Stacks and Queues

1. Write a function `removeUntil()` that removes all values from a stack of integers until but not including the first occurrence of a given value. The function accepts two parameters: a reference to the stack and the value to stop at. The function definition is as follows:

```
def removeUntil(stack, value):
```

   Given a stack [1 2 3 4 5 6 5 4 3 2 1] with the topmost number displayed on the left, calling `removeUntil()` with value = 5 will produce the stack [5 6 5 4 3 2 1].

2. Write a recursive function `recursiveReverse()` that reverses the order of items stored in a queue of integers. The function accepts a single parameter: a reference to the queue. The function definition is as follows:

```
def recursiveReverse(queue):
```

3. Write a function `palindrome()` that determines whether a given string is a palindrome. The function accepts a single parameter: the word (a string). The function should return 0 if the string is a palindrome and -1 otherwise. The function should ignore whitespace, case, and punctuation. The function definition is as follows:

```
def palindrome(word):
```

   Sample output:

```
Enter a string: A man a plan a canal Panama The
string is a palindrome.

Enter a string: Superman in the sky The
string is not a palindrome.
```

4. Write a function `balanced()` that determines if an expression comprised of the characters ()[]{} is balanced. The function accepts a single parameter: the expression (a string). The function should return 0 if the expression is balanced and -1 otherwise. The function definition is as follows:

```
def balanced(expression):
```

   The following expressions are balanced because the order and quantity of the parentheses match:

```
()
([])
{[]()[]}
```

   The following expressions are not balanced:

```
{()]
[({()])
```