```
Script started on 2025-07-21 15:35:22-05:00 [TERM="xterm" TTY="/dev/pts/1" COLUMNS=
kn55307@ares:~/Portfolio 3/palindrome lab$ pwd

/home/students/kn55307/Portfolio 3/palindrome lab
kn55307@ares:~/Portfolio 3/palindrome lab$ cat palindrome.info

Noah Kang
CSC121-001
Palindrome Lab
Level 6.5:
    Level 2.5: Base Level
    Level 1.5: Allows users to check for both words and phrases
    Level 1.5: Merged two palindrome checking functions
        Level 1: Made the function argument default to checking words

This program allows users to input a word or phrase and check whether it
is a palindrome or not.

kn55307@ares:~/Portfolio 3/palindrome lab$ show-code palindrome.cpp



palindrome.cpp:


     1  #include <iostream>
     2  #include <vector>
     3  #include <string>
     4
     5  using namespace std;
     6
     7  void removePunctuation(std::string& phrase) {
     8      const std::string punct = "?:;.!'\",`~@#$%^&*()-_+=|\\}{][/><";
     9      std::string result;
    10      for (char c : phrase) {
    11          if (punct.find(c) == std::string::npos && c != ' ') {
    12              result += c;
    13          }
    14      }
    15      phrase = result;
    16  }
    17
    18
    19  bool isPalindrome(string& phrase, bool isPhrase = false) {
    20      int left = 0;
    21      int right = phrase.size() - 1;
    22
    23      if (isPhrase){
    24          removePunctuation(phrase);
    25      }
    26
    27      while (left < right) {
    28          if (tolower(phrase[left]) != tolower(phrase[right])) {
    29              return false;
    30          } else {
    31              left++;
    32              right--;
    33          }
    34      }
    35      return true;
    36  }
    37
    38  int main(){
    39      bool running = true;
    40      while (running){
    41          cout << "Would you like to check a word/phrase? (y/n): ";
    42          string running_response;
    43          cin >> running_response;
    44          if (running_response[0] == 'y'){
    45              cout << "Would you like to check a word or phrase? (w/p): ";
    46              string type_response;
    47              cin >> type_response;
    48              if (type_response[0] == 'w') {
    49                  cout << "Enter a word: ";
    50                  string word;
    51                  cin >> word;
    52                  if (isPalindrome(word)) {
    53                      cout << "The word is a palindrome." << endl;
    54                  } else {
    55                      cout << "The word is not a palindrome." << endl;
    56                  }
    57              } else {
    58                  cout << "Enter a phrase: ";
    59                  string phrase;
    60                  cin.ignore();
    61                  getline(cin, phrase);
    62                  removePunctuation(phrase);
    63                  if (isPalindrome(phrase, true)) {
    64                      cout << "The phrase is a palindrome." << endl;
    65                  } else {
    66                      cout << "The phrase is not a palindrome." << endl;
    67                  }
    68              }
    69          } else {
    70              cout << "Exiting!" << endl;
    71              running = false;
    72          }
    73      }
    74
    75
    76      return 0;
    77  }
kn55307@ares:~/Portfolio 3/palindrome lab$ CPP palindrome

palindrome.cpp***
.palindrome.cpp: In function ‘bool
isPalindrome(std::string&, bool)’:
palindrome.cpp:21:31:
```

```
warning: conversion from
'std::__cxx11::basic_string<char>::size_type' {aka
'long unsigned int'} to 'int'
may change value [-Wconversion]
   21 |     int right = phrase.size() - 1;
      |                 ~~~~~~~~~~~~~~^~~


kn55307@ares:~/Portfolio 3/palindrome lab$ ./palindrome.out

Would you like to check a word/phrase? (y/n): y
Would you like to check a word or phrase? (w/p): w
Enter a word: tacocat
The word is a palindrome.
Would you like to check a word/phrase? (y/n): y
Would you like to check a word or phrase? (w/p): w
Enter a word: tacocats
The word is not a palindrome.
Would you like to check a word/phrase? (y/n): y
Would you like to check a word or phrase? (w/p): p
Enter a phrase: I Love Me, Vol. I.
The phrase is a palindrome.
Would you like to check a word/phrase? (y/n): y
Would you like to check a word or phrase? (w/p): p
Enter a phrase: I do Love Me, Vol. I.
The phrase is not a palindrome.
Would you like to check a word/phrase? (y/n): n
Exiting!
kn55307@ares:~/Portfolio 3/palindrome lab$ cat palindrome.tpq

1. How can you access individual characters within a string?

    You can use the [] operators to access chars at certain indicies in
    a string.

2. How can you tell how long a string's content is?

    You can use the .size().

3. How can you tell that the characters at the beginning and end of a word
are the same?

    You can use two indexes that start from the beginning and end (size()
    -1) and compare the chars at those two values as they move closer to
    the middle (add to the starting index and substract from the ending
    index).

4. What do you do with odd length words?

    You can do the same procedure described in TPQ #3, but you can set
    a condition where the left index must be greater than the right index
    for the loop to continue, so when the two variables equal each other,
    being they meet at the middle char, the loop will exit.

5. How many characters do you have to look at to determine a word's
'palindromicness' if it contains 10 characters? 11 characters? n
characters?

    You would divide n by 2 using integer division and then multiply by 2.

6. Does your program consider 34743 a palindrome? Why/Why not?

    My program does since it treats 34743 as a string and each digit as a
    char.


TPQs for optional checking for phrases addition
1. Comparing the phrase algorithm to your original word palindrome
checker, is there really much difference? I'd be willing to bet the loop
head is identical and only the position update(s) need(s) to change...
(But perhaps I've said too much already...)

    There's no difference other than removing punctuation and spacing.

2. How can you detect that the user's input is a single word or a whole
phrase? (Hint: What does one have that the other doesn't? And how can you
look for that within a string?)

    My program asks the user to determine whether they're inputting a word
    or a phrase, but to automatically check, you can check for whitespace
    in the inputted string (phrases will have spaces in between the
    words).
kn55307@ares:~/Portfolio 3/palindrome lab$ exit

exit

Script done on 2025-07-21 15:36:21-05:00 [COMMAND_EXIT_CODE="0"]
```