**Name:**

**Advanced Programming in C++**

**Lab Exercise 1/27/2023**

In this exercise, you will explore some of the issues with regard to how numbers are stored in a computer. When you have completed submit your source code with sample output.

1. Roundoff errors. Consider the common fractions 1/2 , 1/3, 1/4 , 1/5, 1/6, and 1/7. In our decimal number system, we can represent the 1/2, 1/4, and 1/5 exactly as 0.5, 0.25, and 0.2 exactly, but the others such as 1/3 are represented as 0.3333333 repeating indefinitely. In our computers, numbers are represented in binary format which causes what we call as roundoff errors. You task is to write a program that output the ratio of these fractions (one per line). You are to then sum each of these fractions 840 times (why 840 times?). **Be sure you display your result to 20 decimal places if you are using double data types**. Your output should look like:

Output:

1/2 = 0.50000000000000000000

1/3 = 0.33333333333333331000

1/4 = 0.25000000000000000000

1/5 = 0.20000000000000001000

1/6 = 0.16666666666666666000

1/7 = 0.14285714285714285000

1/2 summed 840 times = 420.00000000000000000000

1/3 summed 840 times = 280.00000000000153000000

1/4 summed 840 times = 210.00000000000000000000

1/5 summed 840 times = 167.99999999999901000000

1/6 summed 840 times = 140.00000000000077000000

1/7 summed 840 times = 119.99999999999901000000

From this output, can you determine the precision of the machine?

1. Write a program that takes as an input the number of kilometers and prints the corresponding number of nautical miles. Use the following approximations:
   1. A kilometer represents 1/10000 of the distance between the geographic North Pole and the Equator
   2. There are 90 degrees each containing 60 minutes of arc between the geographic North Pole and the Equator.
   3. A nautical mile is one minute of arc.

**The Storywriter**

Write a program that generates a story based on several user inputs. The program should ask the user to enter the following items:

* Their name
* Their age
* The name of a city
* The name of a college
* A profession
* A type of animal
* A pet’s name

After the user enters these items, the program should display the following story, inserting the users input into the appropriate locations.

There once was a person named NAME who lived in CITY. At the age of AGE, NAME went to college at COLLEGE. NAME graduated and went to work as adopted a(n) *(see note)* PROFESSION. Then NAME adopted a(n) *(see note)* ANIMAL named PETNAME. They both lived happily ever after.

Note: if the type of animal or profession starts with a vowel (a, e, i, o, u) you need to have “an” instead of “a”.

Here is an example how you can test for a vowel.

**//Demonstration of using the find String class function**

**//find method will return the index where a character is found**

**//Author: nmessa**

#include <iostream>

#include <string>

using namespace std;

int main()

{

//Create a test string

string vowels = "aeiou";

//Create two test characters

char ch = 'u';

char ch1 = 'z';

int answer;

//Test the letter 'u' - should print Vowel

answer = vowels.find(ch);

if (answer != -1)

cout << "Vowel" << endl;

else

cout << "Not a vowel" << endl;

//Test the letter 'z' - should print Not a vowel

answer = vowels.find(ch1);

if (answer != -1)

cout << "Vowel" << endl;

else

cout << "Not a vowel" << endl;

return 0;

}

**Your Turn**

Repeat the above project, but this time, you are to create your own story. You must have a minimum of 7 user inputs.