



Computer Programming Language

【Fall, 2018】

Homework 3

Program A : Calculation of π (25%)

Write a program to calculate the value of the ratio of a circle's circumference to its diameter, π , using Leibniz series.

$$\frac{\pi}{4} = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$$

Calculate the π values for $k = 1000, 10000, 100000$, and estimate the errors by comparing a constant variable PI equals 3.141592653589793238463 in your program. Show the π values and errors for different k values on the screen. What is the most accurate value of π you can obtain?

■ Web-Cat Submission Check:

```
double answer1; // Store the value of  $\pi$  (  $k = 100000$ ) you calculated in this global variable
                // before exiting your program
```

Program B : Calculation of prime numbers (25%)

A prime integer number is one that has exactly two different divisors, namely 1 and the number itself. Write, run, and test a C++ program that finds and displays all the prime numbers less than 1000. (*Hint: For each number from 2 to 1000, find Remainder = Number % n, where n ranges from 2 to sqrt(Number). If n is greater than sqrt(Number), the number is not equally divisible by n. Why? If any Remainder equals 0, the number is not a prime number.*)

■ Web-Cat Submission Check:

```
int answer1; // Store the largest prime number less than 1000 in this global variable
```

Bonus points (25%): (This is an optional problem)

A pair of positive integer numbers are called twin primes if they are both prime numbers and the difference between them is 2, i.e., they are consecutive odd numbers and they are prime numbers. (3, 5), (5, 7) and (11, 13) are three examples of such pair of twin prime numbers. Write a program to display all the pairs of twin prime numbers that are less than 1000. What is the greatest twin primes you can obtain?

■ Web-Cat Submission Check:

```
int answer1; // First prime number of the largest pair of twin prime number less than 1000
int answer2; // Second prime number of the largest pair of twin prime number less than 1000
```



Program C : Use of selection structure and repetition structure (25%)

A happy number is defined by the following process. Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers. In other words, numbers that are happy follow a sequence that ends in 1. All non-happy numbers follow sequences that reach the cycle. Design a program to find happy numbers up to 1000.

For example, 19 is a happy number, as the associated sequence is:

$$1^2 + 9^2 = 82$$

$$8^2 + 2^2 = 68$$

$$6^2 + 8^2 = 100$$

$$1^2 + 0^2 + 0^2 = 1.$$

■ *Web-Cat Submission Check:*

```
int answer1;    // The largest happy number less than 1000
```

Program D : Rock-Paper-Scissors game (25%)

There is an ancient game played between two participants who simultaneously cast one of three figures by their fingers - Rock, Paper or Scissors. If both cast the same figure, the round is considered a draw. Otherwise the following rules are applied:

Rock beats Scissors (by blunting them)

Scissors beat Paper (by cutting it)

Paper beats Rock (by covering it)

Often the game is played on the staircase. Player who wins the round advances one step. One who reaches the end of the staircase first is the winner of the game.

Please design a computer program for a person to play Rock-Paper-Scissors game against computer. Assume the number of steps of the staircase is 3 and let the computer be your opponent. Play the game as you cast your figure at each round and find who is the winner of this game. Display the result of each round, and finally show who wins the game and report the total number of rounds in the game. Design the user input for Rock, Scissors, and Paper to be 'R', 'S' and 'P', respectively.

■ *Web-Cat Submission Check:*

```
int answer1;    // Store the total number of rounds in this variable
```



Notes:

1. Please submit your programs (source codes) to the Web-CAT grading system website (http://140.112.94.129:8080/Web-CAT_1.4.0/WebObjects/Web-CAT.woa/) before **Oct. 25.** (3:30PM)
2. Late submission will have a penalty of 10% discount per day of your grade toward a minimum score of 60. No late submission over a week will be accepted.
3. Criteria of grading include: (1) Program functionality; (2). User interface; (3). Structure of the program; (4). Suitable comments; (5). Programming style; (6). Creativity.