CSCI204/MCS9204/CSCI804
Object and Generic Programming in C++
Laboratory Exercise 10 (Week 12)

Laboratory Exercise 10 (Week 12)

Task One: Palindrome (0.5 Marks)

A palindrome is a sequence of characters, numbers or words that have the property of reading the same in either direction (backwards or forwards). In a palindrome white space and punctuation have no significance (they are typically ignored).

Examples of palindromes are in the table below.

121 - A numeric sequence
GACTTCAG - A DNA Sequence
Sator Arepo tenet opera rotas. - A latin phrase for: *The farmer by his labour keeps the wheels to the plough.*

As you can see in the last example, white space and capitalization are ignored. ***You are to write a program that takes in a sequence and works out if it is/ isn't a palindrome***. We are going to do this in two different ways.

Write a C++ program that reads a string/sequence from standard input till EOF. The string/sequence, including all white space, punctuation and newline characters should be read in character-by-character and stored into a vector. The vector should hold objects of type char. When the sequence is finished being read in create two iterators. One iterator points to the beginning and the other the end of the vector.

By using these iterators you can work out if the sequence is the same in both forward and reverse directions. Remember that you should ignore case, punctuation and white space characters. If the sequence is a palindrome, you should print it out in both forward and reverse directions and indicate it is a palindrome. Otherwise display an error message.

Save your code as **`task1Main.cpp`**. You can compile the program by

```
$g++ -o palindrome task1Main.cpp
```

Run the program like

```
$./palindrome < exp1.txt
“GACTtCAaG” is a palindrome

$./palindrome < exp2.txt
“A Toyota! Race fast, safe car! A Toyota!” is a palindrome

$./palindrome < exp3.txt
“A nut for one jar of tuna.” is not a palindrome
```

Task 2: Maps (0.5 Marks)

The Australian Tax Office (ATO) is currently rewriting a piece of software. As part of the process the ATO needs to migrate a text file, which contains an integer as unique identifier (also known as a tax file number), and a string (a persons full name which can contain white space characters) per line. Each tax file number is 'normally' associated with one name.

The file may look something like this:

```
654342 Fred E Smith
213233 Joe Blow
212221 Sarah T Brown
897654 Jamie Shotton
874356 Richard G Lyn
```

Assume the appropriate headers are included; the file stream `ins` is opened for input and a map is created using the declaration,

```
map<int, string> taxrecord()
```

Assume the following `typedefs` exist.

```
typedef map<int, string>::iterator mapit;
typedef map<int, string> maptype;
```

(a) Write a function with the prototype

```
void populatemap(maptype& taxrecords, ifstream& ins);
```

Where the map `taxrecords`, is passed by reference and key/ data pairs are read from the stream `ins` one by one and inserted into the map. You should report any duplicate insertions into the map with an error message.

(b) Write a function with the prototype

```
mapit findrecord(maptype& taxrecords, int key);
```

Where the map `taxrecords` is passed by reference and an integer key, key, is passed. The function should attempt to find the key value in the map. If such a value is found, the function should return an iterator pointing to the matched pair, otherwise the value of the return value should be

```
taxrecords.end().
```

`atoMap.cpp` is a sketch of the program including a `main()` function to define and populate the `taxrecord` map and test the two function defined in (a) and (b).

Complete the code `atoMap.cpp`. Compile and test is as follows:

```
$g++ -o atoMap atoMap.cpp

$./atoMap < ato.txt

The total number of records is 5
Find the person whose TFN is 212221
212221 -> Sarah T Brown

Find the person whose TFN is 212221
Not find
```

Submission:

You should submit the files of tasks to the server by **11:59 PM on Friday, 27 May 2016** via command:

submit –u your-user-name –c CSCI204 –a L10 task1Main.cpp atoMap.cpp

and input your password.

Make sure that you use the correct file names. The UNIX system is case sensitive. You must submit all files in one *submit* command line.

After submit your assignment successfully, please check your email of confirmation. You should keep this email for the reference.

You would receive ZERO of the marks if your program codes could not be compiled correctly.

Later submission will not be accepted. Submission via e-mail is NOT acceptable.

End of Specification