OPTION # 1: QUIZ APP

*You now have the tools to make a **Quiz App***

- strings, ints/floats, booleans, conversions
- user input, variables
- conditionals, logic, nesting

You can choose any topic you would like to quiz the user on. However, it must follow these parameters:

1) At least five (5) questions about the topic
2) Print the user's overall score
3) ***Challenge:*** Print summary of correct/incorrect answers

Examples: School Subjects, Movies, Pop Culture

OPTION # 2: Create Your Own!

You have the option of creating your own program, but it must include the elements we have been discussing thus far. These include:

1) User input; output
2) Variables
3) At least 2 data types (strings, ints/floats, booleans)
4) Conditionals, logic, nesting

There is no minimum number of lines of code, but it should be a program that:

1) Gives the user choice
2) Takes user input
3) Compares user input to a piece of information
4) Gives an output based on the input.

OPTION # 3: ATM Interface

Instructions:

Create a program that will simulate an ATM interface.

1) It should start by asking the user for their starting balance and save it as a number (either float or integer) in a variable. The program should then present the user with three options and allow the user to make a choice:
    - Withdraw
    - Deposit
    - Exit

2) If the user chooses Withdraw, the program should ask the user how much they want to withdraw. Again, converting it to a number. The balance should then be updated by adding the amount to the balance.

3) If the user chooses Deposit, the program should ask the user how much they want to deposit. Again, converting it to a number. The balance should then be updated by subtracting the amount from the balance.

4) If the user chooses Exit, the program does not change the balance.

5) If the user enters an unrecognizable input, the program should say so. The balance should not change here either.

At the end of the program, the program should print out the final balance in a sentence - for example “Your final balance is _____”.

Example Program Outputs:

Sample 1 - Withdraw

```
Enter starting balance: 100
Choose from three options:
1. Withdraw
2. Deposit
3. Exit
Choose [1|2|3]: 1
How much to withdraw? 40
The final balance is 60
```

Sample 2 - Deposit

```
Enter starting balance: 500
Choose from three options:
1. Withdraw
2. Deposit
3. Exit
Choose [1|2|3]: 2
How much to deposit? 350
The final balance is 850
```

Sample 3 - Exit

```
Enter starting balance: 305
Choose from three options:
1. Withdraw
2. Deposit
3. Exit
Choose [1|2|3]: 3
No action taken.
The final balance is 305
```

Sample 4 - Invalid Input

```
Enter starting balance: 401
Choose from three options:
1. Withdraw
2. Deposit
3. Exit
Choose [1|2|3]: lol
Input not recognized.
The final balance is 401
```

Rubric

Points: _____ / 12

	2	**1**	**0**
Functionality	My program works perfectly.	My program is broken somewhere in the middle.	My program doesn't work.
Variables	My variables have logical names and are typed in Camel Case when necessary. The values of my variables are formatted correctly (ex. strings are in quotes).	One of those elements is missing.	No evidence.
Operators	I correctly use operators to compare two pieces of data.		No evidence.
User Input	The program is interactive and the input statements are coded correctly.	One of those elements is missing.	No evidence.
Conditional	My program uses at least two if-then-else statements to evaluate the user input.	I included an if-then statement, but not an "else."	No evidence.
Printing	My program prints statements that appropriately respond to the user input.		My print statements are not appropriate.