This lab investigates the use of the `for()` loop and the `switch()` decision structure.

1. Make a copy of the sample program `euler.cpp` and compile it changing the name of the executable to the basename `euler`
   
   ```
   g++ -g euler.cpp -o euler < cr >
   ```
   
   Run it (by typing)
   ```
   euler < cr >
   ```
   
   several times. Two numbers are said to be *relatively prime* if their greatest common divisor is one. This is equivalent to saying that they share no common factors (other than one). The program accepts a positive integer $n$ greater than one and it computes the Euler phi-function $\phi(n)$ of the number. This is the number of positive integers less than the number and relatively prime to it. For example, let $n = 8$. The positive integers less than $n$ are the set $\{1, 2, 3, 4, 5, 6, 7\}$. Of these numbers, only $\{1, 3, 5, 7\}$ are relatively prime to 8. Therefore, $\phi(8) = 4$. The program has a diagnostic statement in the `for()` loop which you will want to comment out once you are sure that you understand how the algorithm works.

2. Give the program a prime number $p$. What should $\phi(p)$ be and why? If $p$ and $q$ are *different* primes what should $\phi(pq)$ be and why? Test this by doing some examples with small prime numbers like $2, 3, 5, 7, 11, 13, 17, 19$.

3. Make a copy of the program `pzoft.cpp` and compile and execute it. It is basically a spoof of the campus PeopleSoft database for faculty, students, and staff. Look at the code and see how a `do ... while()` together with a `switch()` statement can be used to implement a menu.

**Assignment** Write a program which performs the task described below and email me the pathname of your program *in plain text*; **do not send an attachment**. For example, you might say

```
My lab3 program is /usr/stu/demo/cs221/homewk3.cpp
```

There are many ways to estimate $\pi$. Leibniz noted that

$$
\pi = 4 \left( \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{2k-1} \right)
$$

Of course, you cannot do an infinite sum but if you sum up to $k = 100000$ you will get a reasonably good estimate. Machin noted that

$$
\pi = 16 \tan^{-1}(1/5) - 4 \tan^{-1}(1/239)
$$

Again, the inverse tangent function `atan()` in `math.h` will not be exactly correct but you will get a good approximation. Write a menu program with three menu entries which will let the user choose between approximating $\pi$ to **10 decimal places** using either Leibniz’ method or Machin’s method, or exiting the program.