Problem 1 – Converting Dates

Write a program, named **`DateConvert`**, that reads a date in the format: **`mm/dd/yyyy`** and displays the date in the format: **`Month dd, yyyy`**

You may assume that the month number, **`mm`**, is an integer between 1 and 12, inclusive. The corresponding month name should be one of the following: `January, February, March, April, May, June, July, August, September, October, November, December`.

The date field, **`dd`**, may contain 1 or 2 digits. The year field ,**`yyyy`**, will always contain 4 digits.

The output should have 1 space between the name of the month and the date, and 1 space between the comma and the year. No other spaces should be displayed.

Sample Input	Corresponding Output
12/25/2007	December 25, 2007
10/9/2007	October 9, 2007
9/11/2001	September 11, 2001
1/1/1900	January 1, 1900

Problem 2 – Palindromes

Write a program, named **Palindromes**, that reads a string of characters and displays either YES or NO, depending on whether or not the string is a palindrome.

A string is a palindrome if it reads the same backwards as it does forward (ignoring all spaces and punctuation). For instance, the following strings are palindromes:

```
radar
a toyota
madam i'm adam
able was i ere i saw elba
a man a plan a canal panama
```

The input data will contain only lower case letters; however, it may also contain spaces and punctuation marks. Your program should display "YES" if the string is a palindrome; otherwise, it should display "NO".

Sample Input	Corresponding Output
radar	YES
a toyota	YES
madam i'm adam	YES
able was i ere i saw elba	YES
a man a plan a canal panama	YES
able saw i ere i saw elba	NO
toyota	NO

Problem 3 – igPay atinLay

Write a program, named **`PigLatin`**, that reads an English sentence from the keyboard, translates it into pig Latin, and displays the result. Pig Latin is a form of coded language often used for amusement. Many variations exist in the methods used to form pig Latin words. For purposes of this program, we'll use the following algorithm to translate each English word into a pig Latin word:

1) If the first letter of the English word is a consonant, place it at the end of the word and add the letters "ay". Thus, the word "jump" becomes "umpjay", the word "the" becomes "hetay", and the word "computer" becomes "omputercay".
2) If the first letter is one of the five vowels (a,e,i,o,u), do not move any letters, simply add the letters "ay" to the end of the word.
3) Blanks between words remain as blanks.

Assume that the English phrase consists of words separated by blanks, there are no punctuation marks, and all words have two or more letters.

Sample Input	Corresponding Output
`Sam has nine fingers`	`amSay ashay inenay ingersfay`
`Joe went to town`	`oeJay entway otay owntay`
`Did you get this right`	`idDay ouyay etgay histay ightray`
`How about this one`	`owHay aboutay histay oneay`

Problem 4 – Mountain Tops

Write a program, named **`MntTops`**, that reads elevation data and reports the height of each mountain top.

The program should first ask for and read two integers that represent a number of rows (`NumRows`) and a number of columns (`NumCols`). (`0<NumRows<10 & 0<NumCols<10`) The program should then read `NumRows` lines; and each line contains `NumCols` integers (separated by a single space). These integers represent the elevations of land areas as reported by a topographical map. Adjacent integers correspond to adjacent land areas.

For example, given the following 9 integers in the arrangement: $\begin{matrix} 78 & 77 & 70 \\ 79 & 80 & 79 \\ 78 & 77 & 78 \end{matrix}$, the land area corresponding to the middle number is at height 80; furthermore, it is adjacent to the other 8 areas.

A "mountain top" is defined to be a land area that is at an elevation that is higher than all of its adjacent neighbors (adjacent horizontally, vertically, and diagonally). For example, the land area in the example above that is at elevation = 80, corresponds to a mountain top.

Your program should find and report the elevations of all of the mountain tops.

Sample Input

```
7 6
70 71 72 73 74 75
76 77 78 77 76 75
70 71 72 73 74 75
70 70 70 70 70 70
75 70 74 70 73 70
70 70 70 70 70 70
69 69 72 69 69 69
```

Corresponding Output

```
Mountain Top Elevations:
78
75
74
73
72
```