



PRIFYSGOL
BANGOR
UNIVERSITY

Imperative Programming in C

Laboratory 10: Linked Lists

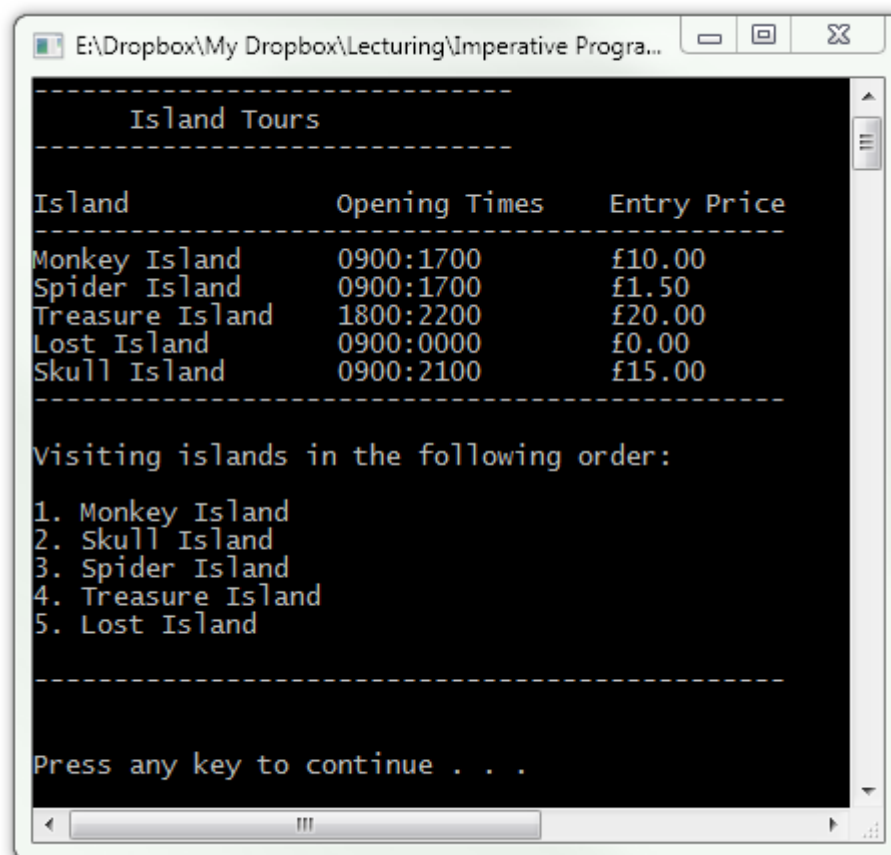
Dr. Llyr ap Cenydd

Exercise 1 - Island Tours

Your task for this exercise is to create a data structure that can hold information about several islands on a tour. For example the structure could hold the island's name, opening times and price. Your structure should also store a pointer to another island.

Once you have created your structure, connect them up as a single linked list, with each island holding a pointer to the next. This will define the order the islands will be visited.

For the final part of the program, use a pointer to loop through the linked list, starting at the first island. The code should iterate through the linked list until it reaches the end. Your output should resemble the following:



```
E:\Dropbox\My Dropbox\Lecturing\Imperative Progra...
-----
Island Tours
-----
Island           Opening Times    Entry Price
-----
Monkey Island    0900:1700        £10.00
Spider Island     0900:1700        £1.50
Treasure Island   1800:2200        £20.00
Lost Island       0900:0000        £0.00
Skull Island      0900:2100        £15.00
-----

Visiting islands in the following order:

1. Monkey Island
2. Skull Island
3. Spider Island
4. Treasure Island
5. Lost Island

-----

Press any key to continue . . .
```

Hints

- Look over the lecture notes for example code
- `(myStructPtr->next == NULL)` will be false if `myStructPtr->next` is undefined.

Challenge - Island Tours database

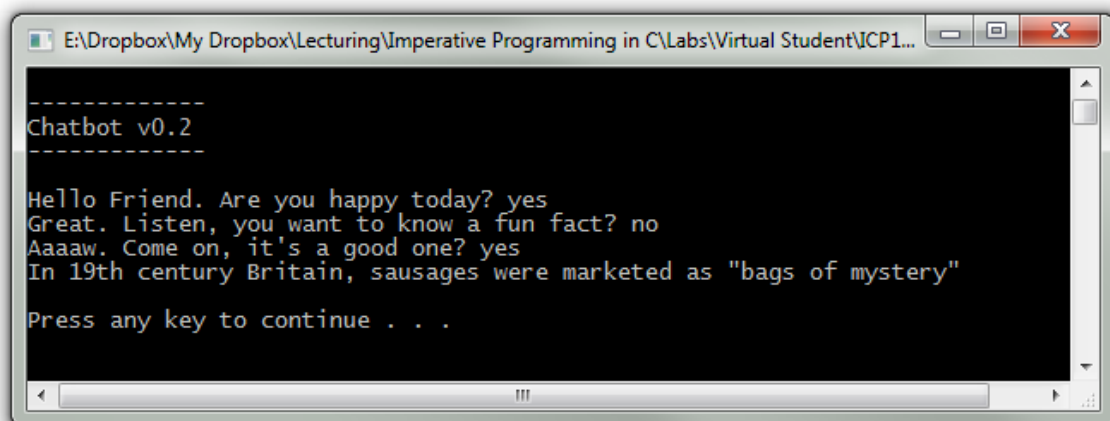
Extend your program from Exercise 1 so that the user can insert, remove and modify the order of islands in the linked list.

Exercise 2 -Chatbot v0.2

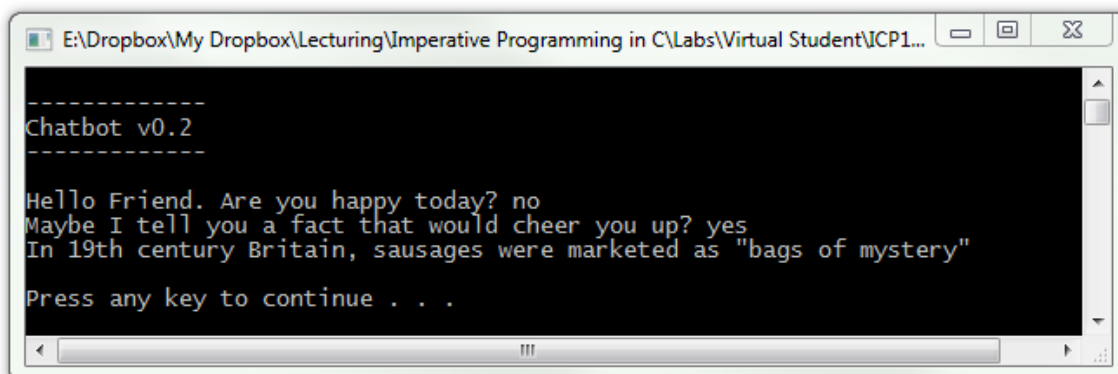
Recall that in lab 2 you were asked to create a primitive chatbot. Your task for this exercise is to create the next generation of primitive chatbots, this time using a binary decision tree.

Define a structure called Node that can store some text, and pointers to two other Nodes. Once you have done this design a series of yes/no conversations by linking together the Nodes.

When complete, your chatbot should be able to quiz the user similar to this:



```
-----  
Chatbot v0.2  
-----  
  
Hello Friend. Are you happy today? yes  
Great. Listen, you want to know a fun fact? no  
Aaaaw. Come on, it's a good one? yes  
In 19th century Britain, sausages were marketed as "bags of mystery"  
  
Press any key to continue . . .
```



```
-----  
Chatbot v0.2  
-----  
  
Hello Friend. Are you happy today? no  
Maybe I tell you a fact that would cheer you up? yes  
In 19th century Britain, sausages were marketed as "bags of mystery"  
  
Press any key to continue . . .
```

Hints

- The lecture notes contain example code
- Remember that conversations don't have to be divergent. You can merge in and out like seen in the example above
- It might be worth defining a script / flow on paper so that it is easier to create the binary tree

Appendix

[Code Snippets](#) – Updated with example C code

[Online C Programming Resources](#)

[Complete C Reference Library](#)

C Programming IDE's

[Dev-C++](#) (Windows)

[Code::Blocks](#) (Windows, Mac, Linux)

[Visual Studio/C++ Express](#) (Windows)

[Netbeans C/C++](#) (Windows, Mac, Linux)

[Codelite](#) (Windows, Mac, Linux)