

1. Имплементирайте рекурсивните решения на следните задачи чрез стек:
 - a. Принтирайте числата от n до 0 на нов ред
 - b. Принтирайте числата от 0 до n на нов ред
 - c. Напишете функция, която намира сумата на числата в масив с размерност n
 - d. Напишете функция, която намира най-малкото число в масив с размерност n
2. Реализирайте клас CompositeStack (последователна реализация), който да поддържа два отделни стека. CompositeStack трябва да поддържа следните методи като минимум:
 - a. bool push1(T&)
 - b. bool pop1(T&)
 - c. bool peek1(T&)
 - d. bool push2(T&)
 - e. bool pop2(T&)
 - f. bool peek2(T&)
3. Реализирайте клас UndoRedoStringBuilder, чиято цел е да може да строи стринг и да изпълнява операциите undo & redo, които мутират състоянието на стринга. Класът трябва да съдържа следните методи като минимум:
 - a. UndoRedoStringBuilder(const char* initialString)
 - b. void append(const char* appendString)
 - c. char* getResult()
 - d. void deleteLastNCharacters(int n)
 - e. void deleteFirstNCharacters(int n)
 - f. void replace(const char* searchString, const char* replacementString)
 - g. void undo()
 - h. void redo()

*по ваше желание може да използвате string вместо char *

```
UndoRedoStringBuilder sb("This is a sample text for the sample program filled with sample
string parts called sample...");
sb.append("<= no more samples here :)");
cout<<sb.getResult();//This is a sample text for the sample program filled with sample string
parts called sample...<= no more samples here :)
sb.replace("sample","cool");
sb.deleteFirstNCharacters(8);
cout<<sb.getResult();//a cool text for the cool program filled with sample string parts called
cool...<= no more cools here :)
sb.undo();
cout<<sb.getResult();//a sample text for the sample program filled with sample string parts
called sample...<= no more samples here :)
sb.deleteLastNCharacters(80);//remove all letters after "a sample text" :)
cout<<sb.getResult();//a sample text
sb.redo();
cout<<sb.getResult();//a sample text
sb.undo();
```

```
cout<<sb.getResult();//a sample text for the sample program filled with sample string parts  
called sample...<= no more samples here :)
```