

CST 370
Homework (Stacks)

1. **(20 points)** Consider two stacks each of size 10. When you pop an element from the first stack you multiply it by 2 and add it to the second stack. When you pop an element from the second stack you multiply it by 3 and add it to the first stack.

Push numbers 1 to 5 to the first stack in order. Push numbers 11 to 15 to the second stack in order. Pop two numbers from the first stack (remember these numbers are going to be added to the second stack). Pop three numbers from the second stack (remember these numbers are going to be added to the first stack).

- a) What is the value in the top of the first stack?
- b) What is the value at the top of the second stack?
- c) What is the current size of the first stack?
- d) What is the current size of the second stack?

2. **(20 points)** Draw the stack after the following code executes

```
Stack s;  
s.push(60);  
s.push (25);  
s.push(50);  
s.pop( );  
s.pop( );
```

3. **(20 points)** Draw the stack after the following code executes

```
Stack s;  
s.push (44);  
s.push (33);  
s.push (22);  
while (!s.empty())  
    s.pop();
```

4. **(20 points)** Draw the stack after the following code executes

```
Stack s;  
for (int i = 1; i <= 5; i++)  
    s. push (2*i);
```

5. **(20 points)** Suppose that some application requires using two stacks whose elements are of the same type. A natural storage structure of such a two-stack data type would consist of two arrays and two top pointers. Explain why this may not be a space- wise efficient implementation.