Iain Thorpe
Mr. Orkney
Computer Science Principles, Per. 3
21 February 2018

Written Response

2a.) My program is a Job and Personality selector that can determine the personality and job of your hero in my demo version of the digital world of Miitopia, a video game on the Nintendo 3DS. The programming language I used for this program was Python 2 in the structure application, Canopy. The purpose of my program is to determine what your hero's personality and job would be in the multitude of options based off the options in the game itself after a series of questions about preference and about "Would you rather have this or that?" The video included illustrates the program in action and a brief look at the code itself.

2b.) My program was plain and simple to put together. An example about how I was able to put my program together are when I incorporated if, then, else if statements. These allowed me to separate the options into eight branches and sixteen branches. However, there were some problems I had when making the whole thing. For example, a problem I encountered was that when I was going back and programming in the "elif no" option, I noticed that the else statements didn't go back to the question you were on. In order to fix that, I implemented the "vars" function in order to allow you to try again on any question. Another problem I faced was that when I was working on the quizzes, I had a hard time determining what the questions were going to be. In order to fix that, I created a side Google drawing of the branching lines, which I organized by finding which ones could be separated by one question, then I went back to my program and added in the questions and the separating branching paths.

2c.)

```
vars[2] = 3
Print(‘ ‘)
Print(‘ ‘)
Print(‘ ‘)
While vars[2] == 3:
```

This segment of code helps my program fundamentally achieve its purpose because the algorithms implemented here help my program. The algorithms used here function individually because of the vars function. However, the vars function also allows the algorithms to work together by having the vars function be implemented throughout the whole section of code with the while statement. The while statement, when incorporated with the vars function, allows the algorithms to run together.

2d.)

```
print('Do you prefer close quarters combat?\n')
answer = raw_input()
print('\n'*5)
answer = answer.lower()
```

This abstraction helped manage my program’s complexity because the == in the while function allows the program to stay on any question without repeating all the text before the question. The program utilizes the == abstraction because the program can’t return to the previous question when the viewer responds with anything other than yes or no.