/*----------DESIGN EXPLANATION Q6----------*/

This program is made with one class "ReverseStr". This class holds a function that takes a string and reverses it using stack. In Main function, it creates a file with Chuck Norris' jokes and reads the file to provide the string to the "reverse" function

ReverseStr:

In this program, we aim to reverse a passed in string.
To do that, the Main function creates a file and reads the file to provide the string to the "reverse" function.

Our "reverse" function takes a string and creates a stack. With member functions that belong to stack, we are able to insert, remove in the order that it mimics a stack. As we want to use the last in first out fashion that belongs to stack, we insert in first in order, then print out the last (outer) character using pop. The string is inserted first-in order but the removal happens on the top-outer element.

Method:

```
void reverse(string & str, int num)
```

```
int main(){

    ReverseStr r1;

    ofstream myfile;
    myfile.open("jokes.txt");//creates a file and save below text
    myfile << "If Chuck Norris were to travel to an alternate dimension in which there was another Chuck Norris and they both fought, they would both win..";

    myfile.close();

    ifstream myfile1("jokes.txt");//reads the text file
    string line;

    if (myfile1.is_open()) {
        while (myfile1>> line) {
            r1.reverse(line, line.length());
            cout<< line<< " ";
        }
        myfile.close();
    }

    return 0;
}
```

//the file is saved and passed into the function as a string until the end of the file

```
class ReverseStr {// class to reverse a passed in string

public:
    void reverse(string & str, int num) {//this function takes a string array and uses stack functions to add and remove elements
        stack<char> stk;
        for (int i = 0; i < num; i++) {
            stk.push(str[i]);//inserts element
        }
        int k = 0;
        while (!stk.empty()) {
            str[k++] = stk.top();//assigns next element
            stk.pop();//removes top element---->the string was inserted first-in order but the removal happens on the top-outer element
        }
    }

};
```

//the passed in string is broken down and gets saved as a stack in first in order. It pops outer-most character and the program prints in reversed order

```
fI kcuhC sirroN erew ot levart ot na etanretla noisnemid ni hcihw ereht saw reht
ona kcuhC sirroN dna yeht htob ,thguof yeht dluow htob ..niw
```