

Programming 2 – Laboratory 7 - U2T1

Implement given `MyText` and `MyStack` classes.

Uncomment given `main()` function, do not change anything in the code.

Part 1 – 2p

Add to class `MyText` missing constructors, destructor and operators.

Use there the given `init()` function which allocate sufficient number of characters. `init()` takes as first parameter a number of characters, to be copied from second source address. The function will allocate additional memory for `'\0'`. In case of empty text one element holding null terminating character `'\0'` is allocated.

Part 2 – 4p

Implement class `MyStack`, add constructors, destructor and operators.

Stack store some number of `MyText` type elements.

Member field `size` stores size of the memory available in stack array; member field `tos` stores 'top of the stack' indicator ie. number of elements placed at stack.

Implement `push()` and `pop()` methods. Method `push()` add element at the top of the stack and increment `tos` value. When element is added while all memory in array is used (the stack is full) then the stack size is doubled (i.e. twice of current memory is allocated for stack).

`pop()` method removes element form the top of the stack and decrease `tos` value, not affect stack memory, just `tos` index.

Part 3 – 2p

Write method `sort()` sort `MyText` on stack by length and return a stack with sorted content by length of the text.

Example output below (next page):

----- Part 1 -----

Sentence 1 : (empty sentence)

Sentence 2 : The drone on Mars.

Sentence 3 : The drone on Mars.

Sentence 1 = Sentence 2 : The drone on Mars.

PS : C++ is fun!

----- Part 2 -----

After stack test

Pop copy:

Good luck!
(empty sentence)
What is your name?
To be, or not to be.

Write in console 3 lines of text (79 characters max):

test

it

now

--- Size = 4

Pop your sentences:

now

it

test

----- Part 3 -----

Pop your sentences sorted:

test

now

it

Bye