

PRF – Fundamentals of Programming Coursework 1

Write a Python program that meets the following specification. Pay attention to good programming style, and be sure to split the program into functions and/or modules as appropriate.

The specification is presented as a series of four stages, and you are expected to work through it in the way. **Be sure to keep a final copy of your program files at the end of each stage, as these will be needed for assessment.**

Stage 1 [8 marks]

There is much debate as to the *best* television program of all time. To sort the matter out once and for all, a nationwide poll has been organised; the country has been split into a number of voting areas for this purpose. There are five candidate programs in the final voting, and each has been assigned a voting code. The codes (together with a useful reference URL) are as follows:

Program	Code	Useful Reference URL
The Clangers	CLA	http://www.bbc.co.uk/cult/classic/clangers/
Rainbow	RBW	http://rainbow.web.com/
Bagpuss	BAG	http://www.bbc.co.uk/cult/classic/bagpuss/
Mr Benn	MRB	http://www.toonhound.com/mrbenn.htm
Noggin the Nog	NOG	http://www.nogginthenog.co.uk/

The program for this stage should handle the votes from one area as follows:

- Prompt the user to enter the name of the voting area (this can be any string, and need not be validated).
- Prompt the user to enter the number of votes for each programme in turn.
- Verify that the number of votes entered each time is a positive integer, and if appropriate request that the number is entered again.
- Display the name of the most popular program. If there is no one most popular program it is sufficient to display "Dead Heat".
- Display the name of the voting area and the percentage of the vote achieved by each programme (and the vote itself) in a neat table.

The precise dialogue produced is up to you, provided that all the functions are included. It might look something like this:

```
cslin101% ./voting.py
Which Voting Area: Leeds
Votes for "The Clangers": 10
Votes for "Rainbow": 20
Votes for "Bagpuss": 39
Votes for "Mr Benn": 40
Votes for "Noggin the Nog": 40
```

Here are the Results from Leeds

The Result is a Dead Heat!

Voting Statistics for Leeds

```
There were 149 votes cast, as follows:
"The Clangers" 10 6.71140939597%
"Rainbow" 20 13.4228187919%
"Bagpuss" 39 26.1744966443%
"Mr Benn" 40 26.8456375839%
"Noggin the Nog" 40 26.8456375839%
```

Stage 2 [6 marks]

This program extends the program from the first stage as follows:

- In the event of a dead heat the names of the programmes involved in the tie should be displayed.
- The program should run for a number of voting areas. It should start by prompting the user to enter the number of areas, and should run as before for each one. At the end the program should display the total number of votes and percentages achieved by each programme across all the areas, and should thus report the overall most popular programme.

Once again the precise details of dialogue and layout produced is up to you. It might resemble:

```
cslin101% ./voting.py
Enter the Number of Areas: 2
```

```
Enter Voting Area #1: Leeds
Votes for "The Clangers": 10
Votes for "Rainbow": 20
Votes for "Bagpuss": 30
Votes for "Mr Benn": 40
Votes for "Noggin the Nog": 50
```

Here are the Results from Leeds

The Winner is "Noggin the Nog"

Voting Statistics for Leeds

There were 150 votes cast, as follows:

"The Clangers"	10	6.66666666667%
"Rainbow"	20	13.3333333333%
"Bagpuss"	30	20.0%
"Mr Benn"	40	26.6666666667%
"Noggin the Nog"	60	40.0%

Stage 3 [3 marks]

This program extends the program from the second stage by adding a bar chart display of the final overall results. The bars should be made up of # characters, one character should be displayed for every ten votes, and the bar should be preceded with the programme's voting code. There should not be a space between each character in the bar. Any remainder should be disregarded (so 198 votes appears as 19 characters, 101 as 10, and so on).

Stage 4 [3 marks]

Amend the program from the third stage so that:

- All percentages are given to exactly two places of decimals.
- The program ends by reporting the names of the areas in which each program was the most popular.

Assessment

This coursework is worth 10% of the assessment for PRF.

The deadline for this work is 5pm on 21st October 2005. Your final programs from each stage that you have completed should be submitted via SIS in the usual way.

Purpose

The purpose of this assessment is to give you practice in writing a reasonably complex Python program. The program will use most, if not all, of the Python statements, functions, and techniques that we have covered in the first part of the module.

AMJ, 31st October 2002, 4th October 2005