

School of Computer Science and Software Engineering  
 Clayton Campus, Monash University

**CSE1303 Part A**  
**Summer Semester, 2002**

**Tutorial 2: Stacks and Queues**  
**Solutions**

**Exercise 1.**

top:	<input type="text" value="0"/>	entry:	<input type="text" value="1.2"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="1"/>	entry:	<input type="text" value="1.2"/>	<input type="text" value="1.0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="0"/>	entry:	<input type="text" value="1.2"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="-1"/>	entry:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="0"/>	entry:	<input type="text" value="0.2"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="1"/>	entry:	<input type="text" value="0.2"/>	<input type="text" value="4.9"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="2"/>	entry:	<input type="text" value="0.2"/>	<input type="text" value="4.9"/>	<input type="text" value="5.1"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="1"/>	entry:	<input type="text" value="0.2"/>	<input type="text" value="4.9"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="0"/>	entry:	<input type="text" value="0.2"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="1"/>	entry:	<input type="text" value="0.2"/>	<input type="text" value="10"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="0"/>	entry:	<input type="text" value="0.2"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
top:	<input type="text" value="-1"/>	entry:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
y:	<input type="text" value="2"/>						

**Exercise 2.**

```
/* Return the top element of the stack */
float
topElement(const StackPtr* stackPtr)
{
    if (stackEmpty(stackPtr))
    {
        fprintf(stderr, "Stack is empty\n");
        exit(1);
    }

    return stackPtr->entry[stackPtr->top];
}
```

**Exercise 3.**

```
/* Print all the elements in the stack top down */
void
printStack(const Stack* stackPtr)
{
    int i;

    for (i = stackPtr->top; i >= 0; i--)
    {
        printf("%d\n", stackPtr->entry[i]);
    }
}
```

**Exercise 4.**

```
/* Return the size of the stack */
int
stackSize(const Stack* stackPtr)
{
    return stackPtr->top + 1;
}
```

## Exercise 5.

count:	1	front:	0	rear:	0	entry:	10				
count:	2	front:	0	rear:	1	entry:	10	5			
count:	3	front:	0	rear:	2	entry:	10	5	2		
count:	4	front:	0	rear:	3	entry:	10	5	2	1	
count:	3	front:	1	rear:	3	entry:		5	2	1	
count:	4	front:	1	rear:	4	entry:		5	2	1	9

x: 9

## Exercise 6.

```

/* Print all the elements in the queue front to rear */ void
printQueue(const Queue* queuePtr)
{
    int index = queuePtr->front;
    int i;

    for (i = 0; i < queuePtr->count; i++)
    {
        printf("%f\n", queuePtr->entry[index]);
        index = (index + 1) % MAXQUEUE;
    }
}

```

**Exercise 7.**

```
/*
 * Reverse all the entries in a queue.
 */
void
reverse(Queue* queuePtr)
{
    Stack theStack;

    initializeStack(&theStack);

    while(!queueEmpty(queuePtr))
    {
        push(&theStack, serve(queuePtr));
    }

    while (!stackEmpty(&theStack))
    {
        append(queuePtr, pop(&theStack));
    }
}
```

**Exercise 8.**

- a) 8 7 + 3 2 + \* 4 3 / +
- b) 3 4 - 2 + 7 \* 1 +
- c) (5 + 3) \* (4 - 7) / 2
- d) 7 \* (2 + 3) - 5