   Num2 =\= 0,
   write('Division of 2 Numbers is : '),
   Div is Num1/Num2,
   write(Div),nl.

run_opt(0) :- write('Bye'), nl, halt.
run_opt(_) :- write('Invalid option'), nl.
```

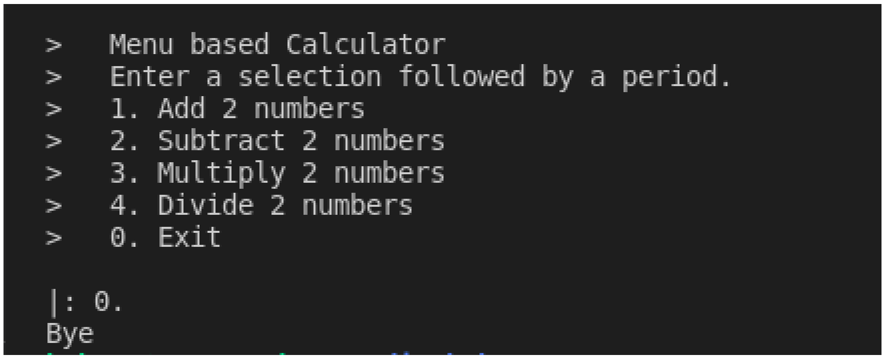
```
main :-
   nl,
   write('>   Menu based Calculator'), nl,
   write('>   Enter a selection followed by a period.'), nl,
   write('>   1. Add 2 numbers'), nl,
   write('>   2. Subtract 2 numbers'), nl,
   write('>   3. Multiply 2 numbers'), nl,
   write('>   4. Divide 2 numbers'), nl,
   write('>   0. Exit'), nl, nl,
   read(Choice),
   run_opt(Choice), main.
```



```
?- [1].
true.

?- main.

>   Menu based Calculator
>   Enter a selection followed by a period.
>   1. Add 2 numbers
>   2. Subtract 2 numbers
>   3. Multiply 2 numbers
>   4. Divide 2 numbers
>   0. Exit

|: 1
|: .

>    Enter number 1 followed by a period.
|: 100
|: .
>    Enter number 2 followed by a period.
|: 200.
Addition of 2 Numbers is : 300
```



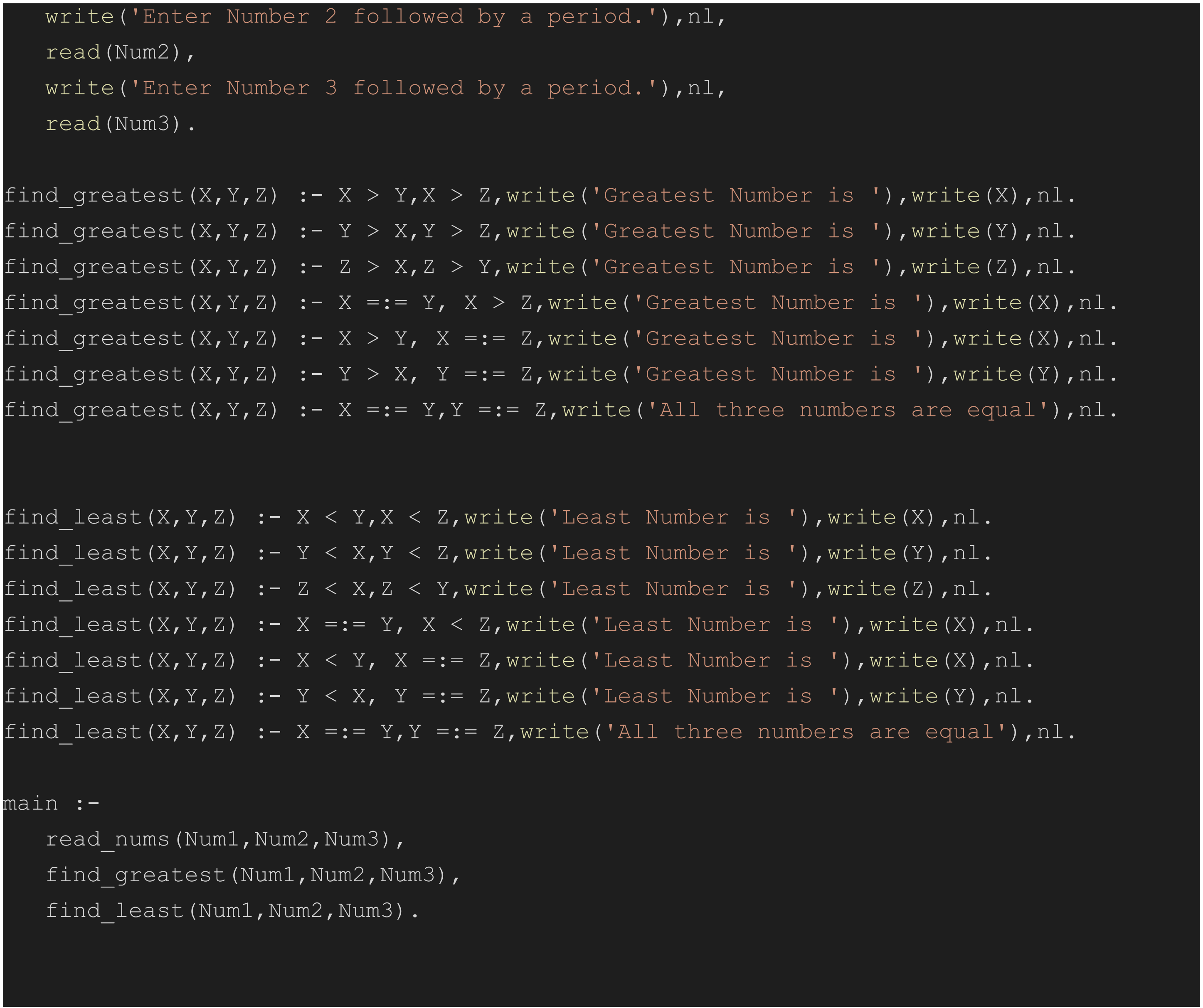
```
>   Menu based Calculator
>   Enter a selection followed by a period.
>   1. Add 2 numbers
>   2. Subtract 2 numbers
>   3. Multiply 2 numbers
>   4. Divide 2 numbers
>   0. Exit

|: 4.

>    Enter number 1 followed by a period.
|: 100.
>    Enter number 2 followed by a period.
|: 5.
Division of 2 Numbers is : 20
```



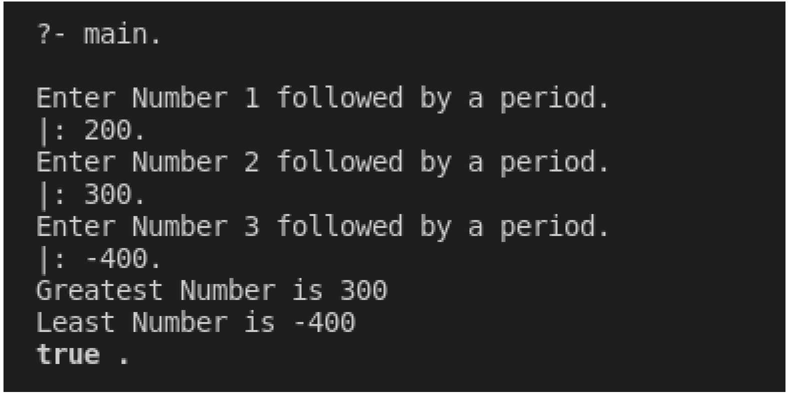
```
>   Menu based Calculator
>   Enter a selection followed by a period.
>   1. Add 2 numbers
>   2. Subtract 2 numbers
>   3. Multiply 2 numbers
>   4. Divide 2 numbers
>   0. Exit

|: 0.
Bye
```

2

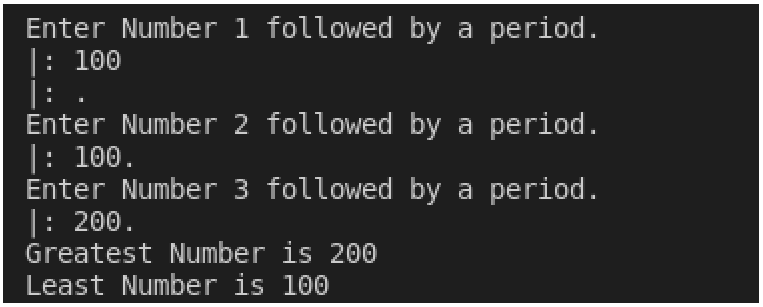
```
read_nums(Num1,Num2,Num3) :-
   nl,
   write('Enter Number 1 followed by a period.'),nl,
   read(Num1),
```



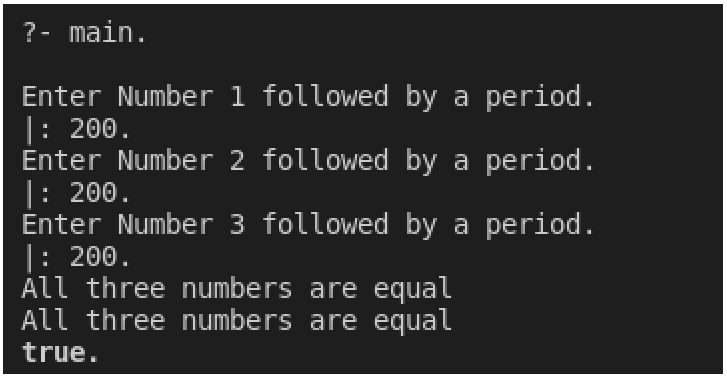
```
   write('Enter Number 2 followed by a period.'),nl,
   read(Num2),
   write('Enter Number 3 followed by a period.'),nl,
   read(Num3).

find_greatest(X,Y,Z) :- X > Y,X > Z,write('Greatest Number is '),write(X),nl.
find_greatest(X,Y,Z) :- Y > X,Y > Z,write('Greatest Number is '),write(Y),nl.
find_greatest(X,Y,Z) :- Z > X,Z > Y,write('Greatest Number is '),write(Z),nl.
find_greatest(X,Y,Z) :- X =:= Y, X > Z,write('Greatest Number is '),write(X),nl.
find_greatest(X,Y,Z) :- X > Y, X =:= Z,write('Greatest Number is '),write(X),nl.
find_greatest(X,Y,Z) :- Y > X, Y =:= Z,write('Greatest Number is '),write(Y),nl.
find_greatest(X,Y,Z) :- X =:= Y,Y =:= Z,write('All three numbers are equal'),nl.


find_least(X,Y,Z) :- X < Y,X < Z,write('Least Number is '),write(X),nl.
find_least(X,Y,Z) :- Y < X,Y < Z,write('Least Number is '),write(Y),nl.
find_least(X,Y,Z) :- Z < X,Z < Y,write('Least Number is '),write(Z),nl.
find_least(X,Y,Z) :- X =:= Y, X < Z,write('Least Number is '),write(X),nl.
find_least(X,Y,Z) :- X < Y, X =:= Z,write('Least Number is '),write(X),nl.
find_least(X,Y,Z) :- Y < X, Y =:= Z,write('Least Number is '),write(Y),nl.
find_least(X,Y,Z) :- X =:= Y,Y =:= Z,write('All three numbers are equal'),nl.

main :-
   read_nums(Num1,Num2,Num3),
   find_greatest(Num1,Num2,Num3),
   find_least(Num1,Num2,Num3).
```



```
?- main.

Enter Number 1 followed by a period.
|: 200.
Enter Number 2 followed by a period.
|: 300.
Enter Number 3 followed by a period.
|: -400.
Greatest Number is 300
Least Number is -400
true .
```

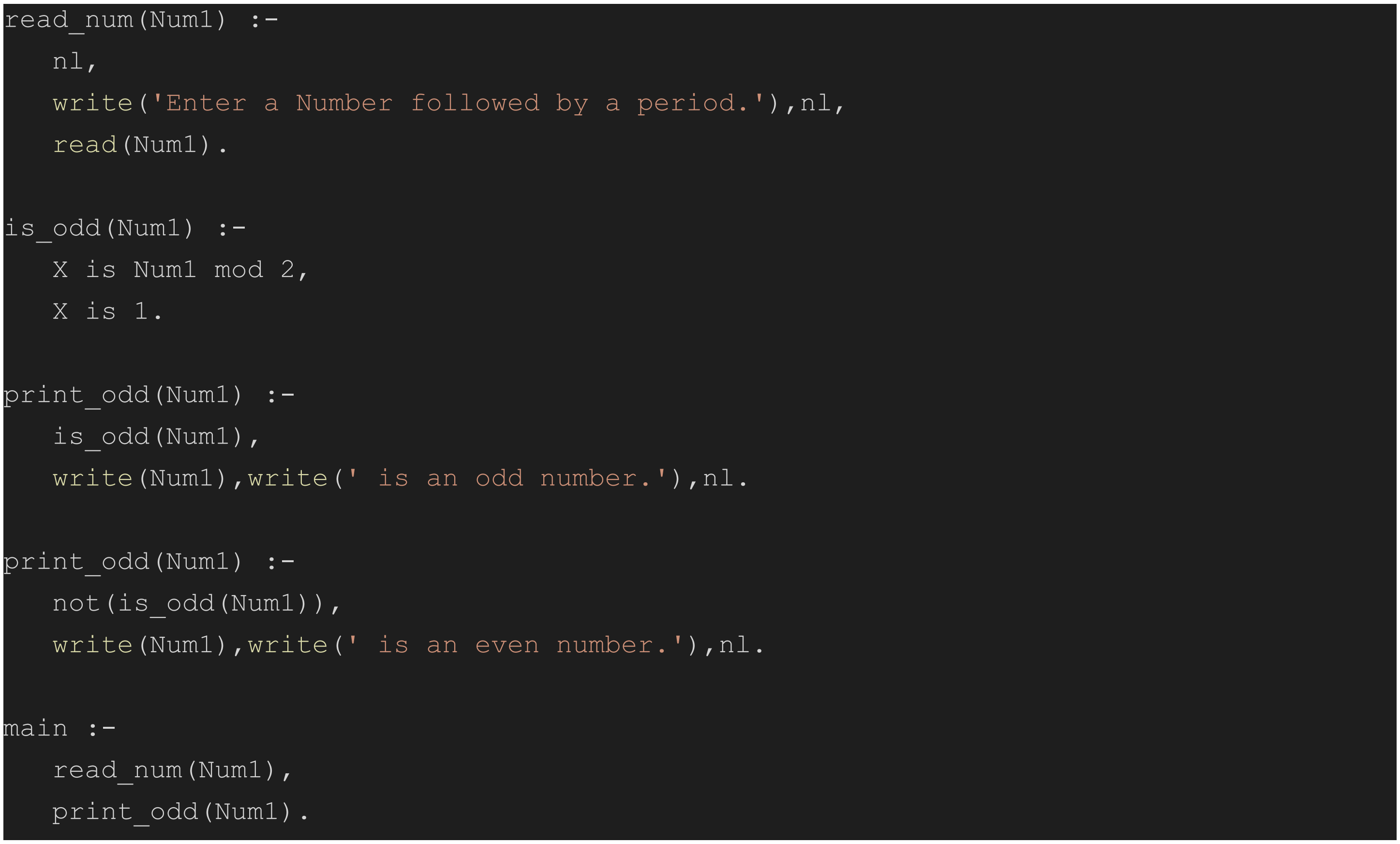


```
Enter Number 1 followed by a period.
|: 100
|: .
Enter Number 2 followed by a period.
|: 100.
Enter Number 3 followed by a period.
|: 200.
Greatest Number is 200
Least Number is 100
```

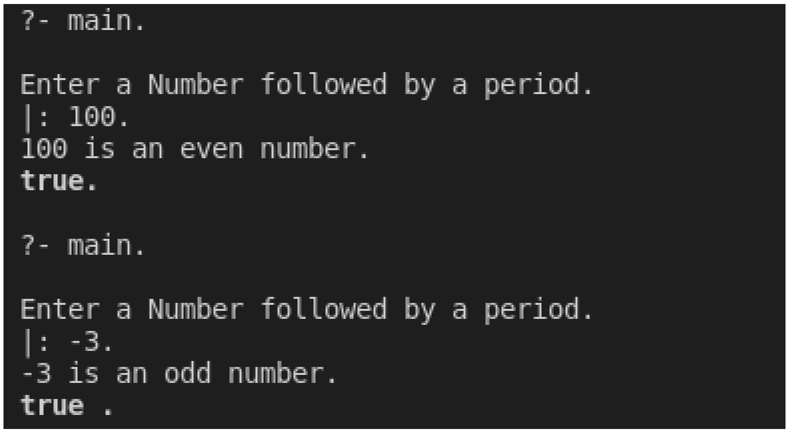


```
?- main.

Enter Number 1 followed by a period.
|: 200.
Enter Number 2 followed by a period.
|: 200.
Enter Number 3 followed by a period.
|: 200.
All three numbers are equal
All three numbers are equal
true.
```

3



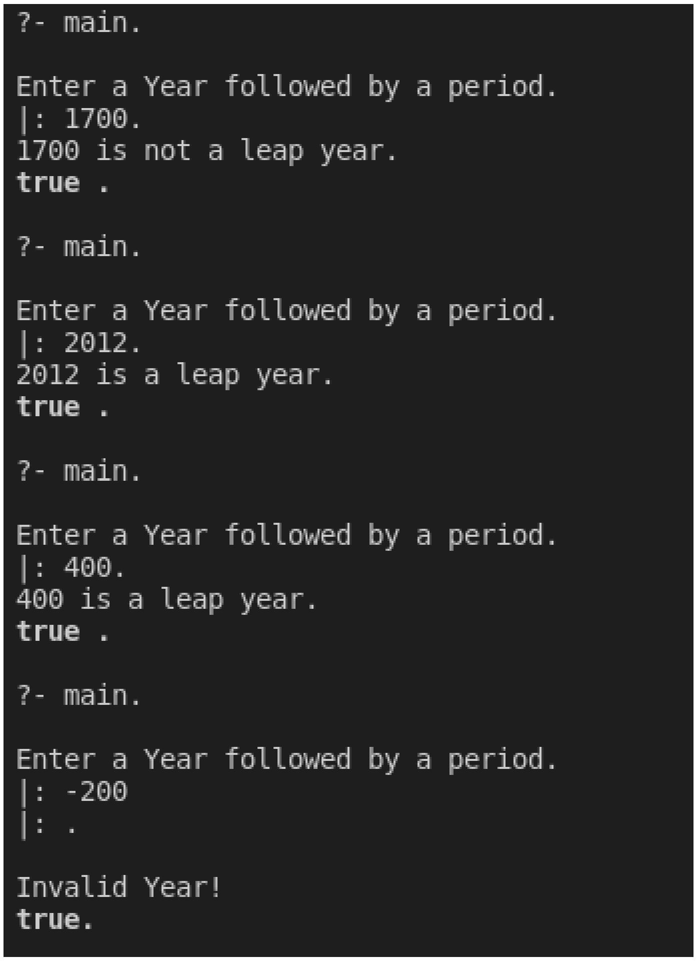
```
read_num(Num1) :-
   nl,
   write('Enter a Number followed by a period.'),nl,
   read(Num1).

is_odd(Num1) :-
   X is Num1 mod 2,
   X is 1.

print_odd(Num1) :-
   is_odd(Num1),
   write(Num1),write(' is an odd number.'),nl.

print_odd(Num1) :-
   not(is_odd(Num1)),
   write(Num1),write(' is an even number.'),nl.

main :-
   read_num(Num1),
   print_odd(Num1).
```



```
?- main.

Enter a Number followed by a period.
|: 100.
100 is an even number.
true.

?- main.

Enter a Number followed by a period.
|: -3.
-3 is an odd number.
true .
```

4.

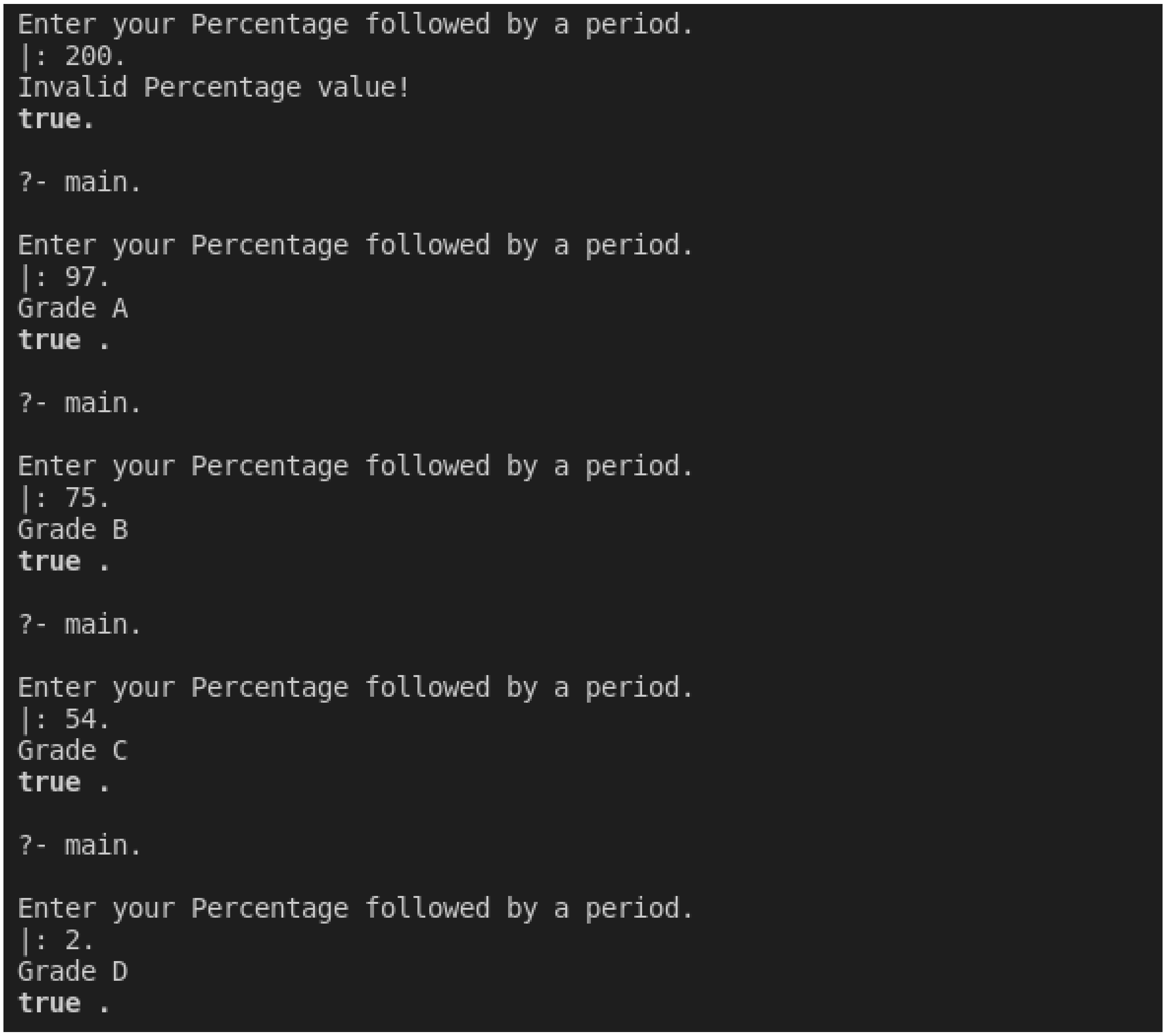
```
read_num(Num1) :-
   nl,
   write('Enter a Year followed by a period.'),nl,
   read(Num1).

is_leap_year(Num1) :- 0 is Num1 mod 4, not(0 is Num1 mod 100).
is_leap_year(Num1) :- 0 is Num1 mod 100, 0 is Num1 mod 400.

print_leap(Num1) :- Num1 >= 0, is_leap_year(Num1), write(Num1),write(' is a leap
year.'),nl.
print_leap(Num1) :- Num1 >= 0, not(is_leap_year(Num1)), write(Num1),write(' is not a
leap year.'),nl.

print_leap(_) :- nl, write('Invalid Year!'),nl.

main :-
   read_num(Num1),
   print_leap(Num1).
```



```
?- main.

Enter a Year followed by a period.
|: 1700.
1700 is not a leap year.
true .

?- main.

Enter a Year followed by a period.
|: 2012.
2012 is a leap year.
true .

?- main.

Enter a Year followed by a period.
|: 400.
400 is a leap year.
true .

?- main.

Enter a Year followed by a period.
|: -200
|: .

Invalid Year!
true.
```

5

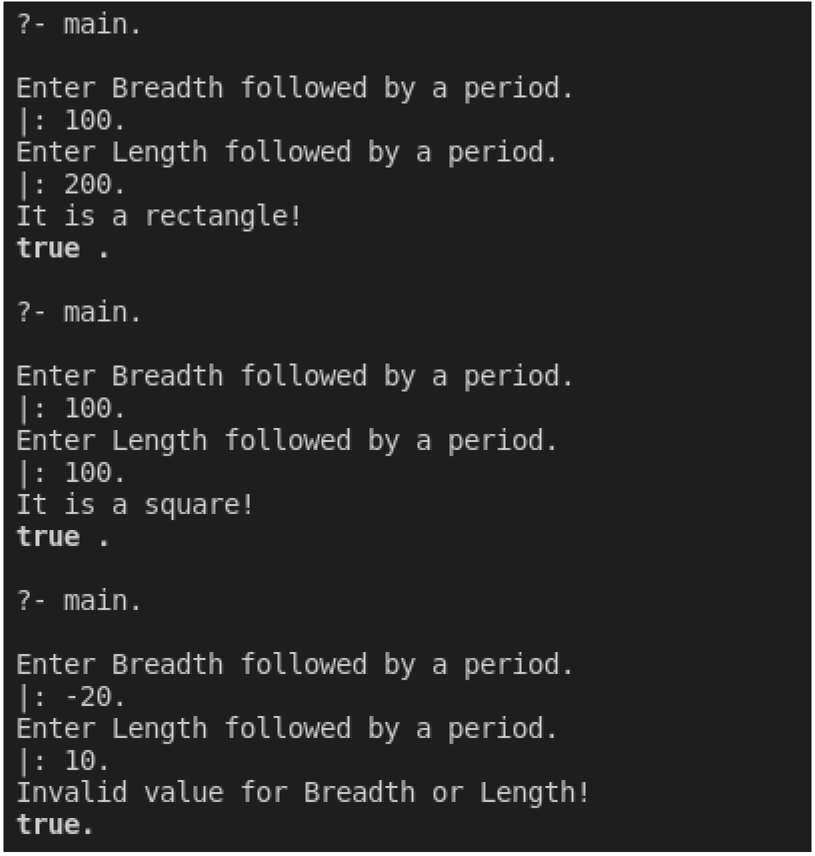
```
read_num(Num1) :-
   nl,
   write('Enter your Percentage followed by a period.'),nl,
   read(Num1).

print_grade(Num1) :- Num1 =< 100, Num1 >= 80, write('Grade A'),nl.
print_grade(Num1) :- Num1 < 80, Num1 >= 60, write('Grade B'),nl.
print_grade(Num1) :- Num1 < 60, Num1 >= 35, write('Grade C'),nl.
print_grade(Num1) :- Num1 < 35, Num1 >= 0, write('Grade D'),nl.
print_grade(_) :- write('Invalid Percentage value!'),nl.

main :-
   read_num(Num1),
   print_grade(Num1).
```



```
Enter your Percentage followed by a period.
|: 200.
Invalid Percentage value!
true.

?- main.

Enter your Percentage followed by a period.
|: 97.
Grade A
true .

?- main.

Enter your Percentage followed by a period.
|: 75.
Grade B
true .

?- main.

Enter your Percentage followed by a period.
|: 54.
Grade C
true .

?- main.

Enter your Percentage followed by a period.
|: 2.
Grade D
true .
```

6

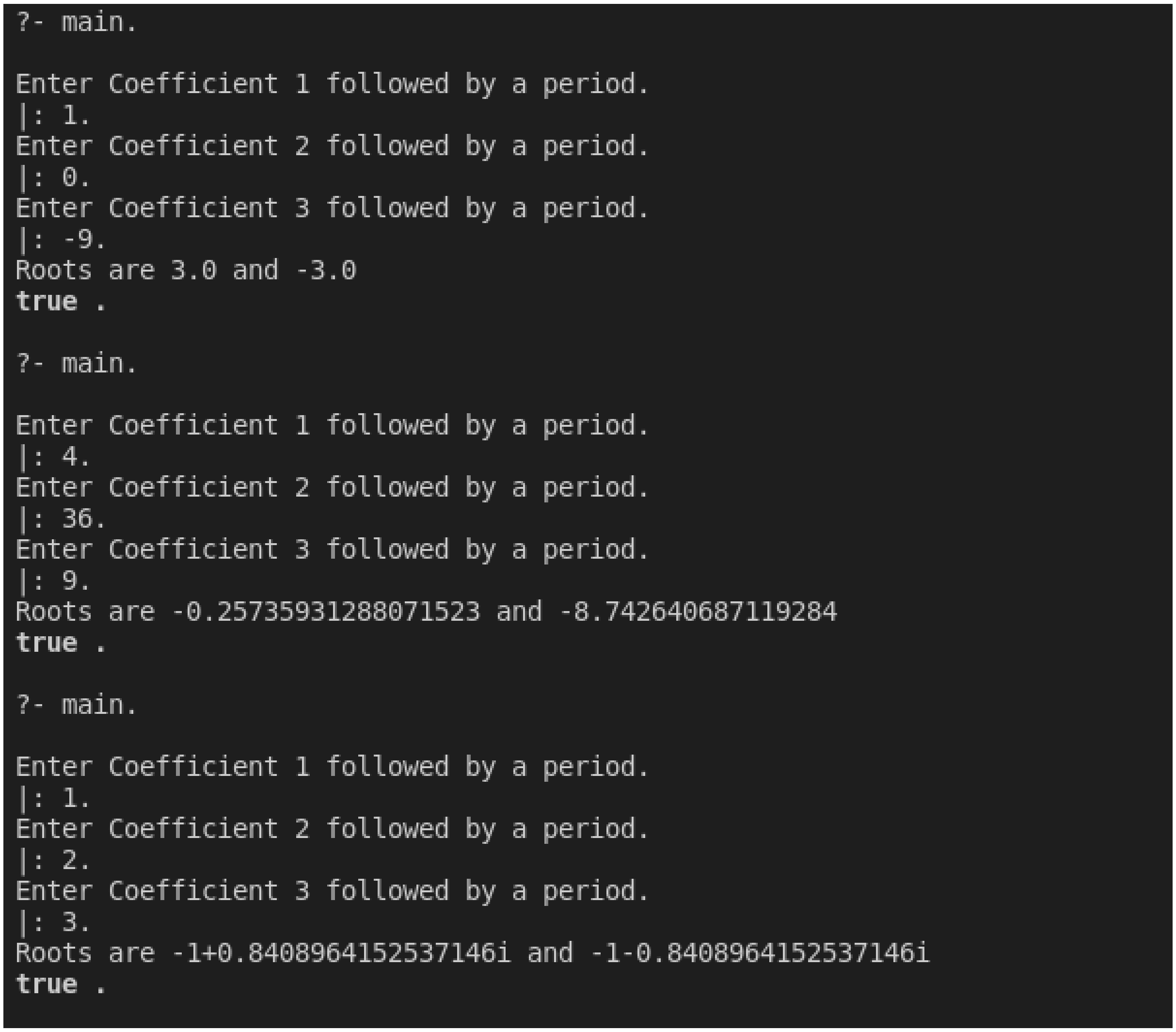
```
read_nums(Num1,Num2) :-
   nl,
   write('Enter Breadth followed by a period.'),nl,
   read(Num1),
   write('Enter Length followed by a period.'),nl,
   read(Num2).


check_square(Num1,Num2) :- Num1 > 0, Num2 > 0, Num1 =:= Num2, write('It is a
square!'),nl.
check_square(Num1,Num2) :- Num1 > 0, Num2 > 0, Num1 =\= Num2, write('It is a
rectangle!'),nl.
check_square(_,_) :- write('Invalid value for Breadth or Length!'),nl.

main :-
   read_nums(Num1,Num2),
   check_square(Num1,Num2).
```



```
?- main.

Enter Breadth followed by a period.
|: 100.
Enter Length followed by a period.
|: 200.
It is a rectangle!
true .

?- main.

Enter Breadth followed by a period.
|: 100.
Enter Length followed by a period.
|: 100.
It is a square!
true .

?- main.

Enter Breadth followed by a period.
|: -20.
Enter Length followed by a period.
|: 10.
Invalid value for Breadth or Length!
true.
```

7

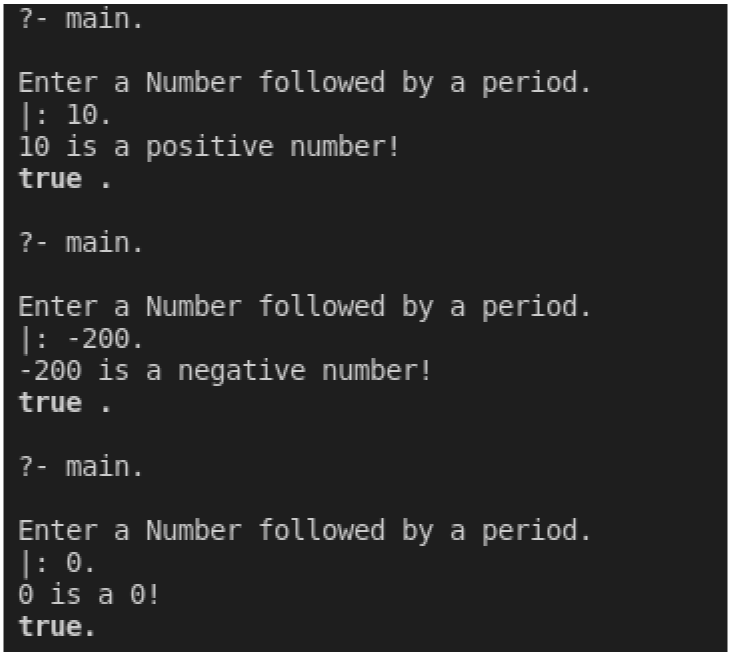
```
read_nums(A,B,C) :-
   nl,
   write('Enter Coefficient 1 followed by a period.'),nl,
   read(A),
   number(A),
   write('Enter Coefficient 2 followed by a period.'),nl,
   read(B),
   number(B),
   write('Enter Coefficient 3 followed by a period.'),nl,
   read(C),
   number(C).

find_roots(A,B,C) :-
   A > 0, D is B^2 - 4*A*C, D > 0,Z is sqrt(D), X1 is (-B + Z)/(2*A), X2 is (-B -
Z)/(2*A),
   write('Roots are '),write(X1),write(' and '),write(X2),nl.

find_roots(A,B,C) :-
   A > 0, D is B^2 - 4*A*C, D =:= 0, X1 is (-B)/(2*A),
   write('Repeated Root is '),write(X1),nl.

find_roots(A,B,C) :-
   A > 0, D is B^2 - 4*A*C, D < 0,D1 is -D,Z is sqrt(D1),Real is -B/(2*A),Imaginary
is sqrt(Z)/(2*A),
   write('Roots are '),
   write(Real),write('+'),write(Imaginary),write('i'),
   write(' and '),
   write(Real),write('-'),write(Imaginary),write('i'),
   nl.

find_roots(_,_,_) :- write('Invalid Coefficients! Coefficient 1 needs to be non
zero.'),nl.

main :-
   read_nums(A,B,C),
   find_roots(A,B,C).
```



```
?- main.

Enter Coefficient 1 followed by a period.
|: 1.
Enter Coefficient 2 followed by a period.
|: 0.
Enter Coefficient 3 followed by a period.
|: -9.
Roots are 3.0 and -3.0
true .

?- main.

Enter Coefficient 1 followed by a period.
|: 4.
Enter Coefficient 2 followed by a period.
|: 36.
Enter Coefficient 3 followed by a period.
|: 9.
Roots are -0.25735931288071523 and -8.742640687119284
true .

?- main.

Enter Coefficient 1 followed by a period.
|: 1.
Enter Coefficient 2 followed by a period.
|: 2.
Enter Coefficient 3 followed by a period.
|: 3.
Roots are -1+0.8408964152537146i and -1-0.8408964152537146i
true .
```

8

```
read_num(Num1) :-
   nl,
   write('Enter a Number followed by a period.'),nl,
   read(Num1).



check(Num1) :- Num1 > 0, write(Num1),write(' is a positive number!'),nl.
check(Num1) :- Num1 < 0, write(Num1),write(' is a negative number!'),nl.
check(Num1) :- Num1 =:= 0, write(Num1),write(' is a 0!'),nl.

main :-
   read_num(Num1),
   check(Num1).
```



```
?- main.

Enter a Number followed by a period.
|: 10.
10 is a positive number!
true .

?- main.

Enter a Number followed by a period.
|: -200.
-200 is a negative number!
true .

?- main.

Enter a Number followed by a period.
|: 0.
0 is a 0!
true.
```