Practice tasks: Loops

Dry run the following programs and tell the output without running the code in editor (MS Visual c++).

1. ```
   for (int count = 1; count < 5; count++)
           cout << (2 * count) << " ";
   ```
2. ```
   for (int n = 10; n > 0; n = n - 2)
   {
           cout << "Hello ";
           cout << n << endl;
   }
   ```
3. ```
   for (double sample = 2; sample > 0; sample = sample - 0.5)
           cout << sample << " ";
   ```
4. ```
   int n = 1024;

   int log = 0;
   for (int i = 1; i < n; i = i * 2)
   log++;
   cout << n << " " << log << endl;
   ```
5. ```
   int n = 1024;
   int log = 0;
   for (int i = 1; i < n; i = i * 2);
   log++;
   cout << n << " " << log << endl;
   ```
6. ```
   int n = 1024;
   int log = 0;
   for (int i = 0; i < n; i = i * 2)
   log++;
   cout << n << " " << log << endl;
   ```
7. ```
   int n = 5;

   while (—n > 0)
   {
   if (n == 2)
   break;
   cout << n << " ";
   }
   ```
   ```

```
cout << "End of Loop.";
```

8.
```
int n = 5;
while (—n > 0)
{
if (n == 2)
exit(0);
cout << n << " ";
}
cout << "End of Loop.";
```

9.
```
int n, m;
for (n = 1; n <= 10; n++)
for (m = 10; m >= 1; m–)
cout << n << " times " << m << " = " << n * m << endl;
```

Write a C++ program for the following problems

1. Calculate the value of π from the infinite series using loop.

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \cdots$$

   Print a table that shows the approximate value of π after each of the first 1000 terms of this series.

2. Write a program to score the paper-rock-scissor game. Each of two users types in either P, R, or S. The program then announces the winner as well as the basis for determining the winner: Paper covers rock, Rock breaks scissors, Scissors cut paper, or Nobody wins. Be sure to allow the users to use lowercase as well as uppercase letters. Your program should include a loop that lets the user play again until the user says she or he is done.

3. Write a program to compute the interest due, total amount due, and the minimum payment for a revolving credit account. The program accepts the account balance as input, then adds on the interest to get the total amount due. The rate schedules are the following: The interest is 1.5 percent on the first $1,000 and 1 percent on any amount over that. The minimum payment is the total amount due if that is $10 or less; otherwise, it is $10 or 10 percent of the total amount owed, whichever is larger. Your program should include a loop that lets the user repeat this calculation until the user says she or he is done.

4. Write an astrology program. The user types in a birthday, and the program responds with the sign and horoscope for that birthday. The month may be entered as a number from 1 to 12. Then enhance your program so that if the birthday is only one or two days away from an

adjacent sign, the program announces that the birthday is on a "cusp" and also outputs the horoscope for that nearest adjacent sign. This program will have a long multiway branch. Make up a horoscope for each sign. Your program should include a loop that lets the user repeat this calculation until the user says she or he is done.

The horoscope signs and dates are:

Aries	March 21–April 19
Taurus	April 20–May 20
Gemini	May 21–June 21
Cancer	June 22–July 22
Leo	July 23–August 22
Virgo	August 23–September 22
Libra	September 23–October 22
Scorpio	October 23–November 21
Sagittarius	November 22–December 21
Capricorn	December 22–January 19
Aquarius	January 20–February 18
Pisces	February 19–March 20

5. Write a program that finds and prints all of the prime numbers between 3 and 100. A prime number is a number such that 1 and itself are the only numbers that evenly divide it (for example, 3, 5, 7, 11, 13, 17, …). One way to solve this problem is to use a doubly nested loop. The outer loop can iterate from 3 to 100 while the inner loop checks to see if the counter value for the outer loop is prime. One way to see if number n is prime is to loop from 2 to n 21 and if any of these numbers evenly divides n, then n cannot be prime. If none of the values from 2 to n 21 evenly divides n, then n must be prime. (Note that there are several easy ways to make this algorithm more efficient.)

6. Buoyancy is the ability of an object to float. Archimedes' principle states that the buoyant force is equal to the weight of the fluid that is displaced by the submerged object. The buoyant force can be computed by

   Fb = V × y

   where Fb is the buoyant force, V is the volume of the submerged object, and y is the specific weight of the fluid. If Fb is greater than or equal to the weight of the object, then it will float, otherwise it will sink.

7. Write a program that inputs the weight (in pounds) and radius (in feet) of a sphere and outputs whether the sphere will sink or float in water. Use y = 62.4 lb/ft3 as the specific weight of water. The volume of a sphere is computed by (4/3)πr3.

8. Write a program that finds the temperature that is the same in both Celsius and Fahrenheit. The formula to convert from Celsius to Fahrenheit is

$$\text{Fahrenheit} = \frac{(9 \times \text{Celsius})}{5} + 32$$

   Your program should create two integer variables for the temperature in Celsius and Fahrenheit. Initialize the temperature to 100 degrees Celsius. In a loop, decrement the

Celsius value and compute the corresponding temperature in Fahrenheit until the two values are the same. Since you are working with integer values, the formula may not give an exact result for every possible Celsius temperature. This will not affect your solution to this particular problem.

9. Write a program that scores a blackjack hand. In blackjack, a player receives from two to five cards. The cards 2 through 10 are scored as 2 through 10 points each. The face cards—jack, queen, and king—are scored as 10 points. The goal is to come as close to a score of 21 as possible without going over 21. Hence, any score over 21 is called "busted." The ace can count as either 1 or 11, whichever is better for the user. For example, an ace and a 10 can be scored as either 11 or 21. Since 21 is a better score, this hand is scored as 21. An ace and two 8s can be scored as either 17 or 27. Since 27 is a "busted" score, this hand is scored as 17.

   The user is asked how many cards she or he has, and the user responds with one of the integers 2, 3, 4, or 5. The user is then asked for the card values. Card values are 2 through 10, jack, queen, king, and ace. A good way to handle input is to use the type char so that the card input 2, for example, is read as the character '2', rather than as the number 2. Input the values 2 through 9 as the characters '2' through '9'. Input the values 10, jack, queen, king, and ace as the characters 't', 'j', 'q', 'k', and 'a'. (Of course, the user does not type in the single quotes.) Be sure to allow upper- as well as lowercase letters as input.

   After reading in the values, the program should convert them from character values to numeric card scores, taking special care for aces. The output is either a number between 2 and 21 (inclusive) or the word Busted. You are likely to have one or more long multiway branches that use a switch statement or nested if-else statement. Your program should include a loop that lets the user repeat this calculation until the user says she or he is done.

10. Interest on a loan is paid on a declining balance, and hence a loan with an interest rate of, say, 14 percent can cost significantly less than 14 percent of the balance. Write a program that takes a loan amount and interest rate as input and then outputs the monthly payments and balance of the loan until the loan is paid off. Assume that the monthly payments are one twentieth of the original loan amount, and that any amount in excess of the interest is credited toward decreasing the balance due. Thus, on a loan of $20,000, the payments would be $1,000 a month. If the interest rate is 10 percent, then each month the interest is one-twelfth of 10 percent of the remaining balance. The first month, (10 percent of $20,000)/12, or $166.67, would be paid in interest, and the remaining $833.33 would decrease the balance to $19,166.67. The following month the interest would be (10 percent of $19,166.67)/12, and so forth. Also have the program output the total interest paid over the life of the loan. Finally, determine what simple annualized percentage of the original loan balance was paid in interest. For example, if $1,000 was paid in interest on a $10,000 loan and it took 2 years to pay off, then the annualized interest is $500, which is 5 percent of the $10,000 loan amount. Your program should allow the user to repeat this calculation as often as desired.

11. The Fibonacci numbers F_n are defined as follows. F_0 is 1, F_1 is 1, and

$$F_{i+2} = F_i + F_{i+1}$$

i = 0, 1, 2, … . In other words, each number is the sum of the previous two numbers. The first few Fibonacci numbers are 1, 1, 2, 3, 5, and 8. One place that these numbers occur is as certain population growth rates. If a population has no deaths, then the series shows the size of the population after each time period. It takes an organism two time periods to mature to reproducing age, and then the organism reproduces once every time period. The formula applies most straightforwardly to asexual reproduction at a rate of one offspring per time period.

Assume that the green crud population grows at this rate and has a time period of 5 days. Hence, if a green crud population starts out as 10 pounds of crud, then in 5 days there is still 10 pounds of crud; in 10 days there is 20 pounds of crud, in 15 days 30 pounds, in 20 days 50 pounds, and so forth. Write a program that takes both the initial size of a green crud population (in pounds) and a number of days as input, and that outputs the number of pounds of green crud after that many days. Assume that the population size is the same for 4 days and then increases every fifth day. Your program should allow the user to repeat this calculation as often as desired.

12. The value e^x can be approximated by the sum

$$1 + x + x^2/2! + x^3/3! + \ldots + x^n/n!$$

Write a program that takes a value x as input and outputs this sum for n taken to be each of the values 1 to 100. The program should also output ex calculated using the predefined function exp. The function exp is a predefined function such that exp(x) returns an approximation to the value ex. The function exp is in the library with the header file cmath. Your program should repeat the calculation for new values of x until the user says she or he is through. U se variables of type double to store the factorials or you are likely to produce integer overflow (or arrange your calculation to avoid any direct calculation of factorials). 100 lines of output might not fit comfortably on your screen. Output the 100 output values in a format that will fit all 100 values on the screen. For example, you might output 10 lines with 10 values on each line.

13. The keypad on your oven is used to enter the desired baking temperature and is arranged like the digits on a phone:

1	2	3
4	5	6
7	8	9
	0	

Unfortunately the circuitry is damaged and the digits in the leftmost column no longer function. In other words, the digits 1, 4, and 7 do not work. If a recipe calls for a temperature

that can’t be entered, then you would like to substitute a temperature that can be entered. Write a program that inputs a desired temperature. The temperature must be between 0 and 999 degrees. If the desired temperature does not contain 1, 4, or 7, then output the desired temperature. Otherwise, compute the next largest and the next smallest temperature that does not contain 1, 4, or 7 and output both. For example, if the desired temperature is 450, then the program should output 399 and 500. Similarly, if the desired temperature is 375, then the program should output 380 and 369.