Program 59.Evaluate Boolean Expression
SQL Schema
Table Variables:

Column Name	Type
name	varchar
value	int

name is the primary key for this table.
This table contains the stored variables and their values.

Table Expressions:

Column Name	Type
left_operand	varchar
operator	enum
right_operand	varchar

(left_operand, operator, right_operand) is the primary key for this table.
This table contains a boolean expression that should be evaluated.
operator is an enum that takes one of the values ('<', '>', '=')
The values of left_operand and right_operand are guaranteed to be in the Variables table.

Example 1:

Input: nums = [1,0,0,0,1,0,0,1], k = 2
Output: true
Explanation: Each of the 1s are at least 2 places away from each other.
Program:

```
# Sample data structures to represent the tables
variables = {
   "a": 5,
   "b": 10,
   "c": 15
}

expressions = [
   {"left_operand": "a", "operator": ">", "right_operand": "b"},
   {"left_operand": "b", "operator": "<", "right_operand": "c"},
   {"left_operand": "a", "operator": "=", "right_operand": "a"}
]


def evaluate_expression(variables, expressions):
   results = []

   for expr in expressions:
      left_val = variables[expr["left_operand"]]
      right_val = variables[expr["right_operand"]]
      operator = expr["operator"]

      if operator == ">":
```

```
            result = left_val > right_val
        elif operator == "<":
            result = left_val < right_val
        elif operator == "=":
            result = left_val == right_val
        else:
            raise ValueError(f"Unknown operator {operator}")

        results.append(result)

    return results


# Example usage
results = evaluate_expression(variables, expressions)
print(results)  # Output: [False, True, True]
```

Output:

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 59.py"
[False, True, True]

Process finished with exit code 0
```

Time complexity:
O(m)