

## LIFT SIMULATION

---

In this assignment, you are required to simulate the process of lift operation using C program.

Please refer to [http://www.cs.swarthmore.edu/~newhall/unixhelp/c\\_codestyle.html](http://www.cs.swarthmore.edu/~newhall/unixhelp/c_codestyle.html) for the C coding style to write good-format and easy-understanding code.

There are in total 4 tasks, for each task, you need to carefully read the task description and finish them before deadline. The time and mark allocation are as follows:

1. Task 1: 1 week, 5 marks (Deadline: 2 Feb, 11:59pm, 2015)
  - a. Pass all test case (2 marks)
  - b. deallocates the memory after use (1 mark)
  - c. Good coding style (1 mark)
  - d. Easy-understanding comment (1 mark)
2. Task 2: 2 weeks, 6 marks (Deadline: 16 Feb, 11:59pm, 2015)
  - a. Pass all test case (2 mark)
  - b. Reuse the function of task1 (1 mark)
  - c. deallocates the memory after use (1 mark)
  - d. Good coding style 1 mark (1 mark)
  - e. Easy-understanding comment (1 mark)
3. Task 3: 2 weeks, 7 marks (Deadline: 2 Mar, 11:59pm, 2015)
  - a. Pass all test case (4 mark)
  - b. deallocates the memory after use (1 mark)
  - c. Good coding style 1 mark (1 mark)
  - d. Easy-understanding comment (1 mark)
4. Task 4: 4 weeks, 12 marks + 5 marks (bonus) (Deadline: 30 Mar, 11:59pm, 2015)
  - a. Pass all test case (6 mark)
  - b. deallocates the memory after use (1 mark)
  - c. Good coding style 1 mark (1 mark)
  - d. Easy-understanding comment (1 mark)
  - e. Good-writing report (3 marks, the report requirement is listed in Task4)
  - f. One new idea and its implementation worth one mark (total bonus marks cannot exceed 5)

## TASK 1

---

### TASK 1 STATEMENT

---

In this task, you are required to write a program **liftSimulation\_task1.c** to print a static image of the lift. This image contains one building and a lift. But instead of real-life object, we use some notations to represent them.

Notation:

"_"	floor
" "	Wall
"[_]"	Lift (2 '_' between '[')'
">"	Current position of the lift

Assume the building has 5 floors and has a width of 12 unit length. The lift is 4 unit length wide. You are required to "draw" the building with the lift using the above notations, given the current position of the lift. In addition, you need to add one more line using the connection of "-" to represent roof of the building.

More specifically, you are required to implement the following function:

```
char** staticLift(int cur_pos)
```

Input:

cur\_pos : current position of the lift

Output:

A two-dimension array that record the static image.

### EXAMPLE

---

```
char** output = staticLift(3);
```

The output array is a 6\*12 array.

```
Char output[6][12]=
```

```
{{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
```

```
{'| ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '|},
```

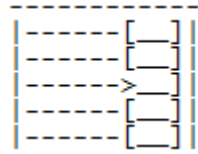
```
{'| ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '|},
```

```
{'| ',' ',' ',' ',' ',' ',' '>',' ',' ',' ',' '|},
```

```
{'| ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '|},
```

$$\{ '|', '-', '-', '-', '-', '-', '-', '-', '[', '-', '-', ']', '|', '| \} \};$$

When the array is printed in the screen, it should looks like this:



## SAMPLE TEST DATA

- Input file: task1.in
- Output file: task1.out

You can download the above files from IVLE. Please print out the lift format following the output example.

## TASK 2

## TASK2 STATEMENT

In this task, you are required to simulate the running of the lift, i.e., the process of lift's moving from one floor to another floor. Specifically, you are required to output all the frames (i.e., arrays) of the running lift, where each frame represents the static image of the building with the lift in certain floor (you have done this in Task 1). You are required to reuse the function you designed in task 1.

To better understand this issue, let's see the example first.

All the notations and assumptions remain the same as Task 1. You need to implement the following function:

Function:

```
char*** moveLift(int cur_pos, int des)
```

Input:

1. cur\_pos: current position of the lift
2. des: destination floor

Output:

A three-dimension array that record the lift at each moving step. Each two-dimension array record one step of the lift, with the first array representing the initial state and the last array representing the destination state.

### EXAMPLE

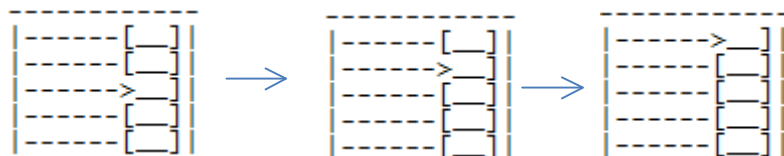
```
char*** output = moveLift(3, 5);
```

The output array is a (5-3+1)\*6\*12 array.

```
char*** output=
```

```
{{{'-','-','-','-','-','-','-','-','-','-','-'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','>','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'}},
{{{'-','-','-','-','-','-','-','-','-','-','-'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','>','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'}},
{{{'-','-','-','-','-','-','-','-','-','-','-'},
{'|','-','-','-','-','-','-','>','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'},
{'|','-','-','-','-','-','-','[','_','_','_'],'|'}}};
```

When the array is printed in the screen, it should look like this:



To make the process looks more clearly here, I list each image one by one as a sequence. But in your terminal, after printing one image, you should wait for a while, then clear the screen and print the following images. So in this way it looks more like animation.

## SAMPLE TEST DATA

- Input file: task2-eg1.in | task2-eg2.in

- Output file: task2-eg1.out | task2-eg2.out

You can download the above files from IVLE. Please print out the lift format following the output example.

---

## TASK 3

---

---

### TASK3 STATEMENT

---

When the animation of the lift operation is done, we can invite a person to use the lift.

The person shall be placed on a given floor, and gives out a command to reach another floor.

Given the following parameters:

1. current position of the lift
2. current position of the person
3. destination floor

You are required to show the animation of the following process:

1. The lift arrives at current floor of the person
2. The person enters the lift
3. The lift moves to the destination floor
4. The person leaves the lift

Here we use “P” to represent the person. Other notations and assumptions remain the same as Task 1. You are required to implement the following function:

Function:

```
char*** peopleUseLift(int cur_pos_lift, int cur_pos_people, int des)
```

Input:

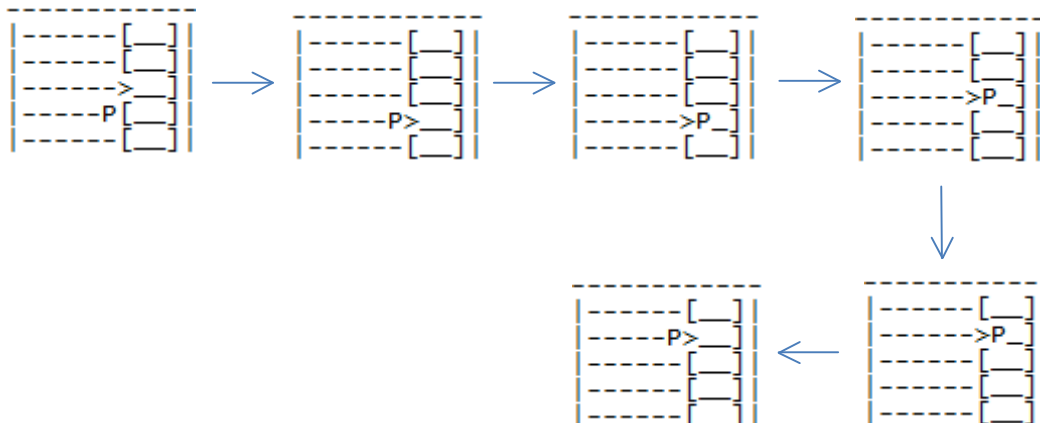
1. cur\_pos\_lift: current position of the lift
2. cur\_pos\_people: current position of the people
3. des: destination floor

Output:

A three-dimension array that record the lift and people at each moving step. Each two-dimension array records one step of the lift, with the first array representing the initial state and the last array representing the destination state.



When the array is printed in the screen, it should look like this:



## SAMPLE TEST DATA

- You can download the above files from IVLE. Please print out the lift format following the output example.

## TASK 4

**In Task 4, we will do lift scheduling. Details will be released to you after you finish the previous three tasks.**