

Stack library.

Implement a program which provides a stack (LIFO) data structure, made by an array of 20 elements at most.

The library shall provide the following interface:

1. `bool push(integer elem_in):`
Add an element at the first available position. (after the last occupied one)
2. `integer elem_out = Pop():`
Remove and return the integer element in the last occupied position
3. `bool sort():`
Implement the bubble sort algorithm over array elements.
4. `bool search(integer elem_in):`
Implement a binary search over array elements.

Every call should place in output the following port configuration:

	<i>P0</i>	<i>P1</i>	<i>P2</i>	<i>P3</i>
Push	0x11	# of elem in stack	elem_in	Transition from 0x00 to 0xFF when P0, P1, P2 are valid.
Pop	0x22	# of elem in stack	elem_out	Transition from 0x00 to 0xFF when P0, P1, P2 are valid.
Sort	0x33	# of elem in stack	0xFF	Transition from 0x00 to 0xFF when P0, P1, P2 are valid.
Search	0x44	# of elem in stack	0x00 if not found; 0xFF if found;	Transition from 0x00 to 0xFF when P0, P1, P2 are valid.
Exit	0x00	0x00	0x00	0x00

In addition implement a simple main which using stack functions (possibly implemented in a separate file).

The main shall:

5. creates a stack
6. pushes at least 5 elements,
7. sort the stack,
8. search for a value contained in the stack and one not contained,
9. pop all inserted elements.

An output example is the following:

~/workspaces/i8051/ISASim/Release/ISASim main.hex

```
P0    P1    P2    P3
0xFF  0xFF  0xFF  0xFF
0xFF  0xFF  0xFF  0x00 // Init condition
0x11  0xFF  0xFF  0x00
0x11  0x01  0xFF  0x00
0x11  0x03  0x03  0x00
0x11  0x03  0x03  0xFF // push(stack,3) was executed. Note P3 = 0xFF
0x11  0x03  0x03  0x00
0x00  0x00  0x01  0x00
0x00  0x00  0x00  0x00 // exit condition
```

Queue library

Implement a program which provides a queue (FIFO) data structure, made by an array of 20 elements at most.

The library shall provide the following interface:

1. `bool push(integer elem_in):`
Add an element at the first available position. (after the last occupied one)
2. `integer elem_out = pop():`
Remove and return the integer element in the first position,
Consequently moves all the other elements.
3. `bool sort():`
Implement the selection sort algorithm over array elements.
4. `bool search(integer elem_in):`
Implement a linear search over array elements.

Every call should place in output the following port configuration:

	P0	P1	P2	P3
Push	0x11	# of elem in queue	elem_in	Transition from 0x00 to 0xFF when P0, P1, P2 are valid.
Pop	0x22	# of elem in queue	elem_out	Transition from 0x00 to 0xFF when P0, P1, P2 are valid.
Sort	0x33	# of elem in queue	0xFF	Transition from 0x00 to 0xFF when P0, P1, P2 are valid.
Search	0x44	# of elem in queue	0x00 if not found; 0xFF if found;	Transition from 0x00 to 0xFF when P0, P1, P2 are valid.
Exit	0x00	0x00	0x00	0x00

In addition implement a simple main which using queue functions (possibly implemented in a separate file).

The main shall:

5. creates a queue
6. pushes at least 5 elements,
7. sort the queue,
8. search for a value contained in the queue and one not contained,
9. pop all inserted elements.

An output example is the following:

~/workspaces/i8051/ISASim/Release/ISASim main.hex

```
P0    P1    P2    P3
0xFF  0xFF  0xFF  0xFF
0xFF  0xFF  0xFF  0x00 // Init condition
0x11  0xFF  0xFF  0x00
0x11  0x01  0xFF  0x00
0x11  0x03  0x03  0x00
0x11  0x03  0x03  0xFF // push(stack,3) was executed. Note P3 = 0xFF
0x11  0x03  0x03  0x00
0x00  0x00  0x01  0x00
0x00  0x00  0x00  0x00 // exit condition
```