

НАЦИОНАЛЕН ПРОЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА ПЛОВДИВ, 27–28 МАЙ 2006

Задача A3. WAC – Word Auto Completion

Пешо Хакера завърши следването си и както често се случва в днешни дни единствената работа, която си намери бе в новоизлюпена фирма, пишеща софтуер за мобилни телефони. Телефоните, за които фирмата прави софтуер са доста слаби като процесор и памет, но имат изключително бърза връзка към централния сървър и затова на Пешо беше дадена задачата да напише програма, която да върви на сървъра и да обработва милионите заявки за автоматично завършване на думи, които се генерират всеки ден от различни мобилни телефони. Пешо не разбира много от алгоритми затова Вие ще трябва да напишете програмата **WAC** вместо него.

Програмата на Пешо трябва да изпълнява команди получени от библиотека и пак чрез нея да връща отговорите на командите. Командите са следните видове :

ADD <нова_дума>
FIRST <номер_на_мобилен_телефон> <представка>
NEXT <номер_на_мобилен_телефон>

При команда '**ADD**' думата <нова_дума> трябва да се прибави в речника на програмата.

На команда '**FIRST**' трябва да се отговори с първата добавена в речника дума, която има <представка> като свояставка.

На команда '**NEXT**' трябва да се отговори с поредната дума от речника(по реда на добавянето им), която има заставка <представка> от последната подадена от <номер_на_мобилен_телефон> команда '**FIRST**'(такава задължително ще има). В случай, че думите се изчерпят на тази команда трябва да бъде отговорено отново с първата добавена дума отговаряща на <представка>. След това с втората и т.н.

Работа с библиотеката

Преди викането на която и да е друга процедура на библиотеката, трябва да се извика процедурата **init()**. След това поредната команда се получава с помощта на процедурата **getQuery()** единствения параметър, на която е низ, в който библиотеката записва поредната команда в същия формат както е описано по-горе(отделните полета са разделени с точно един интервал). Този низ трябва да е достатъчно дълъг за да побере всяка заявка според условието на задачата. След вземане на команда от вид '**FIRST**' или '**NEXT**' на нея трябва да бъде отговорено с **answerQuery()** преди вземане на друга заявка. Низът, който се предава на **answerQuery()** е думата отговор на заявката или само една точка ако такава дума не съществува. След отговора на последната заявка библиотеката ще прекрати изпълнението на програмата Ви. При неспазване на горните правила получавате 0 точки за съответния тест. При опит за четене от стандартния вход или от файл или писане на стандартния изход или на файл получавате 0 точки за съответния тест.

ЗА ПИШЕЩИТЕ НА FreePascal

Библиотека: (module.ppu, module.o)

```
procedure init;  
procedure getQuery(var input_line : string);  
procedure answerQuery(const answer : string);
```

Програмата `example.pas` е пример за използване на тази библиотека.

Инструкции: За да компилирате вашия файл `wac.pas`, включете оператора `uses module;` в изходния текст и компилирайте с

fpc -So -O2 -XS wac.pas

ЗА ПИШЕЩИТЕ НА GNU C/C++

Библиотека (module.h, module.o)

```
void init();
```

```
void getQuery(char *input_line);
```

```
void answerQuery(char *answer);
```

Програмата example.c е пример за използване на тази библиотека.

Инструкции: За да компилирате вашия файл wac.c или wac.cpp, използвайте #include "module.h" в изходния текст и компилирайте с:

```
gcc -O2 -static wac.c module.o -lm -o wac.exe
```

```
gxx -O2 -static wac.cpp module.o -lm -o wac.exe
```

За пишешите на C/C++ в RHIDE: Задайте на Options->Linker configuration стойност module.o.

Вход

Модулът чете заявките от файл с име "**wac.in**". На първия ред е записан броя на командите, а на всеки следващ по една команда във формата описан по-горе.

Изход

Модулът записва отговорите на заявките в изходен файл с име "**wac.out**" в текущата директория.

Ограничения

Нека с **N** бележим броя команди **ADD**. С **M** сумата от дължините на думите добавени с **ADD**. С **W** дължината на най-дългата дума от речника или представка. С **Q** броя команди от тип **FIRST** и тип **NEXT** сумарно.

$$1 \leq N \leq 10^5$$

$$1 \leq M, Q \leq 10^6$$

$$1 \leq W \leq 30$$

$$1 \leq \text{length}(\text{<номер_на_мобилен_телефон>}) \leq 16$$

Броят различни номера на мобилни телефони за едно стартиране на програмата не надвишава 100000. Номерата на мобилни телефони се състоят от цифрите от 0 до 9. Всички думи и представки се състоят от малки латински букви.

Пример

wac.in

```
11
FIRST 002 ala
ADD alakazala
NEXT 002
ADD alamazala
ADD alabala
ADD ala
FIRST 003 alak
NEXT 002
NEXT 003
FIRST 001 bala
NEXT 002
```

wac.out

```
.
alakazala
alakazala
alamazala
alakazala
.
alabala
```