HCMC UNIVERSITY OF TECHNOLOGY, VNU-HCM
Faculty of Computer Science and Engineering

Exercises for week 10

(Chapter 5)

In this week, we practice implementing in C++ small code with repetition structures. Students can see some sample exercises and must prepare solution for all exercises in part B.

A. Sample exercises.

Exercise 1. Write and run a program to compute the sum of square of the integers from 1 to *N*, where *N* is an input parameter.

SOLUTION:

```
#include <iostream>
using namespace std;

int main(){
   int n;
   cout << "Give the value of n: ";
   cin >> n;
   int result = 0;

   for (int i=1;i<n+1;i++)
   {
      result+=i*i;
   }

   return result;
}
```

Exercise 2. Write and run a program that reads a positive integer value for *K* and then computes K! = 1*2*3*…*(K-1)*K and displays the result out.

SOLUTION:

```
#include <iostream>
using namespace std;

int main() {
   int k;
   cout << "Input value of k: ";
   cin >> k;
```

```
    int result = 1;

    for (int i=2; i<k+1; i++)
       result*= i;

    return result;
}
```

B. Exercises must to do.

Exercise 3. Write and run a program that inputs an array of *N* real numbers, and then finds the largest element in the array. *N* should be an input parameter.

Exercise 4. Write and run a program that inputs an array of *N* real numbers, and then computes the average value of the array elements. *N* should be an input parameter.

Exercise 5. Write and run a program that computes *x* raised to the power *n* by repetitive multiplication. Then modify your program to calculate *x* raised to the power (-*n*).

Exercise 6. Write and run a program to read a list of real numbers and then find the number of positive values and the number of negative ones among them. The number of entries is also entered by the user.

Exercise 7. Write and run a program that inputs an integer matrix of order n and transposes it and then prints it out. Transposing a square matrix means: $\mathbf{a_{ij} \leftrightarrow a_{ji}}$ **for all i, j.**

Exercise 8. Write and run a program to compute the value of *pi*, using the series for approximating: pi /4 = 1-1/3+1/5-1/7+ ... + $(-1)^n/(2*n+1)$

Hint: Use a *while* loop that terminates when the difference between two successive approximations is less than 1.0E-6.

C. Exercises in advanced.

Exercise 9. Write and run a program to tabulate sin(x), cos(x) and tan(x) for x = 5, 10, 15,…,85 degrees. Notice that we have to convert *x* from degrees to radians before using standard functions sin(x), cos(x), tan(x).

Exercise 10. The value of Euler's number, *e*, can be approximated using the formula:

e = 1 + 1/1! + ½! + 1/3! + ¼! + … +1/n!

Using this formula, write a program that approximates the value of *e* using a *while* loop that terminates when the difference between two successive approximations is less than 1.0E-6.

Exercise 11. The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13,…, where the first two terms are 0 and 1, and each term thereafter is the sum of the two preceding terms, that is, $Fib_n = Fib_{n-1} + Fib_{n-2}$. Using this information, write a program that calculates the *n*th number in a Fibonacci sequence, where *n* is entered into the program by the user.

Exercise 12. The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13,…, where the first two terms are 0 and 1, and each term thereafter is the sum of the two preceding terms. Write a program that computes and stores the Fibonacci sequence in an integer array *F*, such that F[i] will store the *i*th number in a Fibonacci sequence. The size of the array is an input parameter which is entered by the user.

Exercise 13. Write a program that stores the following hourly rates in an array name *hourly_rates*: 9.5, 6.4, 12.5, 5.5, 10.5. Your program should also create two arrays named *working_hours* and *wages*, each capaple of storing five double-precision numbers. Using a *for* loop and a *cin* statement, have your program accept five user-input numbers into *working_hours* array when the program is run. Your program should store the product of the corresponding values in the *hourly_rates* and *working_hours* arrays in the wages array (for example, wages[1] = hourly_rate[1]*working_hours[1]) and display the output as a table consisting of three columns.

Exercise 14. Write and run a program that reads three strings and prints them out in an alphabetical order. (Hint: Use the *strcmp()* function).

Exercise 15. The following program reads a set of name, roll number, sex, height and weight of the students from the keyboard using a structure within an array.

```
#include<iostream.h>
#include<string.h>
const int MAX = 100
struct student{
        char name[20];
        long int rollno;
        char sex;
        float height;
        float weight;
};
```

```
void main(){
   student cls[MAX];
   int i,n;
   cout << " How many names ? \n";
   cin >> n;
   for( i = 0; i <= n-1; ++i){
         cout << "record = "<< i+1 << endl;
         cout << "name : "; cin>> cls[i].name;
         cout << "rollno : "; cin>> cls[i].rollno;
         cout << "sex : "; cin>> cls[i].sex;
         cout << "height : "; cin>> cls[i].height;
         cout << "weight : "; cin>> cls[i].weight;
         cout >> endl;
   }
……
}
```

Include into the above program the code that performs two tasks:

a. displaying data of *n* students in the following format:

Name	Rollno	Sex	Height	Weight
------	--------	----	-------	--------
------	--------	----	-------	--------

b. computing and displaying the average of heights and the average of weights of the students.

-- END --