Problem 2

Shuttle Displays

4 Points

Program Name: shuttle.cpp Input File: shuttle.dat

[This is a real story/situation.] The original displays on board the Space Shuttle (before the recent update) used green phosphorous semi-intelligent displays that displayed characters based on an embedded ROM (read-only memory). This allowed a very simple interface from the computers. It supplied the triplet (x,y,character) describing the position on the screen and the character that should appear there.

One of the maintenance checks on this display required a check of the ROM. A failure in the ROM could cause the distortion of a character (Q appears as an O for example) or the wrong character being displayed. The only way to check the proper operation of the ROM is to manually verify that it can display all characters. (We will limit it to the upper-case alphabetic characters only.) The human folks believe the most effective way to facilitate this is not to force the technician to check each character individually but rather to display a sentence that uses all characters. The technician then intuitively gravitates to spelling errors.

Your company is building these monitors and the test procedures. You need to write a program that, given a sentence of up to 80 characters, will determine if all upper case alphabetic characters are used at least once in the sentence. They can then submit sample sentences to your program to determine which are viable candidates for the test procedure.

For example, you can see that the sentence

THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS

Contains all of the upper case alphabetic characters at least once and is therefore a viable candidate. You can also see that

THE LAZY DOGS DID NOT SEE THE QUICK BROWN FOX

Does not contain a J, M, P, or V and is therefore not a viable candidate sentence.

Input

Input to your program is a series of sentences consisting of only upper case alphabetic characters and spaces up to 80 characters in length. Each sentence is contained on a line by itself in the input file. Your program should ignore the blank spaces.

Output

For each input sentence, your program should determine if the sentence is a viable candidate. If it is a viable candidate, your program should print exactly the message “Sentence is a candidate” on a line by itself. If it is not a viable candidate, your program should print exactly the message “Sentence is not a candidate” on a line by itself.

Example: Input File

```
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS
THE LAZY DOGS DID NOT SEE THE QUICK BROWN FOX
QUEEN JACKIE WAS VERY PERPLEXED WHEN THE COMPUTER FELLOW BEGAN COUNTING AT ZERO
OSCAR THE GROUCH ZIPPED QUICKLY THROUGH THE TRASH LOOKING FOR SOMETHING TO EAT
```

Output to screen

```
Sentence is a candidate
Sentence is not a candidate
Sentence is a candidate
Sentence is not a candidate
```