1. **Scope:**

Expanding from the stubbing project, my goal for this project is to create an animal fun-fact program. For example, I plan to have a list of about 10 different animals presented on the console, and the user can choose which animal they would like to learn about. Once they input their choice, the screen will display a fun fact about the user-selected animal.

**1.1 Identification:**

This project will be written in C++ within the Visual Studio 2019 IDE. I am utilizing Visual Studio since it is a robust software that allows me to optimally write code in the C++ language.

**1.2 System Overview:**

(Why) This program’s purpose is to provide interesting information regarding a type of animal. (Who) It can be an educational tool to be used by children and adults looking to learn unique facts about different species in the animal kingdom. (What) It can be a quick learning tool to teach people specific data. (When and Where) This program can be used at any date and any location as long as there is access to a computer that has Visual Studio or a similar software installed.

Specific system goals include:

1. Display a list of 10 animals to the user
2. Allow user to select one animal at a time
3. Error checking to make sure the user enters a valid selection each time
4. Display information about the animal the user selected
5. Allow the program to continue running until the user selects the option to exit

**1.3 Document Overview:**

The purpose of this document is to detail information based on a coding project that presents fun facts about specific animals as selected by the user. Other information consists of the language (C++) and software (Visual Studio) it is written in, requirements, designing, testing and testing results, and references used relating to the program.

1. **Referenced Documents:**
2. <https://www.treehugger.com/things-you-didnt-know-about-alpacas-4864274>
3. <https://forum.americanexpedition.us/bobcat-facts-information-and-photos#:~:text=The%20bobcat%20is%20the%20most,size%20of%20a%20domestic%20housecat.&text=After%20a%20gestation%20period%20of,kittens%20that%20already%20have%20spots>.
4. <https://forum.americanexpedition.us/coyote-facts-information-and-photos>
5. <https://aquaworld.com.mx/en/blog/15-fun-facts-about-dolphins>
6. <https://www.discoverwildlife.com/animal-facts/mammals/facts-about-elephants/>
7. <https://facts.net/nature/animals/flamingo-facts>
8. <https://www.dosomething.org/us/facts/11-facts-about-giraffes>
9. <https://www.businessinsider.com/how-hamsters-stuff-so-much-food-in-cheeks-2018-5>
10. <https://www.chinahighlights.com/giant-panda/red-panda-faqs.htm>
11. <https://www.tentree.com/blogs/posts/5-wild-facts-about-the-american-pika-the-cutest-endangered-species>

Project Documentation: <https://github.com/sk-panda/SoftwareDesign-repo/blob/master/stubbingFunFactREADME.txt>

Software file:

<https://github.com/sk-panda/SoftwareDesign-repo/blob/master/stubbingAnimalFunFact.cpp>

1. **Requirements:**

It shall display a list of 10 animals.

It shall allow the user to select one animal at a time.

It shall perform error checking to validate the user enters an appropriate choice.

It shall display the fun fact relating to the animal the user selected.

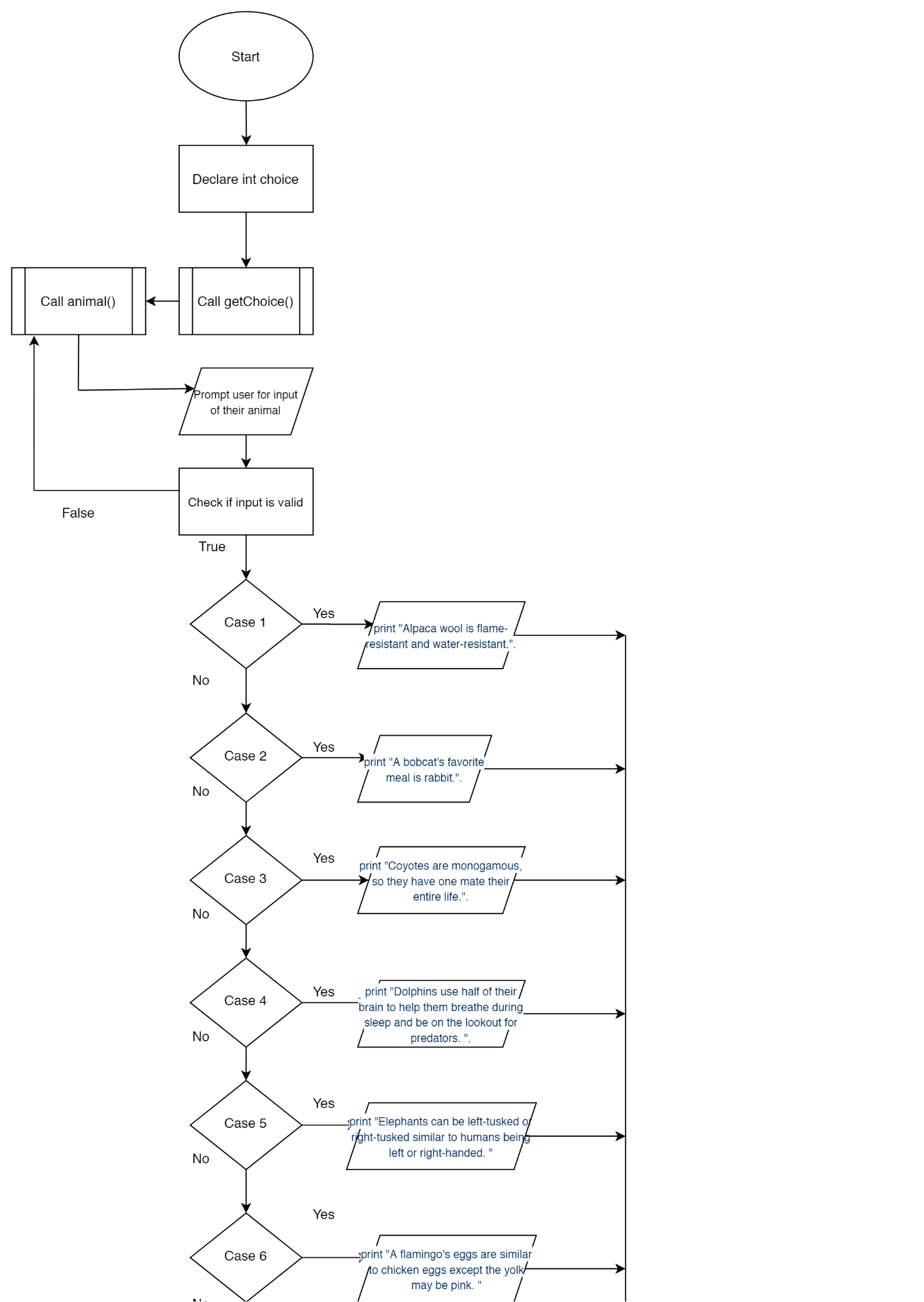
It shall continue running until the user selects the option to terminate the program.

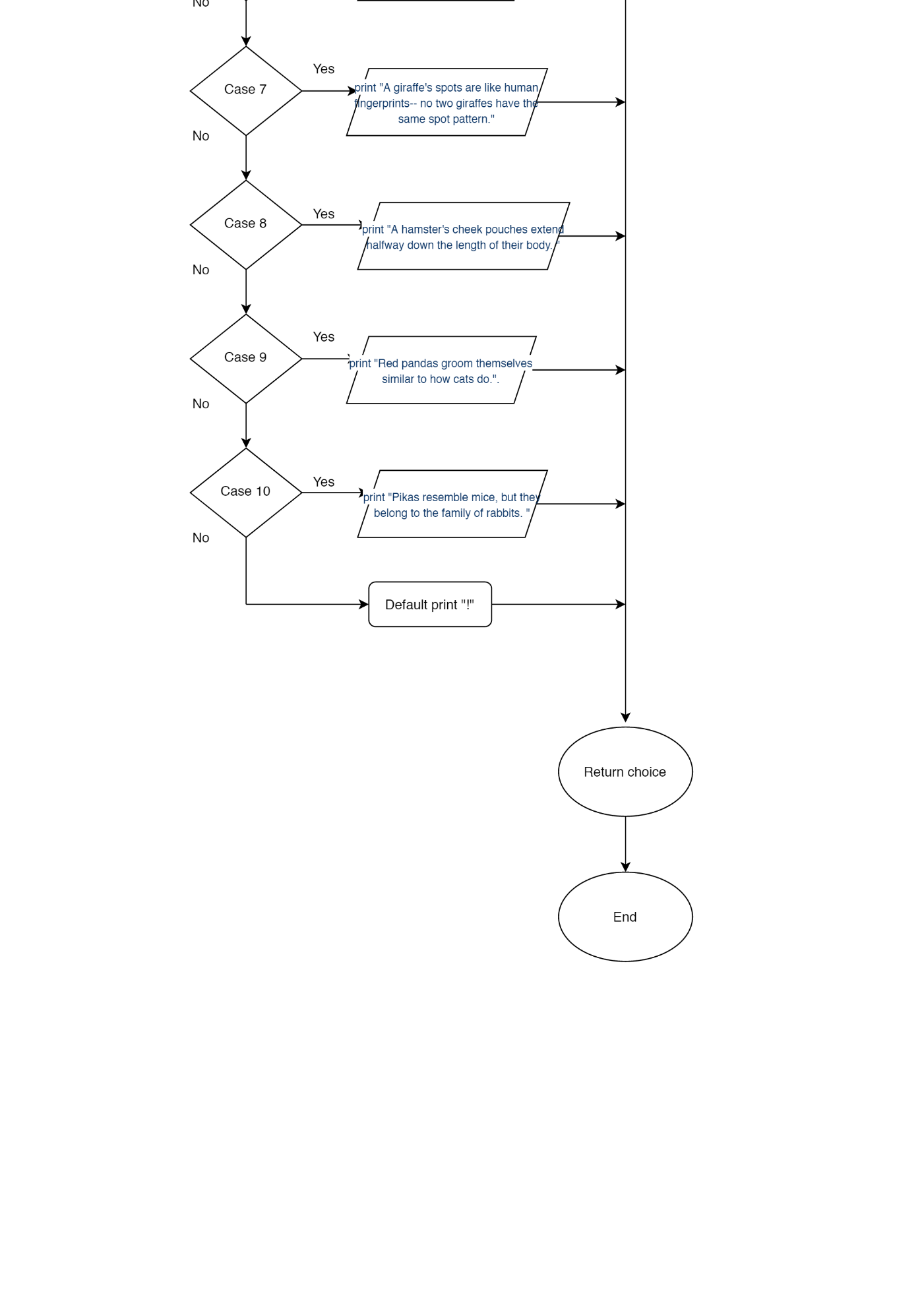
*\*Added the flowchart and state diagram as specified in the Requirements Management Planning slides and lecture \**

Product:

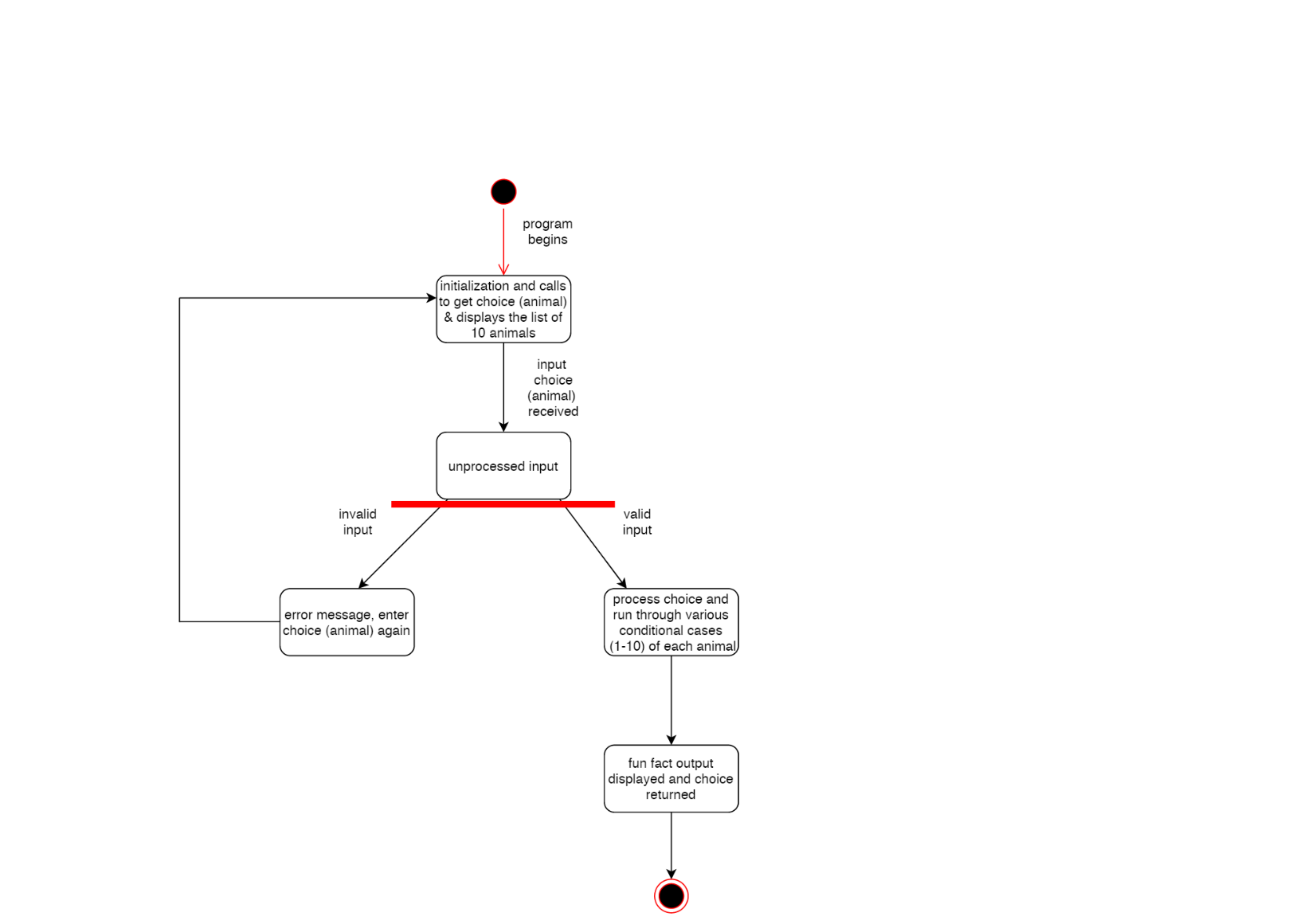
1. Behavior – Once the program is compiled and running, the user should be met with a screen that displays a list of 10 different animals. Then there should be a line prompting the user to enter their choice of animal to learn about (1 to 10 only). The program should then output the appropriate fact about the user’s selected animal. If the user enters invalid data, the program should output an error message and prompt them to enter it again. The list of animals continues to be displayed until the user enters -1 to terminate the program.
2. Inputs – Numbers 1 to 10 (aka the user’s choice) corresponding to different animals
3. Outputs – A fun fact about the animal the user selected
4. Processes - Flowchart and State Diagram attached below

Flowchart:



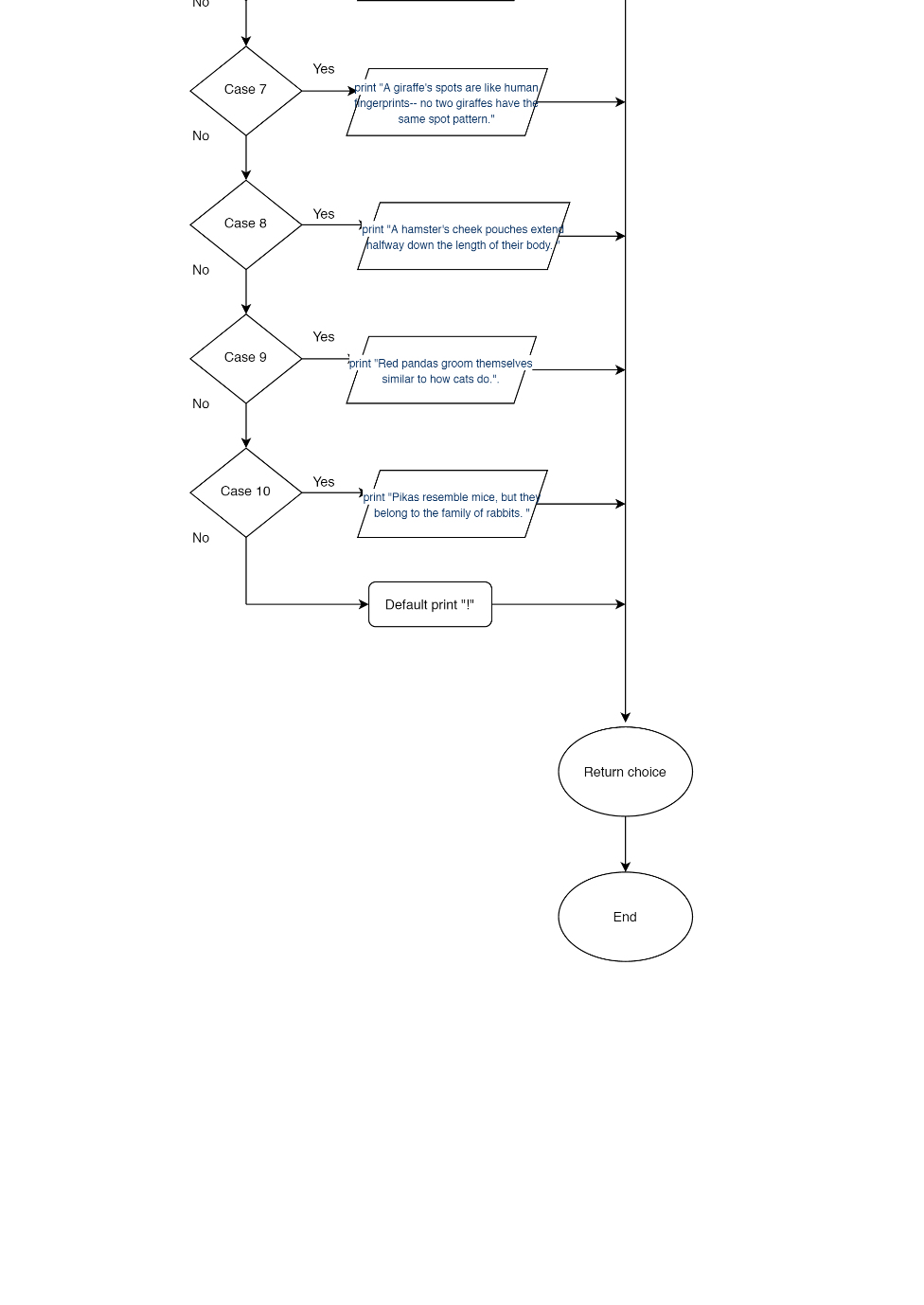
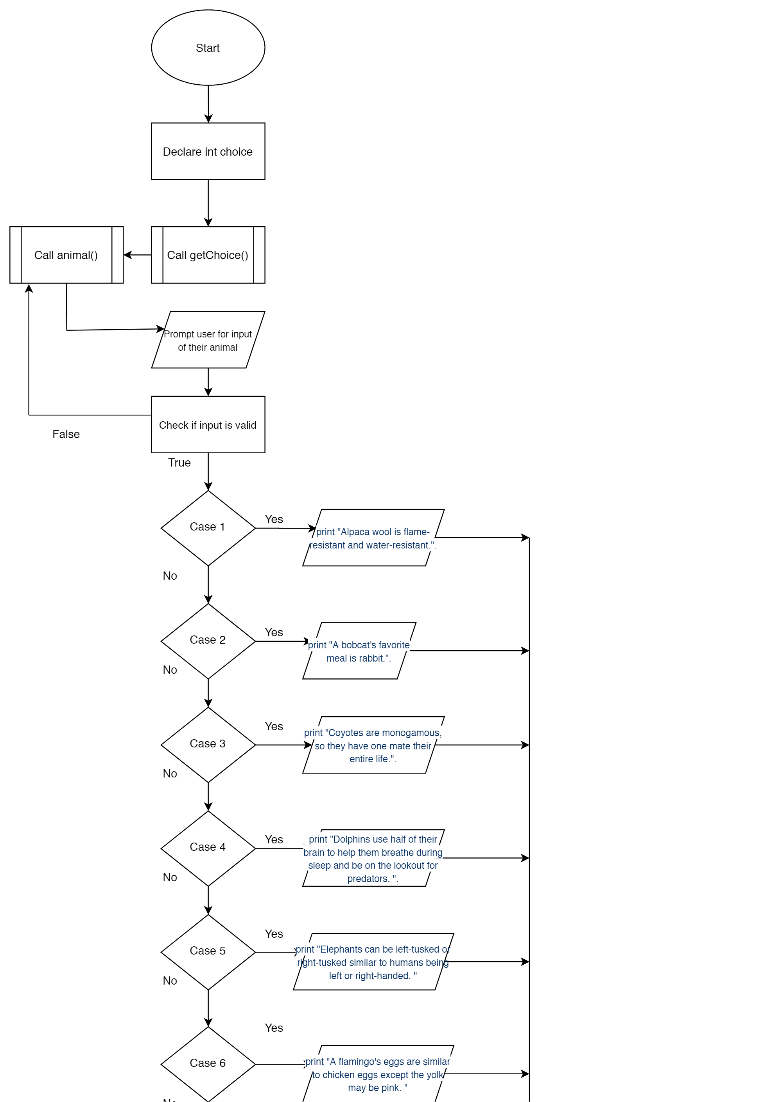


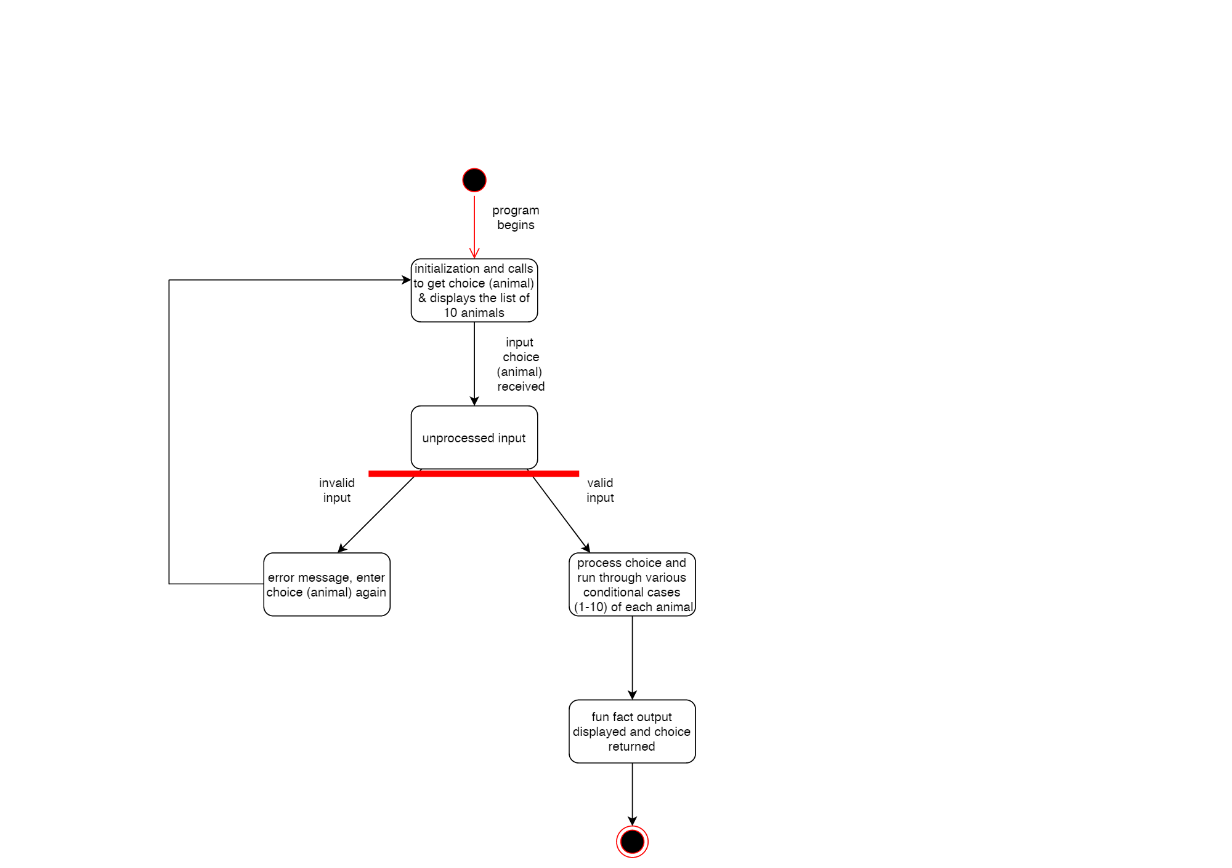
State Diagram:



1. **Design (The Answer to Requirements):**
2. In the main() function, I created a variable called choice and called the getChoice() function to retrieve the user’s choice of animal. The user’s choice is stored in the choice variable.
3. In the getChoice() function, I use a do-while loop to be able to display the list of animals for the user to choose from (call animals() function). Still inside the loop, the user is prompted to make their choice and enter it. Then I error-check the input that was entered to make sure it is 1-10 and not anything less, greater, or something that is not a number using an if statement. The buffer gets cleared to prevent an infinite loop.
4. Still in if statement, I made another if statement for if choice is -1 from the user, we exit the loop because the user wanted to end the program. Otherwise, the program displays a message that the input was invalid and to enter again.
5. A switch statement begins (stubbing portion) and displays all 10 animal facts, and depending on what the user’s choice was, it will run that particular case # and display the fact using cout print statements. The do-while statement runs as long as the input != -1, and the function returns the choice.
6. The function void animals() displays 10 different animal species in a list, and the function is called in the getChoice() function for the user to select from.

Flowchart and State Diagram have been inserted here as well since I assumed they aid in displaying the design of the program.





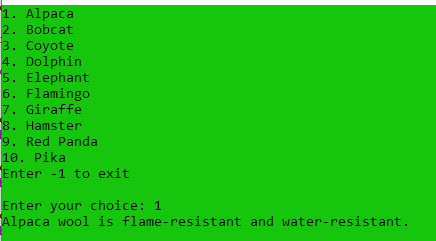
1. **Test Plan:**
2. Tests required include validating user input (so enter correct data (1-10) and incorrect data such as negative numbers, letters, or numbers greater than what is required), and make sure the program handles it appropriately.
3. Another test is making sure the correct fact is displayed corresponding to the choice of animal the user selected.
4. The testing will be completed by the developer (and user if needing feedback on project and planning a second release).

Test Execution:

The desired output on runtime should display a list of 10 animals and a prompt asking the user to select a choice of animal 1-10. The program should then display a fun fact about the user’s choice of animal.

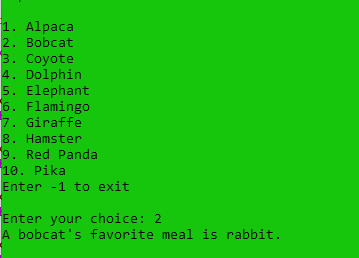
**Appendices (Test Results):**

Test One: User chooses option #1 Alpaca



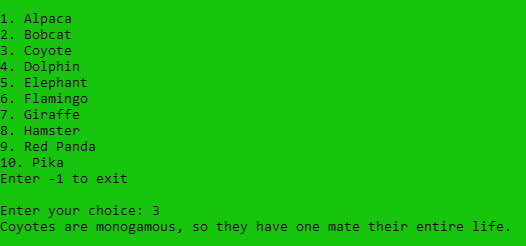
Test One conclusion: Output is a fun fact about alpacas; list displayed until user terminates the program.

Test Two: User chooses option #2 Bobcat



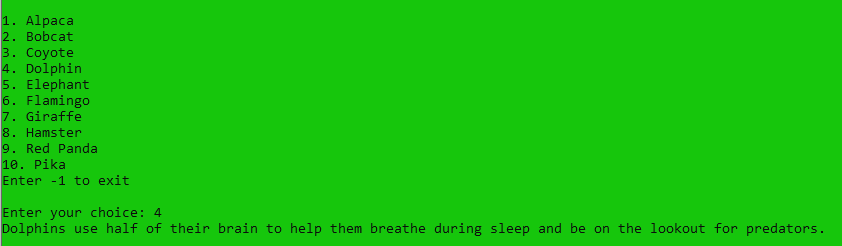
Test Two conclusion: Output is a fun fact about bobcats; list displayed until user terminates the program.

Test Three: User chooses option #3 Coyote



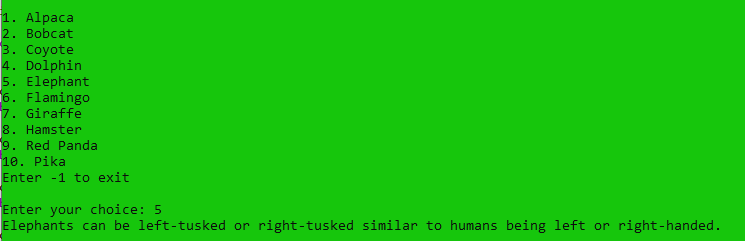
Test Three conclusion: Output is a fun fact about coyotes; list displayed until user terminates the program.

Test Four: User chooses option #4 Dolphin



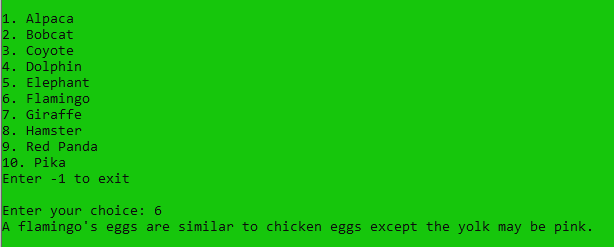
Test Four conclusion: Output is a fun fact about dolphins; list displayed until user terminates the program.

Test Five: User chooses option #5 Elephant



Test Five conclusion: Output is a fun fact about elephants; list displayed until user terminates the program.

Test Six: User chooses option #6 Flamingo



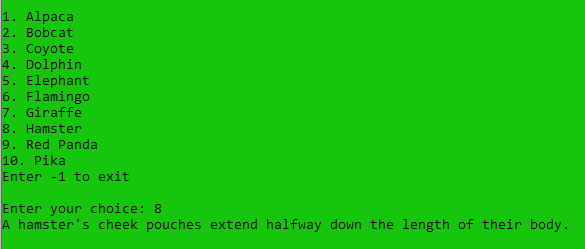
Test Six conclusion: Output is a fun fact about flamingos; list displayed until user terminates the program.

Test Seven: User chooses option #7 Giraffe



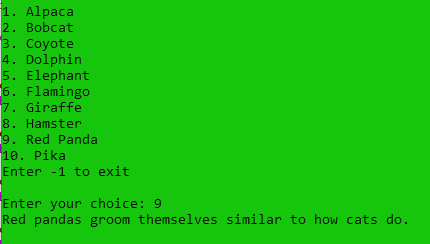
Test Seven conclusion: Output is a fun fact about giraffes; list displayed until user terminates the program.

Test Eight: User chooses option #8 Hamster



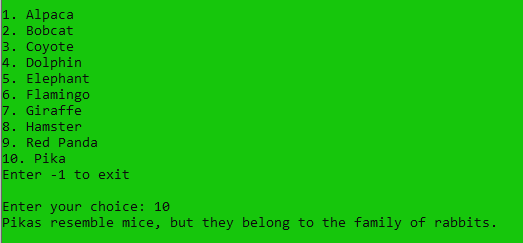
Test Eight conclusion: Output is a fun fact about hamsters; list displayed until user terminates the program.

Test Nine: User chooses option #9 Red Panda



Test Nine conclusion: Output is a fun fact about red pandas; list displayed until user terminates the program.

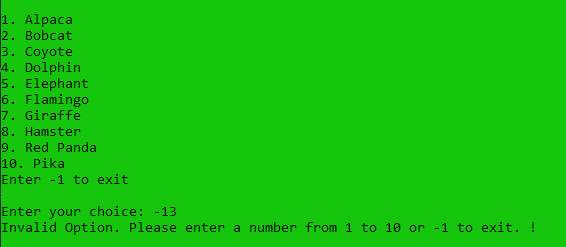
Test Ten: User chooses option #10 Pika



Test Ten conclusion: Output is a fun fact about pikas; list displayed until user terminates the program.

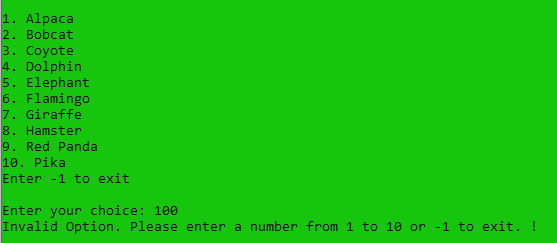
Invalid Testing:

Test 11: User enters a negative number (-13)



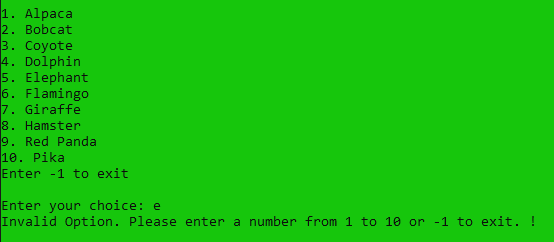
Test 11 conclusion: Output is the error message and prompting the user to enter a number 1-10

Test 12: User enters a number larger than 10 (100)



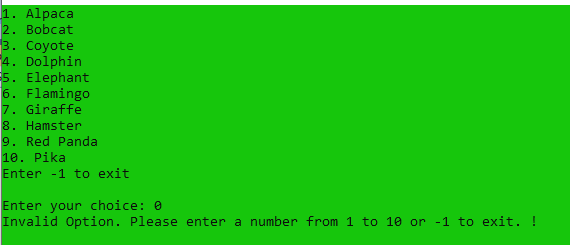
Test 12 conclusion: Output is the error message and prompting the user to enter a number 1-10

Test 13: User enters a letter (e)



Test 13 conclusion: Output is the error message and prompting the user to enter a number 1-10

Test 14: User enters a zero



Test 14 conclusion: Output is the error message and prompting the user to enter a number 1-10

------------------------------------------------------------------------------------------------------------------------------------

Exiting: User enters -1 and the program terminates

