



Първо домашно за поправката

*курс Увод в програмирането
за специалност Информатика
зимен семестър 2017/18 г.*

Задача 1А

Напишете програма, която въвежда от потребителя символен низ и завърта някои от думите в него.

Програмата трябва да използва речник, който съдържа думите, които ще се завъртат. Той трябва да се представи като двумерен масив от тип `char`. Когато програмата започне работа, тя трябва да въведе от потребителя броя на думите в речника. След това тя трябва да задели динамична памет за масива от думи и да въведете самите думи. Всяка дума ще е с дължина не по-голяма от 40 символа.

Програмата трябва също да въведе от потребителя символен низ с дължина до 80 символа. Низът може да съдържа произволни символи, включително интервали и табулации. Програмата трябва да завърти наопаки всички думи в низа, които се съдържат в речника. Останалите думи трябва да останат непроменени. Думите в низа се разделят една от друга с един или повече празни символи (`whitespace`) -- интервали, табулации и т.н. Завъртането трябва да стане на място в низа, без да се използва допълнителен масив.

По-долу е даден пример за работата на програмата. Забележете, че се завъртат само цели думи. Например програмата завърта низа `abc`, защото той се среща в речника, но `abcsabc` трябва да остане непроменен.

```
Enter dictionary size> 4
Enter dictionary words> abc zs34 _1_2! ads
Enter a string> abcdef      abc  abcabc _1_2_! 34          zs34  abc
abcsabc      cba  abcabc !_2_1_ 34          43sz  cba
```

Задача 1Б

Напишете програма, която въвежда от потребителя символен низ, намира определени числа в него и извежда сумата им.

Програмата трябва да използва речник, в който се съдържат числата, които ще се търсят в низа. Той трябва да се представи като масив от тип `unsigned short`. Когато програмата започне работа, тя трябва да въведе от потребителя броя на числата в речника. След това трябва да заделите динамична памет за масива и да въведете самите числа.

Програмата трябва също да въведе от потребителя символен низ с дължина до 80 символа. Низът може да съдържа произволни символи, включително интервали и табулации. Програмата трябва да намери в него всички числа, които се съдържат в речника. Тя трябва да ги преобразува до стойности от тип `unsigned short`, след което да намери сумата им и да я изведе на екрана.

За целите на задачата ще считаме, че число е максимална, непрекъсната поредица от цифри (вижте примера). Ако в низа се съдържа число, което е прекалено голямо, за да се съхрани в `unsigned short`, то трябва да се игнорира.

По-долу е даден пример за работата на програмата. Забележете, че се сумират само числата от речника, като те могат да са оградени с произволни символи от двете си страни. Резултатът е 32346, който се получава като $10000+10000+12345+1$

```
Enter dictionary size> 7
Enter dictionary contents> 10 3 500 12345 321 1 10000
Enter a string> 100000000--23000   ab10000cd __10000__12345      1
32346
```

Задача 2А

Напишете програма, която разпознава числа в символен низ. За да решите задачата, реализирайте следните неща:

А) Напишете функция

```
size_t DigitPrefix(const char* str)
```

която връща броя на цифрите, с които започва низа str. Например:

```
DigitPrefix("") → 0  
DigitPrefix("abc") → 0  
DigitPrefix("555") → 3  
DigitPrefix("123abc678") → 3
```

В) Напишете функция

```
unsigned int ToInt(const char* str, size_len)
```

която получава число представено като текст в низ str с дължина len. Функцията трябва да връща числото като стойност от тип unsigned int. Може да считате, че низът може да съдържа само и единствено цифри и че числото може да се събере в променлива от тип unsigned int. Например:

```
ToInt("", 0) → 0  
ToInt("123", 3) → 123
```

Функциите в точки А) и В) трябва да бъдат рекурсивни. Ако някоя от тях е реализирана нерекурсивно, тя се оценява с нула точки. Функциите трябва да се използват в подточка С), за да решите задачата. Затова, ако не се сещате за рекурсивно решение, разпишете ги така както можете, за да получите точките от С).

С) Напишете програма, която въвежда от потребителя символен низ с дължина до 80 символа. Низът може да съдържа произволни символи, включително интервали и табулации. Програмата трябва да преброи всички числа, които се съдържат в него и да ги съхрани в масив от тип unsigned int. Ако в низа се съдържа число, което е прекалено голямо, за да се съхрани в unsigned int, то трябва да се игнорира. Накрая програмата трябва да изведе на екрана, на три отделни реда, (1) броя на числата, (2) самите числа и (3) тяхната сума. За да решите задачата, използвайте функциите от подточки А) и В). По-долу е даден пример за работата на програмата.

```
Enter a string> 123 abc100def200cz      __1000__12345678901234567890__
```

4

123 100 200 1000

1423

Задача 2Б

Напишете програма, която разпознава нарастващи подредици от числа.

Редица от N-цели числа ще представяме като масив от тип `int`. Казваме, че редицата от числа A е подредица на B, ако всички елементи на A се срещат и в B и то точно в същия ред, както и в A. Например:

- Празната редица е подредица на всяка друга редица
- 1, 5, 6 е подредица на 1, 2, 3, 4, 5, 6
- 1, 2, 2, 2 е подредица на 1, 2, 3, 4, 2, 3, 2
- 1, 1, 2 НЕ Е подредица на 1, 2, 3

За да решите задачата реализирайте следните неща:

A) Напишете функция

```
bool IsSubsequence(const int* s1, size_t len1, const int* s2, size_t len2)
```

която проверява дали редицата `s1` с дължина `len1` е подредица на редицата `s2` с дължина `len2`.

B) Напишете функция

```
size_t IncreasingPrefixLength(const int* s, size_t len)
```

която връща дължината на най-дългия, строго нарастващ префикс на редицата `s` с дължина `len`. Например:

- Този префикс за празната редица е с дължина нула.
- За редицата 1, 2, 100, 11, 20, 50, този префикс е с дължина 3 (съответства на подредицата 1, 2, 100)
- За редицата 10, 9, 8, 7, 6, 5, този префикс е с дължина 1 (съответства на подредицата 10)

Функциите в точки A) и B) трябва да бъдат рекурсивни. Ако някоя от тях е реализирана нерекурсивно, тя се оценява с нула точки. Функциите трябва да се използват в подточка

C), за да решите задачата. Затова, ако не се сещате за рекурсивно решение, разпишете ги така както можете, за да получите точките от C).

C) Напишете програма, която въвежда от потребителя две редици от елементи S1 и S2. За целта програмата трябва най-напред да въведе цяло число N1, не по-голямо от 100. След това тя трябва да въведе N1 цели числа в масива S1. След това се въвежда N2 (отново по-малко от 100) и N2 на брой елемента в масива S2. Накрая програмата трябва да намери и да изведе на екрана най-дългата непрекъсната част от S1, която е строго нарастваща подредица на S2.

По-долу е даден пример за работата на програмата:

```
Enter N1> 8
Enter S1> 4 8 3 0 10 12 20 2
Enter N2> 15
Enter S2> 4 3 0 1 10 18 6 12 5 4 18 8 20 16 20
0 10 12 20
```