

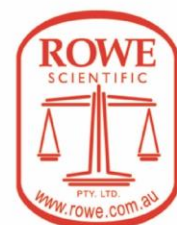


**Prize Winner**

# **Computer Programming, Apps & Robotics Year 9-10**

**Sophie Wegener**

**Glenunga International High  
School**



# Dichotomous Key Written Report

Computer Programming and Robotics

By Sophie Wegener

The aim of this entry is to model a dichotomous key created from user input. A dichotomous key is a tree-like structure used frequently in biology to identify items in the natural world. A dichotomous key has choices at each branch that lead to a final item. The dichotomous key modelled in this program grows as the user engages in the program, acting as basic artificial intelligence. This particular program bases the dichotomous key around animals, however, the questions could be easily adapted to different topics such as identifying plants, biomes or elements on the periodic table.

The program can be run in Python on any computer. In file format, the program must be run in Python. To use the hard copy of the program below, copy and paste the program into a Python file and run it in Python. To use the program, the user must follow the prompts and respond to questions by typing their response and pressing enter.

Program hard copy and explanation

```
'''Create a dichotomous key based on user input.
```

```
Sophie Wegener
```

```
0206-042.Wegener.DichotomousKeyCodingScript.py
```

```
'''
```

```
import time
```

```
class Frontend:
```

```
    '''Communicate with user'''
```

```
    @staticmethod
```

```
    def ask(question):
```

```
        '''Show question and return answer'''
```

```
        answer = input(question)
```

```
        return answer
```

```
    @staticmethod
```

```
    def say(message):
```

```
        '''Show message'''
```

```
        print(message)
```

```
vowels = ['a', 'e', 'i', 'o', 'u']
```

```
def handle_yes_no_input(user_input):
```

```
    '''Handle user yes/no response and return either y or n.'''
```

```
    while True:
```

```
        answer = user_input.strip()[:1].lower()
```

```
        if answer in ['y', 'n']:
```

```
            return answer
```

```
        else:
```

```
            user_input = input('Please
```

```
answer with either yes or no ')
```

Functions in the Frontend class enable the program to communicate with the user by showing the user messages and questions and feeding responses to the back end of the program. Separating the front and back end forms potential for a more appealing user interface. Doing this also creates potential for the program to be 'trained' by another program so that it can store more information without the hassle of entering in every new animal.

This section ensures that the user has inputted either yes or no.

```

def setup_tree():
    '''Create and return a tree.'''
    tree = []
    Frontend.say('Think of an animal...')
    time.sleep(1)
    new_animal = Frontend.ask('What is your
animal? ')
    new_feature = Frontend.ask('What feature does
your animal have? (e.g. wings, a long neck, no.
legs, etc.) ')
    # Branch
    # [Node, [<yes branch>], [<no branch>]]
    new_branch = [new_feature, [new_animal],
[None]]
    tree.append(new_branch)
    return tree

def question_inner_loop(branch_index, branch):
    '''Either deal with leaf or call outer loop
to ask about the next feature.
arguments:
branch_index -- index of inner branch 1 for
yes, 2 for no
branch -- current branch of tree
list with three elements, [feature,
yes branch, no branch]
'''
    is_leaf = len(branch[branch_index]) == 1
    if is_leaf:
        animal = branch[branch_index][0]
        if (animal is not None) or (branch_index
== 1):
            # Guess the animal.
            animal_y_or_n = Frontend.ask('Is your
animal a ' + animal + '?')
            if handle_yes_no_input(animal_y_or_n)
== 'y':
                Frontend.say('yay!')
                time.sleep(1)
            else:
                new_animal = Frontend.ask('What
is your animal? ')
                if animal[:1].lower() in vowels:
                    a_an = 'an'
                else:
                    a_an = 'a'
                new_feature = Frontend.ask(f'What
feature does your animal have that {a_an}
{animal} does not have? ')
                new_branch = [new_feature,
[new_animal], [animal]]
                branch[branch_index] = new_branch
            else:
                new_animal = Frontend.ask('What is
your animal? ')

```

This part of the program uses the first information from the user – an animal and its feature - to set up the dichotomous key.

This section of the program checks whether the user has reached a choice or an animal. If the user has reached a choice, the program switches to the outer question loop. If the user has reached an animal, the program asks the user if they were thinking of that animal. If it was not the right animal, the program prompts the user to reveal their animal and the difference between their animal and that guessed by the program. The dichotomous key will then be modified to include the new animal.

```

        new_feature = Frontend.ask('What
feature does your animal have? (e.g. wings, a
long neck, no. legs, etc.) ')
        new_branch = [new_feature,
[new_animal], [None]]
        branch[branch_index] = new_branch
    else:
        branch = branch[branch_index]
        # Go back to start loop
        question_outer_loop(branch)

def question_outer_loop(branch):
    '''Ask about feature then descend into
branch.
arguments:
branch -- current branch of tree
list with three elements, [feature,
yes branch, no branch]
'''
    # Start loop
    feature = branch[0]
    y_or_n = Frontend.ask('Does your animal have
' + feature + '? ')
    if handle_yes_no_input(y_or_n) == 'y':
        branch_index = 1
    else:
        branch_index = 2
    question_inner_loop(branch_index, branch)

def main():
    '''Set up tree and start game loop.'''
    tree = setup_tree()
    while True:
        if not
handle_yes_no_input(Frontend.ask('Would you like
to continue playing? ')) == 'y':
            break
        Frontend.say('Think of a new animal...')
        time.sleep(2)
        question_outer_loop(tree[0])

main()

```

This part of the program asks the user whether their animal has a certain feature. The program then works out which branch it needs to follow and descends into the inner question loop.

This section of the program calls the dichotomous key to be initialised and starts the game loop.

My dad in discussing data structures early on, introducing classes in the program and modifying variable and function names has provided some support.