Lab 3: Deque

Professor Patt thinks queue and stack are both useful data structures, but combining them into one structure must be better. A list supporting *pop* and *push* on both sides sounds great. Your job is to help Professor Patt to implement such a list, which is called *deque* formally.

Implementation Details

- You are required to write in **LC-3 assembly language**.
- Your program should start at x3000.
- Use instrucion `TRAP x20` (GETC) to get an input char. Do not use instruction `TRAP x23` (IN). Use instrucion `TRAP x21` (OUT) to output a char or use instrucion `TRAP x22` (PUTS) to output a string.
- The list should support the following operations:
    - `+s`: push `s` to the left side
    - `-`: pop a char from the left side and print it
    - `[s`: push `s` to the right side
    - `]`: pop a char from the right side and print it

  The list is empty at first, and `s` is a char in `a-z` and `A-Z`. When the list is empty and pop, just print a `_` as the result. Here is an illustration:

Operation	List	Output
-	(empty at first)	_
+a	a	
+D	Da	
+d	dDa	
+o	odDa	
+A	AodDa	
]	AodD	a
[b	AodDb	
-	odDb	A
]	odD	b

- Sample input and output:

```
-+a+D+d+o+A][b-]
_aAb
```

  Please note that input char is **echoed** in screen. Only after you press `ENTER`, the answer will output.

- Remember to halt your program in the end.

Limitation

- the length of input string: $0 \le l \le 100$
- no overflow

Grading

Lab 3 takes **6%** of the total score, consisting of Check part (50%) and Report part (50%).

Check Part

- First upload your code to Learning in ZJU, then find a TA to check your code in person. TAs will first test the correctness of your program, then ask you some questions to make sure you understand what you code but not cheat.
- You can try again if you fail in checking, but there will be a penalty of -10% (of checking part) for each try.
- We strongly suggest you to make a thorough test by yourself before checking.
- We strongly suggest you to write enough comments in your code so that you will be aware of what's going on in your program and confident to answer TA's questions.

Report Part

- Report must be written **in English**, concise and carrying main ideas. Try to use the report to convince TAs that you complete the task by yourself.
- Your lab report should contains the following contents:
  - Algorithm. Flowchart or Pseudocode is prefered. The complexity of your algorithm will not affect your score.
  - Essential parts of your code with sufficient comments. Please only select the most important code phases and explain them.
  - Questions that TA asked you, and Answers.
- **No more than 2 A4 pages.** No template provided. Be sure to make it readable.

Penalty

- **Wrong Answer**: -10% of Check part each time.
- **Delay**: -20% of the corresponding part per day.
- **Cheating**: -100% of this lab. Additionly, -10% of the final score of this course. **Please note that uploading your answer to the Internet is also CHEATING!!!**