More Stack Questions

1. Create a stack that contains the letters of your favourite movie or anime. Add your last name or repeat the letters until you have pushed 12 letters. You will need to push each letter separately. Print the Stack on the screen using a loop.
2. Find the minimum element in the stack.
3. Print the number of times that ‘E’ or ‘e’ appears in the stack.
4. Print the Stack again, but in the reverse order of question #1.
5. Print the bottom element in the stack. THEN, print top element in the stack. Use loops to do this.

Starter Code

There are two classes: StackQuestions_2 and StackChar. Put them in different files.

```
public class StackQuestions_2 {
	public static void main(String args[]) {
		new StackQuestions_2();
	}
	public StackQuestions_2() {
		//To do: Make a new Stack, add 12 letters from favourite movie/anime.

		//#1 Print every item in the Stack, using a loop
		System.out.println("#1 - Print every item in the Stack using a loop:");

		//To do: re-add 12 letters from fav movie/anime to the Stack

		//#2 Find the minimum
		System.out.println("#2 - Find the minimum:");

		//To do: re-add 12 letters from fav movie/anime to the Stack

		//#3 Print number of times E or e is in Stack
		System.out.println("#3 - How many times does E or e appear?");

		//To do: re-add 12 letters from fav movie/anime to the Stack

		//#4 Print Stack in opposite order it printed in #1
		System.out.println("#4 – Print Opposite");

		//To do: re-add 10 letters from your name to the Stack
		//#5 Print bottom element, then print top, use loop
		System.out.println("#5 – Print Bottom, Print Top (Using Loop)");

	}
}
```

A Char Stack Class

```
public class StackChar
{
   private int count;
   private char data[] = new char [50];
   public StackChar ()
   {
      count = 0;
   }

   public void push (char addMe)
   {
      data [count] = addMe;
      count++;
   }

   public int size ()
   {
      return count;
   }

   public boolean isFull ()
   {
      return (count == 50);
   }

   public char pop ()
   {
      count--;
      return data [count];
   }

   public char peek ()
   {
      return data [count-1];
   }

   public boolean isEmpty ()
   {
      return count == 0;
   }

   public void clear ()
   {
      count = 0;
   }
}
```