Programming Design In-class Practices

C Strings

Ling-Chieh Kung

Department of Information Management
National Taiwan University

Problem 1: Counting punctuation marks

- Given a sentence of English letters, punctuation marks, and spaces, count the number of punctuation marks.
  - The length of the sentence is at most 1000 characters.
  - The punctuation marks includes "`,.:;!?`".

Input:
```
Hi! Let us learn Programming together. What do you think?
```

Output:
```
3
```

Problem 2: Capitalizing initials

- Given a sentence of English letters, punctuation marks, and spaces, capitalize the initials of all words and make all other letters in lowercase.
  - The length of the sentence is at most 1000 characters.
  - The punctuation marks includes "`,.:;!?`".

Input:
```
HI! Let us learn programming together. I love NTU!
```

Output:
```
Hi! Let Us Learn Programming Together. I Love Ntu!
```

Problem 3: Case-insensitive search

- Let *A* and *B* be two sentences of English letters, punctuation marks, and spaces. Determine whether *B* is a case-insensitive substring in *A*.
  - The length of each sentence is at most 1000 characters.
  - The punctuation marks includes "`,.:;!?`".
  - Input: *A* in the first line, *B* in the second line.
  - Output: 1 if *B* is a case-insensitive substring of *A*, 0 otherwise.

Input:
```
When I have questions, I search online.
Search
```

Output:
```
1
```

Problem 4: Dictionary search

- Let x be an English word with English characters only. Let D be a collection of n English words with only lowercase letters. Words in D are sorted alphabetically. Determine whether x exists in D in a case-insensitive manner.
  - x has at most 50 characters. Each word in D has at most 50 characters.
  - n is no larger than the largest possible value of an **`int`** variable.
  - Input: n in the first line, x in the second line, D in the third to $(n + 2)$th line.
  - Output: 1 if x is in D and 0 otherwise.
- Linear search? Binary search?
- This is the basis of spell checking.

- Note: do not use cin.getline(), use cin >>. Ask the instructor about why.

Problem 4: Dictionary search

Input:
```
5
Watermelon
apple
banana
grapefruit
orange
watermelon
```

Output:
```
1
```

Input:
```
5
water
apple
banana
grapefruit
orange
watermelon
```

Output:
```
0
```

Problem 4: Dictionary search

- Note: do not use **`cin.getline()`**, use **`cin >>`**.
- The following program fails:

```
#include<iostream>
using namespace std;

int main()
{
  int i = 0;
  cin >> i;
  char s[100] = {0};
  cin.getline(s, 100); // no problem if cin >> s;
  return 0;
}
```

- Ask the instructor about why.

Problem 5: Replacing 0 as zero

- Let *A* be a sentences of English letters, punctuation marks, and spaces. Print out *A* by replacing all isolating “0” by “zero”. An “0” is isolating if it is not attached to any letters or numbers.
  - The length of *A* is at most 1000 characters.
  - The length of a word is at most 50 characters.
  - The number of words is at most 100.
  - The punctuation marks includes “`, . : ; ! ?`”.

Problem 5: Replacing 0 as zero

Input:
```
Among the 10 students, two got 0 point.
```

Output:
```
Among the 10 students, two got zero point.
```

Input:
```
Among the 10 students, two got 0.
```

Output:
```
Among the 10 students, two got zero.
```