Lab 08: Basic Types

CSE 4108

Structured Programming I Lab

October 2022

Lab Tasks

1. **Give me a break!:**

Write a program that prompts the user to enter a number n ($n < 10^4$), then prints all even squares between 1 and n.

Now, modify the program so that it pauses after every 24 squares and displays the following message:

Press enter to continue...

After displaying the message, the program should use ***getchar*** to read a character. ***getchar*** won't allow the program to continue until the user presses the Enter key.

2. **Double Trouble!:**

Write a program that finds the ***sum*** in a series of numbers (integers) entered by the user. The program must prompt the user to enter numbers one by one. When the user enters **0** or **a negative number**, the program will end and display the results.

Now modify the program to ***sum*** a series of double values.

3. **That sounds Phoney:**

Have you seen one of those commercials where at the end, they go like, "Call **1-800-DOMINOS** for your free delivery"? What do those phone numbers mean? You can't dial **1-800-DOM-INOS** from a dialer, can you? Well these numbers are called Phonewords.

The letters represent numbers from the dialpad of a phone. In case you don't have a phone nearby, here are the letters on the keys: **2=ABC, 3=DEF, 4=GHI, 5=JKL, 6=MNO, 7=PQRS, 8=TUV, 9=WXYZ.**

So the number **1-800-DOM-INOS** actually means **1-800-366-4667**. The main advantages of Phonewords over standard phone numbers include increased memorability and increased response rates to advertising. But they are, at times, cumbersome to convert to actual numbers. That's why you have to write a program that translates an alphabetic phone number into numeric form.

Sample run:

```
Enter phone number: 1-800-DOM-INOS
1-800-366-4667
```

Notice that, if the original phone number contains non alphabetic characters (digits or punctuation, for example), they remain unchanged. You may assume that any letters entered by the user are upper case.

4. **Scrabbler**:

In the Scrabble Crossword Game, players form words using small tiles, each containing a letter and a face value. The face value varies from one letter to another, based on the letter's rarity.

The face values are:

1: AEILNORSTU, 2: DG, 3: BCMP, 4: FHVWY, 5: K, 8: JX, 10: QZ.

Write a program that computes the value of a word by summing the values of its letters.

Sample run:

Enter a word: <u>pitfall</u>
Scrabble value: 12

Your program should allow any mixture of lower-case and upper-case letters in the word.

5. **<u>Look at the sizeof that thing!:</u>**

Write a program that prints the values of sizeof(int), sizeof(short), sizeof(long), sizeof(long long), sizeof(float), sizeof(double), and sizeof(long double).

6. **<u>Plus-Signed Fractions - The Sign Of Four:</u>**

In **Lab 2 - Formatted Input/Output** (Signed fraction problem), you were asked to write a program that added two fractions entered by the user. Modify the program so that the user may add, subtract, multiply, or divide two fractions (by entering either **+, -, *,** or **/** between the fractions).

7. **<u>Welcome Aboard 2:</u>**

In **Lab 4 - Selection Statements** (Welcome Aboard problem), you were asked to write a program that asks the user to enter a time in 24-hour format and display the departure and arrival time for a flight whose departure time is closest to that entered by the user. Modify the program so that the user enters a time using the 12-hour clock. The input will have the form **hours:minutes** followed by either ***A, P, AM, or PM*** (either lower-case or upper-case). White space is allowed (but not required) between the numerical time and the AM/PM indicator.

Example of valid input:
1:15P
1:15PM
1:15p
1:15pm
1:15 P
1:15 PM
1:15 p
1:15 pm

You may assume that the input has one of those forms; there is no need to test for errors.

8. **Hermione's Time Turner:**

Write a program that asks the user for a 12-hour time, then displays the time in 24-hour form:

```
Enter a 12-hour time: 9:11 PM
Equivalent 24-hour time: 21:11
```

9. **Vowels rush:**

Write a program that counts the number of vowels (a, e, i, o, and u) in a sentence.

Sample run:

```
Enter a sentence: I think therefore I am.
Your sentence contains 8 vowels.
```

10. **APA Citation Style:**

Write a program that takes a first name and last name entered by the user and displays the last name, a comma, and the first initial, followed by a period.

Sample run:

```
Enter a first and last name: Dylan Rhodes
Rhodes, D.
```

The user’s input may contain extra spaces before the first name, between the first and last names, and after the last name.

11. **Serial Calculator:**

Write a program that evaluates an expression. Sample run:

```
Enter an expression: 1+2.5*3
Value of expression: 10.5
```

The operands in the expression are floating-point numbers; the operators are **+, -, *,** and **/**. The expression is evaluated from left to right (no operator takes precedence over any other operator).

12. **Average Length:**

Write a program that calculates the average word length for a sentence.

Sample run:

```
Enter a sentence: I have no special talent. I am only passionately curious.
Average word length: 4.8
```

For simplicity, your program should consider a punctuation mark to be part of the word to
which it is attached to. Display the average word length to one decimal point.

13. **<u>Newton's Method:</u>**
Write a program that uses Newton's method to compute the square root of a positive floating point number. Sample run:
Enter a positive number: <u>3</u>
square root: 1.73205

Let x be the number entered by the user. Newton's method requires an initial guess y for the square root of x (we'll use y = 1). Successive guesses are found by computing the average of y and $\frac{x}{y}$. The following table shows how the square root of 3 would be found:

x	y	x/y	Average of y and x/y
3	1	3	2
3	2	1.5	1.75
3	1.75	1.7149	1.73214
3	1.73214	1.73196	1.73205
3	1.73205	1.73205	1.73205

Note that the values of y got progressively closer to the true square root of x. For greater accuracy, your program should use variables of type double rather than float. Have the program terminate when the absolute value of the difference between the old value of y and the new value of y is less than the product of .00001 and y. **Hint:** Call the **fabs** function to find the absolute value of a double. (You'll need to include the **<math.h>** header at the beginning of your program in order to use **fabs**).

14. **Factorial Frenzy:**

Write a program that computes the factorial of a positive integer.

Sample Run:

Enter a positive integer: 6
Factorial of 6: 720

(a) Use a short variable to store the value of the factorial. What is the largest value of n for which program correctly prints the factorial of n?

(b) Repeat part (a), using an int variable instead.

(c) Repeat part (a), using a long variable instead.

(d) Repeat part (a), using a long long variable instead.

(e) Repeat part (a), using a float variable instead.

(f) Repeat part (a), using a double variable instead.

(g) Repeat part (a), using a long double variable instead.

In cases (e)-(g), the program will display a close approximation of the factorial, not necessarily
the exact value.