Homework

- **Write a program that simulates the Undo function of editors (Office, Text Editor, ..) as follows. The program uses two stacks: A stack contains integers and a stack contains elements that are strings and have the following menu interface :**
  - **Add 1 integer to the stack**
  - **View top**
  - **Remove from the stack contains integers 1 element**
  - **Undo**
- **Hint: To set the undo function, we save the name of the operations just done on the integer stack in string stack (except the View Top operation because it does not change the data). The program based on the last action that was taken to do the opposite.**
- **For example :**
  - **PUSH 20 -→ UNDO = POP**
  - **POP 15 → UNDO PUSH 15**

Homework: Process management in OS

- **Simulate a computer that process computing request from OS's programs.**
- **Configuration Input:**
  - **Number of parallel process it can run**
  - **Memory capacity**
- **Program has the menu:**
  - **Create new program (with a given amount of necessary memory and ID)**
  - **Kill a program**
  - **Show the status of running and waitting processes.**

GUI example

- number of parallel process: 2
- memory capacity (MB): 100
- 1. Create new program run 1st time:
  - the memory size of program? 40
  - Program ID? 1 → successful. Process created

  2. Show the status:

ID	Memory
1	40

  1. Create new program run 2th time:
  - the memory size of program? 70
  - Program ID? 2 → in Queue as there is not enough memory (60 vs 70)

  1. Create new program run 2th time:
  - the memory size of program? 50
  - Program ID? 3 → successful. Process created

  2. Show the status:

ID	Memory	Queue
1	40	
3	50	
2		70

- 3. Kill process: Process ID ? 1 - Success

GUI example

2. Show the status:

ID	Memory	Queue
3	50	
2		70

1. Create new program run 3st time:
- the memory size of program? 60
- Program ID? 4 → in Queue

3. Kill process: Process ID ? 3 – Success

2. Show the status:

ID	Memory	Queue
2	70 (vì bộ nhớ trống đã đủ, tiến trình từ queue sẽ vào bộ nhớ)	
4		60