1 Objective

The purpose of this assignment is to test your familiarity with the selection statements (if ... else and switch). You’ll also need to do fundamental algorithmic error checking. These programs have to be tested on linprog before they are turned in.

Email your 2 files ordering.cpp and timer.cpp to jayarama@cs.fsu.edu

2 Program 1 - Ordering 3 numbers

For this program, you need to read in 3 distinct integers from the user and print them in ascending or descending order.

1. Prompt the user and read in 3 integers. Store them in 3 variables. (10 points)
2. The user might enter the same numbers. This is OK. You don’t have to check for that.
3. Prompt the user to enter their preferred ordering. The user will enter ‘A’ for ascending and ‘D’ for descending. (5 points)
4. If the user has entered anything other than ‘A’ or ‘D’, assume ascending order. (5 points)
5. Use if ... else statements, in whatever combination required, to print the numbers out in the appropriate order. (25 points)
6. Add comments wherever appropriate to explain your logic. (5 points)

Sample Run 1

The underlined text is the user input.

Enter 3 numbers:
25
10
17
Enter the ordering required:
A
The numbers in order are : 10 17 25
Sample Run 2

Enter 3 numbers:
-8
40
12
Enter the ordering required:
D
The numbers in order are: 40 12 -8

3 Problem 2 - Timing birds traversing a distance

Arthur, King of the Britons, has tasked you with figuring out the amount of time it would take his faithful messenger birds to carry messages (which may or may not be in coconuts) across certain distances. His assistant, Patsy, has provided you with a table that contains the speed of the birds, and the letter with which the birds will be referred to. Use the table to calculate the times.

<table>
<thead>
<tr>
<th>Bird</th>
<th>Speed (m/s)</th>
<th>Reference Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>African Swallow</td>
<td>11.5</td>
<td>A</td>
</tr>
<tr>
<td>African Swallow + Coconut</td>
<td>9.5</td>
<td>C</td>
</tr>
<tr>
<td>European Swallow</td>
<td>11</td>
<td>E</td>
</tr>
<tr>
<td>European Swallow + Coconut</td>
<td>9</td>
<td>S</td>
</tr>
<tr>
<td>Owl</td>
<td>17.88</td>
<td>O</td>
</tr>
<tr>
<td>Owl + Coconut</td>
<td>15</td>
<td>W</td>
</tr>
</tbody>
</table>

1. Print a menu for the user. In the menu, display the birds and their reference letters. (10 points).
2. Prompt the user the enter the bird in question. Read in the reference letter. (5 points)
3. Prompt the user to enter the distance (in kilometers) and read in the value (5 points).
4. Calculate the time taken by the bird and print it (20 points).
5. If the reference letter entered doesn’t match any bird, print an error message (5 points)
6. Add comments wherever appropriate to explain your logic. (5 points)

Sample Run 1

African Swallow - A
African Swallow + Coconut -C
European Swallow - E
European Swallow + Coconut - S
Owl - O
Owl + Coconut - W
Enter the bird: C
Enter the distance (in kilometers): 12.5
The time taken is 1315.79 seconds.
Sample Run 2
The underlined text is the user input.

African Swallow - A
African Swallow + Coconut - C
European Swallow - E
European Swallow + Coconut - S
Owl - O
Owl + Coconut - W
Enter the bird: X
Enter the distance (in kilometers): 19
Error: Invalid reference letter for the bird.

4 Generic Grading Guidelines

1. Please make sure you’re only using the concepts already discussed in class. That is, please try and restrict yourself to input/output statements, variables and selection statements.

2. Each program is worth 50 points.

3. Please make sure that you’re conforming to specifications (program name, print statements, expected inputs and outputs etc.).

4. Please make sure your code is readable.

5. Please make sure you’ve compiled and run your program before you turn it in. Compilation errors can be quite costly.