Evaluation

# Strengths

One strength is the menu being adaptive. It will only show the full scope of options once a user has created a polling file. This stops any errors from occurring that could potentially arise from not having a voting file to write to.

The system will also automatically add the decided name of a file to a list stored in a file specifically used to store the many file names that may have been used. This continues into the collation of data as it allows for the list of file names to be used to display a list of names to the user for them to choose from to add to a total votes file. I believe this greatly increases the usability of the system as it clearly displays any files available, and I would say it is fair to believe that the average admin user may not remember all file names and as such it is useful to have this function.

Furthermore, in the collation of data it will allow the user to add multiple files without having to return to the menu each time. I believe this to also increase the usability.

Once again in the collation of data it will also display a list of the total votes for each candidate. Once again I believe this to increase the usability of the system.

In the polling section itself there are many checks in place to ensure that invalid data is not capable of being stored in the files. This ensure no errors occur and I believe it to be a major strength.

# Weaknesses

Whilst I believe my system is to a high standard, weaknesses will always be apparent.

The naming convention I chose, being “Belfast\_(name)\_(current date)”, is subject to confusion. This is because I overlooked directly explaining this to the user and as such for first time admin users they may name a file something such as “Belfast East 2022” which would result in “Belfast\_Belfast East 2022\_2022-12-15” being stored. This obviously results in a relatively bad name for a file as it contains repeated data for no reason.

When collating the data for a total votes file an admin user may choose the same file multiple times. This may result in a mistake in results which is an issue. However, if the user is keeping track of which files they choose there may not be anything to worry about.

When a user is voting they may enter a duplicate vote and it will only notify them that it is not allowed after the rest of the results are entered. This may prove to be somewhat annoying for users, however, is not a massive issue as it still does not record duplicate votes.

Another small weakness is that when viewing the statistics for the votes the list is subject to a small amount of rounding error. Whilst it is accurate to two decimal places it is not perfect. And when a user is collating the data it will print the collated votes in the total votes file, however, the rounding is not perfect either and may display more than two decimal places. This is due to issues with how the data was being stored but again is not a major issue, it may only look somewhat unappealing.

# Functions

## Main

The main function was simply to be called to run the program and make an exit function extremely easy to implement. As well as calling the menu function and retrieving the desired menu option the user wished to use. Primarily, while loop, if and elif statements were used to allow for the options to be discerned. The main only used integer values passed between itself and the menu function as it was all that was required for choosing a menu option.

## Menu

The menu function was where the input of a menu option was taken in and where the printing of the menu options took place. This allowed for discerning of if a user had created a voting file or not, which was beneficial for more versatility and lower chances of any errors occurring.

## Polling File

This was where a user created a polling file. It would allow a user to name a voting file whatever they pleased. When the input was received it would create the file by implementing seven zeros into a file using the chosen name. In this function it would also set the currentFile global variable to the decided name. This made it significantly easier to write to the current file when needed. Finally the function also added the new file name to a list of file names to be used later in the collating data function. This all made for a very effective file creation system. It also made it significantly easier on myself when coding as the previously mentioned storing of polling booth names allowed for easy retrieval and use later.

## Polling Enter

This is where the voting took place. It would begin by asking for the user’s gender and would store the input. Disallowing any invalid entries, however, if an admin user wished to return to the menu a secret password was implemented, quite literally being “secret password”. This made debugging and statistics reviewing significantly easier as I did not have to re-open the file every time I wished to test another aspect of the system. The votes were recorded and did not allow for any invalid inputs then would write the votes to the polling file. This all was done relatively easily using for loops to simplify the checks for duplicated votes and writing of data.

### Test Votes

This was another function used only in the pollingEnter function specifically to tests the votes entered by a user. It saved many lines of repeated code by checking for any letter inputted or out of range value. If the inputted value passed the check it would then pass back to the pollingEnter function a “choices” list. This made debugging significantly easier as if I ever needed to change the check then I would need to only change a few lines of code.

## Polling Data

This is where the admin user could collate data. I believe this to be one of the highlights of my system as it very clearly displays all voting files stored by the system. This makes it much easier for the user to select which files they may wish to add to the total as a regular admin user may not remember every file name created. It also neatly displays the totals to be viewed.

## Polling Stats

This is where the admin user can view percentages for each candidate. A very simple yet effective function as it does everything asked of the system in a neat and orderly fashion.

# Viability

The system is easy for a user to understand and use, which is quite obviously a major benefit. The options are very clearly displayed, and the system prevents the user from doing anything that may result in a crash. Such as proceeding to enter votes without a polling file created. A smooth experience for the user is absolutely needed for any program that will be used on a wide scale. The system has been thoroughly tested and prepared for any data inputs the user may wish to enter and has considered any invalid inputs and how to handle them to stop any crashes from occurring. And as such for these reasons, and the previously mentioned strength, I believe my system to be quite viable for use.