Assignments

Assignment - In progress

Add attachment(s), then choose the appropriate button at the bottom.

Draft - In progress
Submitted
Returned
Title
Assignment 2
Due
Feb 8, 2022 11:55 PM
Number of resubmissions allowed
Unlimited
Accept Resubmission Until
Feb 10, 2022 11:55 PM
Status
Not Started
Grade Scale
Points (max 100.00)
Modified by instructor
Feb 2, 2022 11:55 AM

Instructions

Assignment overview

We would like students to experience command line input with C types of character, int, and float, to understand and use C types such as char, int, and float, as well as the flow control structures studied in class.

This assignment consists of two parts.
In part one, you are required to write a C program to perform some simple conversions.
In part two, you are to write a C program to translate integers into the equivalent English word.

Part one: 70%

The goal of the exercise is to implement a simple converter, called ”converter”, which works
as follows.

1. First, the user is asked what she/he wants to do. An integer can be entered with the following six actions associated with different values of the integer. You can assume that the user will always enter an integer.
    - 1 for conversion between Kilometre and Mile (1 kilometre == 0.621371 miles)
    - 2 for conversion between Litres and Gallons (1 litre == 0.264172 gallons)
    - 3 for conversion between Hectares and Acres (1 hectare == 2.47105 acres)
    - 4 for conversion between Kilograms and Pounds (1 kilogram == 2.20462 pounds)
    - 5 for quit
    - {any other integer} for prompting the user to try again.
2. In case of 1 to 4, the program will ask the direction of the conversion. In each case, one of two characters can be entered corresponding to each conversion direction. Your program should handle non-valid single character input values.
    - In case of 1, the program will prompt the user for two choices and wait for a character input
        - K for conversion from Kilometre to Mile
        - M for conversion from Mile to Kilometre
    - In case of 2, the program will prompt the user for two choices and wait for a character input
        - L for conversion from Litres to Gallons
        - G for conversion from Gallons to Litres
    - In case of 3, the program will prompt the user for two choices and wait for a character input
        - H for conversion from Hectares to Acres
        - A for conversion from Acres to Hectares
    - In case of 4, the program will prompt the user for two choices and wait for a character input
        - K for conversion from Kilograms to Pounds
        - P for conversion from Pounds to Kilograms
    - Note/Hint: Since whitespace counts as a character, to read a character properly, your program should handle the leading space character, tab character, and end of line character, if any.
3. Then the program descriptively prompts the user for the input value ("Please enter a value..."), descriptively displays the result ("Your conversion is..."), and returns to the main menu. The input value should be a float number and we assume the user will always enter valid numbers.
4. Your program should follow good programming styles, i.e. write clear code, choose good variable names, make proper comments, etc. Your program must have a comment block at the top describing the name of the program, it's purpose, it's author (ie. you and your student number), etc.

Part two: 30%

1. Write a succinct program to convert any integer from 1-999 into the English word (in lower case). Eg.

    1. ```
       Please enter a value (1-999, 0 to quit): 1
       You entered the number one

       Please enter a value (1-999, 0 to quit): 15
       You entered the number fifteen

       Please enter a value (1-999, 0 to quit): 47
       You entered the number forty-seven

       Please enter a value (1-999, 0 to quit): 135
       You entered the number one hundred and thirty-five

       Please enter a value (0 to quit): 0
       ```

2. You can assume the input is always one valid integer in the range 0-999 (ie. no negatives and no integers higher than 999). The integer will never have preceding zeros (eg. The input would be 1, never 001) or trailing decimal zeros (eg. The input would be 1, never 1.000).
3. Then the program descriptively prompts the user for the input value ("Please enter a value..."), descriptively displays the result ("You entered the number..."), and returns to the main menu.
4. Your program should follow good programming styles, i.e. write clear code, choose good variable names, make proper comments, etc. Your program must have a comment block at the top describing the name of the program, it's purpose, it's author (ie. you and your student number), etc.

Testing and submitting your programs

You should test your program by compiling and running it on Gaul before submitting (This is how the TA will be testing your program). For part one, each case should be tested at least once. For part two, you should test at least two integers 1-9, at least two integers 10-19, at least two integers 20-99, and at least two integers 100-999. Capture the screen of your testing by using script command. There should be two resulting script files, one for each part.

Submit a single tarball called 251xxxxxx-Assignment2.tar containing a directory structure that looks like this:
       ```

```
251xxxxxx-Assignment2
    |----- converter.c
    |----- converter.script
    |----- intToEnglish.c
    |----- intToEnglish.script
```

Additional resources for assignment

No attachments yet

Grading Rubric

Preview Rubric

Submission

Attachments

No attachments yet

Select a file from computer 选择文件 未选择任何文件

Proceed | Preview | Save Draft | Cancel

Don't forget to save or proceed!