```
Script started on 2023-12-07 23:39:01-06:00 [TERM="xterm" TTY="/dev/pts/0" COLUMNS=
ee43254@ares:~$ pwd
/home/students/ee43254
ee43254@ares:~$ cat piglatin.info
Name: Kyle Enkhzul

Class: CSC121-001



Activity: Atwhay isyay isthyay?

Level: 6, 6 (base program),



Description:



The user inputs an word or phrase they want converted into Pig Latin and the

program converts it for them.

ee43254@ares:~$ show-code piglatin.cpp


piglatin.cpp:
```

```cpp
#include <iostream>
#include <string>
#include <cctype>
#include <cmath>
#include <vector>

using namespace std;
constexpr streamsize INF_FLAG{numeric_limits<streamsize>::max()};

bool isVowel(char c);
bool hasNumber(const string & str);

string ConvertToPigLatin(const string & word);
string concatenateWords(const vector<string> & convertedWords);

vector<string> adjustText(const string & normalText, size_t width);
vector<string> convertTextToPigLatin(const vector<string> & normalWords);
vector<string> parsedWords(const string & paragraph);

int main()
{
    char y_n;

    vector<string> normalWords;
    vector<string> convertedWords;

        string normalText;
    string convertedText;

    cout << "\n\t Welcome to the Pig Latin Translation Program!!! \n" ;

    while( tolower(y_n) != 'n'){
        cout << "\n\t\tWhat text do you want converted to Pig Latin?\n\n";
        cin >> normalText;

                normalWords = parsedWords(normalText);
                convertedWords = convertTextToPigLatin(normalWords);
                convertedText = concatenateWords(convertedWords);

        cout << "\nThis is what the normal input looks like: "
             << concatenateWords(adjustText(normalText, 55));

        cout << "\nThis is what the translated input looks like: "
             << concatenateWords(adjustText(convertedText, 55));

        cout << "\nWould you like to translate more? \n\n";

        cin >> y_n;
        cin.ignore(INF_FLAG, '\n');
    }

     cout << "\nThank you for using the PLTP!!\n\n Have a good day! BYE !\n
}

/*
    This method is intended to take the individual words and put them back
        together to form a paragraph after the vector of strings has been
        translated into PigLatin.
    @return vector of strings of concatenated words
*/

string concatenateWords(const vector<string> & convertedWords)
{
    string paragraph;

    if (!convertedWords.empty()) {
        paragraph += convertedWords[0];
        for (size_t i = 1; i < convertedWords.size(); ++i) {
            paragraph += " " + convertedWords[i];
        }
    }

    return paragraph;
}

/*
        This method is intended to check whether or not a char is a vowel
    @return true/false depending on if input char is a vowel
```

```
*/
bool isVowel(char c)
{
    char convertChar = static_cast<char>(tolower(c));
    return (convertChar == 'a' || convertChar == 'e' || convertChar == 'i'
                                       convertChar == 'o' || convertChar =
}

/*
        This method is intended to check whether or not there is a digit l
        within a string by iterating through every element.
    @return true/false depending on if string contains a digit
*/
bool hasNumber(const string & str)
{
    for (char c : str)
    {
        if (isdigit(c))
        {
            return true;
        }
    }
    return false;
}

/*
    This method is intended to extract singular words out of a paragraph o
        line and store them into individual elements within a vector. This
        manipulation of each word in a paragraph.
    @return vector of strings of individual words
*/

vector<string> parsedWords(const string & paragraph)
{
    vector<string> normalWords;

    size_t initial = 0;
    size_t final = 0;

    while (final < paragraph.size()) {
        while (initial < paragraph.size() && paragraph[initial] == ' ') {
            initial++;
        }
        final = initial;
        while (final < paragraph.size() && paragraph[final] != ' ') {
            final++;
        }
        if (initial < final) {
            normalWords.push_back(paragraph.substr(initial, final - initia
        }
        initial = final + 1;
    }

    return normalWords;
}

/*
    This method is intended to convert a vector of strings to PigLatin and
        return an updated vector of strings in Pig Latin.
    @return vector of strings of translated Pig Latin words
*/
vector<string> convertTextToPigLatin(const vector<string> & normalWords)
{
    vector<string> pigLatinWords;
    for (const auto & word : normalWords)
    {
        pigLatinWords.push_back(ConvertToPigLatin(word));
    }
    return pigLatinWords;
}

/*
    This method is intended to convert a singular word to PigLatin.
    @return translated Pig Latin word
*/
string ConvertToPigLatin(const string & word)
{
    if (word.empty() || hasNumber(word))
    {
        return word;
    }

    size_t previousCharPos = word.size() - 1;
    char previousChar = word[previousCharPos];
    bool hasPunctuation = ispunct(previousChar);

    string words = word.substr(0, hasPunctuation ?
                                          previousCharPos : word.size());

    size_t firstVowelPos = 0;
    if (!isVowel(words[0]))
    {
        for (size_t i = 1; i < words.size(); ++i)
        {
            if (isVowel(static_cast<char>(words[i])) ||
               (words[i] == 'y' && i > 0 &&
               !isVowel(words[i - 1])))
            {
                firstVowelPos = i;
                break;
            }
        }
    }

    for (char & c : words)
    {
        c = static_cast<char>(tolower(c));
    }
```

```
187
188     string pigLatinWord;
189     if (firstVowelPos == 0)
190     {
191         pigLatinWord = words + "yay";
192     }
193     else
194     {
195         pigLatinWord = words.substr(firstVowelPos) +
196                        words.substr(0, firstVowelPos) + "ay";
197     }
198     if (isupper(word[0]))
199     {
200         pigLatinWord[0] = static_cast<char>(toupper(pigLatinWord[0]));
201     }
202     if (hasPunctuation)
203     {
204         pigLatinWord += previousChar;
205     }
206     return pigLatinWord;
207 }
208
209 /*
210         This method is intended to format the translated normalText for ni
211         @return formatted output of translated normalText
212 */
213
214 vector<string> adjustText(const string & normalText, size_t width)
215 {
216     vector<string> result;
217     string line;
218
219     for (char c : normalText)
220     {
221         if (c == '\n')
222         {
223             result.push_back(line);
224             line.clear();
225         }
226         else
227         {
228             line += c;
229
230             if (line.size() == width)
231             {
232                 result.push_back(line);
233                 line.clear();
234             }
235         }
236     }
237
238     if (!line.empty())
239     {
240         result.push_back(line);
241     }
242
243     return result;
244 }
```

```
ee43254@ares:~$ CPP piglatin
piglatin.cpp***


ee43254@ares:~$ ./piglatin.out

        Welcome to the Pig Latin Translation Program!!!

               What text do you want converted to Pig Latin?

is

This is what the normal input looks like: is
This is what the translated input looks like: isyay
Would you like to translate more?

y

               What text do you want converted to Pig Latin?

form

This is what the normal input looks like: form
This is what the translated input looks like: ormfay
Would you like to translate more?

y

               What text do you want converted to Pig Latin?

them

This is what the normal input looks like: them
This is what the translated input looks like: emthay
Would you like to translate more?

y

               What text do you want converted to Pig Latin?

through

This is what the normal input looks like: through
This is what the translated input looks like: oughthray
Would you like to translate more?

y

               What text do you want converted to Pig Latin?
```

```
qat

This is what the normal input looks like: qat
This is what the translated input looks like: atqay
Would you like to translate more?

y

                What text do you want converted to Pig Latin?

yellow

This is what the normal input looks like: yellow
This is what the translated input looks like: ellowyay
Would you like to translate more?

no

Thank you for using the PLTP!!

 Have a good day! BYE !

ee43254@ares:~$ cat piglatin.tpq
cat: piglatin.tpq: No such file or directory
ee43254@ares:~$ exit
exit

Script done on 2023-12-07 23:39:47-06:00 [COMMAND_EXIT_CODE="1"]
```