Задачи

Увод в програмирането

29 октомври 2018 г.

Правила

- **Правилното декомпозиране на фунции е много важно. Вашите функции трябва да са кратки (по-малко от 20 реда!) И всеки трябва да изпълнява една единствена, ясна задача.**

- **Имената на променливи трябва да са описателни – да обясняват за какво служи дадената променлива. Примерно за име на човек подходящо име е personName, а неподходящо име е ~~a1~~.**

- За тези задачи не може да ползвате фунцкии (от cstdlib или cstring) за работа с низове

Задачи

1. “WordCount”
   Ако ви е даден на стандартния вход низ. Изкарайте колко символа (неговата дължина) и колко думи има в него

   Input: “this is a string” *Output:* Number of characters is 18. Number of words is 4

2. “*StringToNumber*”
   От стандартния вход четете стринг. Превърнето го в число или принтрайте грешка ако не е възможно. Направете фунция char_to_int която да изпълнява фунционалноста (тя връща превърнатото число от тип int, или 0 ако не може.)

   Input: “12” *Output:* “12 is valid”, *Input*: “-12” *Output:* “-12 is valid”,
   Input: “12foo” *Output:* “12 is valid”, *Input:* “foo12” *Ouput* “Not valid number or zero”

   a. Разширете фунцията да подържа числа в 16-тична бройна система и в осмична(8). Числа които започват с 0x са в 16-тична, а които започват с 0 в 8-чна.

   Input “0x12” *Output*: 18
   Input “012” *Output*: 10
   Input “12” *Output*: 12

3. “*String Compare”*
   Напишете фунция която сравнява два низа лексикографски (започваме едновременно поелементно преглеждане на двата низа в посока от ляво на дясно и сравняваме символ по символ). Отрицателна стойност означава, че първият низ е лексикографски по-малък от втория, нула означава, че двата низа са еднакви, а положителна стойност означава, че първия низ е лексикографски по-голям.

   Input “abcd”, “abcz” *Output*: -1
   Input “abcd”, “ab” *Output*: +1

Input “ab”, “ab” *Output*: 0

4. “*Find*”
   Напишете фунция която намира първото срещане на низ в друг низ, връща индекса на началото на намерния низ или -1 ако няма такъв. Фунциията трябва да приема незадължителен параметър от кой индекс да започне търсенето (под разбиране е 0)

   Input: “a substr and another substr” “substr” *Output*: 2
   Input: “nothing here” “substr” *Output*: -1

5. “*Replace*”
   a. Напишете фунция replace, която сменя всички срещания на даден char със друг.
      Програмата ви взема от стандартния вход три параметра (низ, и двата символа) и вика фунцията, която трябва да “върне” нов модифициран низ .

      Input: “chars to replace” “ “ “*” *Output*: “chars*to*replace”

   b. Напишете фунция replace, която сменя всички срещания на даден низ със друг.
      Използвайте фунцията която сте направили в задача 4) (Hint: направете си фунция copy_string между два индекса)

      Input: “string to replace string” “string“ “********” *Output*: “******** to replace ********”

6. *“Caesar cipher”*
   Шифър на Цезар представлява кодиране на всеки символ от изречението с друг, като към оригинала се прибавя избрано число - **ключ**, например 2
   Да се направи програма която кодира **или** декодира даден низ.
   Помислете върху потребителски интерфейс на програмата ви - Т.е примерно трябва да може да се каже на програмата дали да декодира или кодира когато се вика, и да се подаде ключа
   Помислете как ще декомпозирате фунциите. Примерно фунции които декодират и кодират трябва да са отделени, тези които са за потребителски интерфейс също отделно.

7. “*Memory Layout”*
   Принтирайте адреса в памета на примерни променливи:
   статични, глабални, локални
   всички тези отгоре като константа и като не.
   всички тези отгоре инциализиране и не инциализиране.
   (общо 12 променливи)
   Например static const int static_const_initialized = 1; // статична константа инициализирана.
   Да вземете адреса използвайте оператор “&” т.е
   std::cout << &variable_name // адреса ще бъде принтиран в 16 бройна система
   Може те ли да направите изводи ? Кои променливи имат близки адреси и кои не.
   Направете (може на хартия) приблизителна схема на памета която виждате. Кои имат високи адреси, кои ниски.
   https://en.wikipedia.org/wiki/Data_segment#Program_memory