CS319 SAMPLE Class Test ANS with solutions

Q1 Download, compile and run `Obscure.cpp` from
`www.niallmadden.ie/2324-CS319/ClassTest/Obscure.cpp`

What output does it generate? Write another C++ program which, when compiled, generates the same output, and has at most 5 lines of code.

Answer: The program outputs `Hello world!`. That could also be done with this program:

```
// Solution to Q1
#include <iostream>
int main(void){
  std::cout << "Hello world!\n";
  return(0); }
```

.....

Q2 For this question, it helps to know that

- if we set `x=rand()%n;` then `x` will be a random number between 0 and $n-1$.
- the line:
  `int a[5];`
  creates an array (list) of 5 integers called $a[0]$, $a[1]$, $a[2]$, $a[3]$ and $a[4]$.

Download the program at
`www.niallmadden.ie/2324-CS319/ClassTest/IsSortedStub.cpp`
This is a "stub" of a programme that generates an `int` array, `a`, of length 5, such that `a[i]` is a random integer between 0 and $100i$. (That is, `a[0]` is 0, `a[1]` is between 0 and 100, `a[2]` is between 0 and 200, etc.

The program also has the function with header
`IsSorted(int x[], int n);`
which *should* return "`true`" if the entries in `x` are sorted in non-decreasing order. When I run it I get the (clearly incorrect) output

```
a[0]=0
a[1]=32
a[2]=147
a[3]=47
a[4]=29
List a is sorted? true
```

(a) Modify the code so that the entries of `a` are generated using a `for`-loop, and the output of the values of `a` is produced by a `for` loop.

(b) Write code for the `IsSorted()` function so that it correctly determines if the sequence passed to it is non-decreasing.

(c) Change the code so that if `IsSorted()` is called with just its first argument (i.e., `n` is omitted), it is assumed that the array is of length 2.

The goal of Q2 is to verify that you are competent writing `for`-loops and `if`-statements, and working with functions.

Answer:

```
// Solution to Q2 from 2324-Cs319 Sample Class Test
#include <iostream>
#include <cstdlib>

bool IsSorted(int x[], int n);

int main(void)
{
  int a[5];
  for (int i=0; i<5; i++)
  {
    a[i] = rand()%(i*100+1);
    std::cout << "a["<< i << "]="
              << a[i] << std::endl;
  }

  std::cout << "List a is sorted? " <<
    std::boolalpha << IsSorted(a, 5) <<
    std::endl;
  return(0);
}

bool IsSorted(int x[], int n)
{
  for (int i=0; i<n-1;i++)
    if (x[i+1]<x[i])
      return(false);
  return(true);
}
```

.....

Q3 If a temperature is measured in Celsius as C and in Fahrenheit as F, then

$$F = \frac{9}{5}C + 32.$$

Write a short C++ program that output a table showing that values of Fahrenheit that correspond to values of Celsius, starting at -40 and going in steps of 8 to +40. That is, the output should look something like

```
C = -40 corresponds to F=-40
C = -32 corresponds to F=-25.6
...
```

```
C = 24 corresponds to F=75.2
C = 32 corresponds to F=89.6
```

The purpose of Q3 is to verify that you can write loops and display output.

Answer:

```
// Solution to Q3 from 2324-CS319 Sample Class Test
#include <iostream>

int main(void)
{
  double C, F;

  for (C=-40; C<=40; C+=8.0)
  {
    F = (9.0/5.0)*C + 32.0;
    std::cout << "C = " << C <<
      " corresponds to F=" <<
      F << std::endl;
  }
  return(0);
}
```

.....

Q4 Write a function with header

```
bool IsLeapYear(int year);
```

that returns `true` if the integer stored in `year` corresponds to a leap year. Note that, to be a leap year, the year

- The year must be at least 1582 (the year the Gregorian calendar came into effect).
- The year is divisible by 400, or but devisible by 4 but not 100.

Your code should include a snippet of code in the `main()` function that shows your function working for several years.

The purpose of Q4 is to verify you skills in writing `if` statements and functions.

Answer:

```
// Solution to Q4 from 2324-CS319 Sample Class Test
#include <iostream>
#include <cstdlib>

bool IsLeapYear(int year);

int main(void)
{
  int years[]={1572, 1900, 2000, 2023, 2024};
  std::cout << "--Testing IsLeapYear()--\n";
  for (int i=0; i<=4; i++)
    if (IsLeapYear(years[i]))
      std::cout << years[i] <<
        " is a leap year\n";
    else
      std::cout << years[i] <<
        " is not a leap year\n";
  return(0);
}

// Note: could use nested is statements to make this
// code clearer.
bool IsLeapYear(int year)
{
  if (year > 1582 &&
      ( (year%400)==0 ||
        ((year%4)==0) &&
        ((year%100)!=0) ) )
    return(true);
  else
    return(false);
}
```