

IF184101 Basic Programming (IUP)

Quiz 2

Starting date: 06 November 2020
Deadline: 13 November 2020, 23:59 WIB. **Penalty: 0.15% of grade/minute of tardiness.**
Exam type: Open, Group (of at most three students)
Send to:
MM Irfan Subakti <yifana@gmail.com>
CC to Rosa Valentine Lammora <rosalammora@gmail.com>, Dicksen Alfarsius Novian <dicksenan@gmail.com>, James Rafferty Lee <jamesrafe10@gmail.com>, Michael Ricky <michaelricky007@gmail.com>, Clement Prolifel Priyatama <clement.085w@gmail.com> with the subject:
IF184101_BASPRO_IUP_Q2_StudentID1_Name1_StudentID2_Name2
File type and format: A zip file containing all of the .c source files & the declaration(s)
Filename format: IF184101_BASPRO_IUP_Q2_StudentID1_Name1_StudentID2_Name2.ZIP

Instruction

Please do these steps as in the following.

1. Please create a program, namely `01_linked.c`. The purpose of this program is to sort a singly linked list in ascending order, i.e., at least 5 items in the linked list. You can use any sorting algorithm, e.g., Bubble sort, Selection sort, Merge sort, etc. but not the library based sort (the function `sort()`). Once you have sorted the linked list, then insert an integer at the end of the linked list. **[33 points]**

Input

The integer number you want to insert to the linked list.

Output before the next insertion

The sorted elements in ascending order.

Input

An integer.

Final Output

The sorted elements in ascending order + the inserted integer.

2. Create a program, namely `02_tree_from_file.c`. The purpose of this program is to create a binary tree from a file, namely `02_file_input.txt` as in the following. [34 points]
- First-line in that file will be the type of the node, there are three types of node:
 - 0: means the node is the root (all file will be started by 0)
 - 1: (small L) means the node must be inserted at current's left node
 - uuuur: means the node must be inserted at the number of 'u' before this node's right node. In this example, 'uuuur', there are 4 u's before 'r'.
 - Next line is the data that need to be inserted the node

Example:

The file (`02_file_input.txt`):

```
0
1
1
2
1
4
ur
5
uur
3
```

The output will be:

Line 1: 0, this means it will be the root.

Line 2: 1 (number 1), this means the node will have 1 as its data.

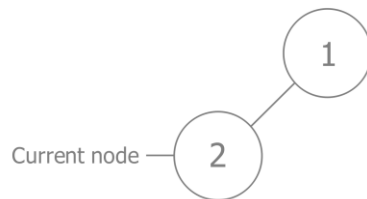
Tree:



Line 3: 1 (small 1); this means this node will be placed at the current node's left.

Line 4: 2; the data is 2.

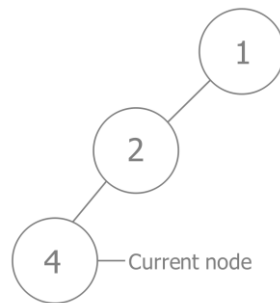
Tree:



Line 5: 1 (small 1); this means this node will be placed at the current node's left.

Line 6: 4; the data is 4.

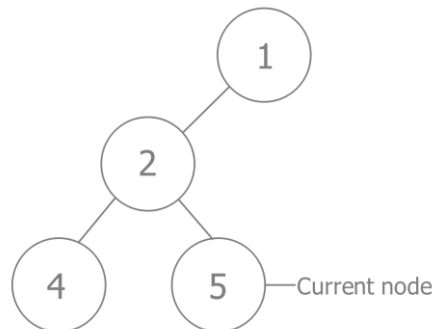
Tree:



Line 7: `ur`; this means from the current node; we need to go to one upper node which is node 2 and put the current node on its right.

Line 8: 5; the data is 5.

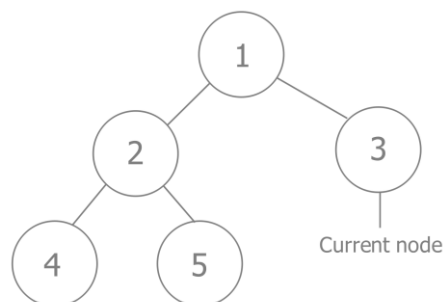
Tree:



Line 9: `uur`; this means from the current node; we need to go to two upper (2 u's) node which is node 1 and put the current node on its right.

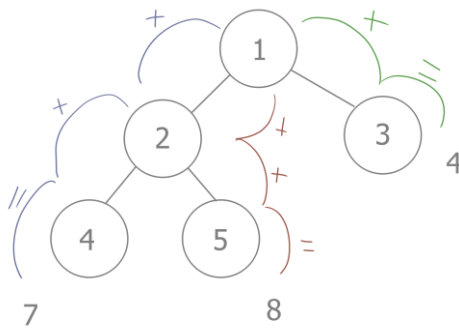
Line 8: 3; the data is 3.

Tree:



- c. The output will be the pre-order traversal of the built tree with exception:
- It will print = node/leaf: data.
 - At the leaf instead, the data print the sum of number to get to that leaf.

Rough illustration:



Example:

The output of tree above will be:

```
node: 1
node: 2
leaf: 7
leaf: 8
leaf: 4
```

The third output is 7 because it is calculated from 1+2+4

The fourth output is 8 because it is calculated from 1+2+5

The fifth output is 4 because it is calculated from 1+3

Hint: Each node will have three pointers instead of two.

3. Create a program, namely `03_chess_shortest_path.c`. The purpose of this program is to find the shortest path (minimum steps of the **horse** piece on chess). Given an 8×8 chessboard, minimum number of steps taken by a **horse** to reach *destination* from the current position. You may use any data structure for the dynamic memory allocation which has been discussed in our class. **[33 points]**

Example:

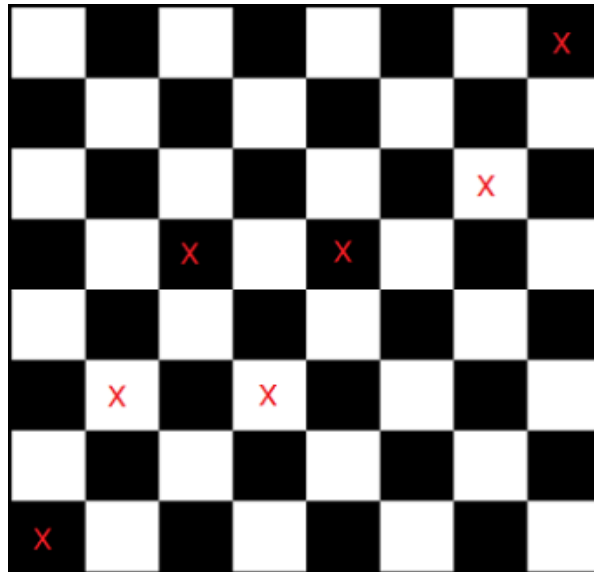
Input:

```
7, 0
0, 7
```

Note: 7,0 is the *current position* and 0,7 is the *destination*

Output:

```
step required: 6
```

Explanation:

4. To avoid plagiarism/cheating, every student needs to pledge and declare, then she/he must submit her/his **signed pledge and declaration** as in the following. Failed to do so will be resulted in getting 0 (zero) grade. Attach the **scanned/photo** of your *declaration* in your report.

“By the name of Allah (God) Almighty, herewith I pledge and truly declare that I have solved quiz 2 by myself, didn’t do any cheating by any means, didn’t do any plagiarism, and didn’t accept anybody’s help by any means. I am going to accept all of the consequences by any means if it has proven that I have been done any cheating and/or plagiarism.”

[Place, e.g., Surabaya], [date, e.g., 13 November 2020]

<Signed>

[Full name, e.g., Sekar Arum Lestari]

[StudentID, e.g., 502520xxxx]

Also to maintain fairness for your awarded grades, state clearly in your declaration, the percentage of works and the contribution(s) for each member in your group, e.g., Sekar Arum Lestari 33.33% [*state Sekar’s contribution(s) here*], Sri Lestari 33.33% [*state Sri’s contribution(s)*], and Ayu Larasati 33.33% [*state Ayu’s contribution(s) here*].

5. ZIP the files of 01_linked.c, 02_tree_from_file.c, 03__chess_shortest_path.c, and your declaration (e.g., Declaration(s).PDF) into 1 (one) only .ZIP file, namely IF184101_BASPRO_IUP_Q2_StudentID1_Name1_StudentID2_Name2.ZIP. Send this .ZIP file to yifana@gmail.com and CC-ed to rosalamhora@gmail.com, dicksenan@gmail.com & jamesrafe10@gmail.com.

6. Have a great day! Good luck! 😊