

CPP Problem Design Example

Subject: Elevator Simulator

Contributor: Yuan, Wei-Cheng

Main testing concept: Pointer

Basics

- C++ BASICS
- FLOW OF CONTROL
- FUNCTION BASICS
- PARAMETERS AND OVERLOADING
- ARRAYS
- STRUCTURES AND CLASSES
- CONSTRUCTORS AND OTHER TOOLS
- OPERATOR OVERLOADING, FRIENDS, AND REFERENCES
- STRINGS
- POINTERS AND DYNAMIC ARRAYS

Functions

- SEPARATE COMPILATION AND NAMESPACES
- STREAMS AND FILE I/O
- RECURSION
- INHERITANCE
- POLYMORPHISM AND VIRTUAL FUNCTIONS
- TEMPLATES
- LINKED DATA STRUCTURES
- EXCEPTION HANDLING
- STANDARD TEMPLATE LIBRARY
- PATTERNS AND UML

Description:

In this task, please write a program to simulate the operations of a simple elevator as stated in the following:

1. There is only one button at each floor to call the elevator to your floor and open the door. There are no other buttons inside the elevator for additional input commands.
2. When the elevator is going up to its destination,
 - (1) Pushing a button on a higher floor comparing to where the elevator currently is can make it keep going up and add the higher floor into its destination list. However, if the floor is already in the destination list, it is ignored.
 - (2) Pushing the button when the elevator is on the same floor would make the newly inputted command neglected.
 - (3) Pushing a button of a lower floor adds the floor to its destination list. The elevator first goes up to all higher destinations then goes down to the targeted floor. However, if the floor is already in the destination list, it will be ignored.
3. When it goes down to a destination,
 - (1) Pushing a button of a lower floor comparing to where the elevator currently is can make it keep going down and add the lower floor into its destination list. However, if the floor is already in the destination list, it is ignored.
 - (2) Pushing the button when the elevator is on the same floor would make the newly inputted command neglected.
 - (3) Pushing a button of a higher floor adds the floor to its destination list. The elevator first goes down to all lower destinations then goes up to the targeted floor. However, if the floor is already in the destination list, it will be ignored.

In this program, you need to design an elevator system that can display the position of the elevator during operation. We define each floor moving as one minute.

1. Our elevator goes up or down one floor in one minute, and the time duration of the opening and closing of doors is neglected.
2. You do not need to deal with conflicts when the destination and the current floor are both the same. In other words, you can just neglect the command.
3. The elevator will start at the first floor.
4. When pushing a button, the elevator does not move immediately. It will move at next minute.
5. If there is no new destination, it stops at the final destination.

Input:

You can use standard input to test your code. First, we input an integer **n** as the count of total operations. Then the next **n** lines each indicates the time of the operation in minutes and which

floor the button is pushed, separated by a space. Additionally, the order of operations is not necessarily listed in numerical order. The input ends with EOF(end of file).

Output:

List out the location of the elevator in which floor at every minute.

Error Handling:

Sample Input / Output :

Sample Input	Sample Output
5	1 1
2 10	2 1
3 6	3 2
8 2	4 3
15 4	5 4
20 8	6 5
	7 6
	8 7
	9 8
	10 9
	11 10
	12 9
	13 8
	14 7
	15 6
	16 5
	17 4
	18 3
	19 2
	20 2
	21 3
	22 4
	23 5
	24 6
	25 7
	26 8

- ☐ Easy,Only basic programming syntax and structure are required.
- ☒ Medium,Multiple programming grammars and structures are required.
- ☐ Hard,Need to use multiple program structures or complex data types.

Expected solving time:

20 minutes

Other notes: