# RBE104TC

# Semester1 Course Work 1

**author:** Xi Nie

**Student ID:** 2255693

**Date:** Sunday 17th September, 2023

# 1 Software Development Process (SDP)

## 1.1 Problem Statement:

1. Provide an interface through which the user can input name(string), phone number(long), age(int), temperature(int),decimalNumber(int),source0(string), target(string); If an illegal input is found, the program will be terminated. The concept of illegality can be defined as follows:

·Read the information from the keyboard and put it in the appropriate variables

·Calculate the quotient of the first six digits of the phone number by the last five, assign it to a variable and display this value.

·Rewrite decimal numbers to octal and hexadecimal and print.

·Convert the temperature from Celsius to Fahrenheit and Kelvin. Round data to the nearest whole number and display them on the screen.

·Finding the real solutions of a quadratic equation for any real values of a, b, and c is the goal.

·Searches the range for the desired string and returns the first result as self.

2. The program should display five series of outputs.
3. The program exists.

## 1.2 Analysis for exercise 1:

• On an input:

Initially, the computer transmits a communication indicating the desired data format to the customer, then monitors the keyboard input to verify its conformity with the specified type.

• On an output:

Upon receiving valid input from the user, the system is capable of performing various calculations, including determining the quotient, generating octal and hexadecimal representations of a decimal number, obtaining alternative temperature representations, solving quadratic equations, and displaying these results along with relevant explanatory details at the top of the screen.

• Data structure:

Only single data is involved. So, there are no complex data structure issues here.

• Algorithm:

The preceding equations are easy to evaluate. There are no algorithm concerns either.

## 1.3 Design for exercise 1:

**i. Declare variables**

*One variable of type "string" and name them appropriately:*

name - to store and represent the name;

*Four variables of type "long" and name them appropriately:*

phoneNumber - to store and represent the phone number;
interimRepositoryPhoneNumber - For temporary storage of telephone numbers;
fiveDigit - to store and represent the last five digits of the phone number;
sixDigit - to store and represent the First six digits of the phone number;

*Two variables of type "int" and name them appropriately:*

decimalNumber - to store and represent two-digit;
interimRepositoryDecimalNumber - For temporary storage of two-digit;

*One variable of type "float" and name them appropriately:*

temperature - to store and represent the temperature;

*Four variables of type "double" and name them appropriately:*

a - to store data and as quadratic coefficients of quadratic equations.
b - to store data and as a primary coefficient of a quadratic equation.
c - to store data and act as a constant term in a quadratic equation.
dscrmt - to store data and as a discriminant of a quadratic equation.

**ii. Displays a textual prompt on the screen, requesting the user to input the relevant content.**
**iii. Retrieve the input data from the keyboard and assign it to the appropriate variable.**
**iv. If the input format is erroneous, the user will be prompted to re-enter the information until the proper format is provided.**
**v. Upon successful reading of the data, the associated procedure is executed.**

*The specific data and their corresponding operations are as follows:*

Phone number - Compute the quotient and remainder by dividing the first six digits and the last five digits,

respectively. Then, divide the resulting numbers to find the quotient value.

Decimal numbers - Convert the given values into octal and hexadecimal representations utilising distinct display modes.

Temperature - The provided algorithm converts temperature readings to both Kelvins and Fahrenheit by utilising the temperature unit conversion formula. Additionally, it employs forced type conversion to round decimal points.

a,b,c - Firstly, it is necessary to ascertain whether the value of an is zero to establish the validity of the quadratic equation. Subsequently, the obtained data should be utilised in the discriminant formula to compute and assess the discriminant value. Finally, depending on the determined value of the discriminant, the roots of the quadratic equation can be calculated.

## 1.4 Analysis for exercise 2:

• On an input:

First, it is vital to furnish the client with a directive that prompts the initiation of a string input. • On an output:

A custom function is implemented to do a traversal search on a given target string within a specified search range. The method provides the client's first occurrence of the target character and displays it on the screen.

• Data structure:

Only one data is involved. No complex data structure difficulties here.

• Algorithm:

The equations are straightforward to assess. There are no problems with algorithms either.

## 1.5 Design for exercise 2:

**i. Declare variables**

*Two variables of type "string" and name them appropriately:*

source0 - to store and represent a source for research;

target - For temporary storage of a target string to research;

**ii. Define a function to find the target string (obj) in the source string and return the first match to the target string.**

**iii. Retrieve the input data from the keyboard and assign it to the appropriate variable.**

**iv. Verify whether the two pointers are null. If they are indeed null pointers, then return null pointers.**

**v. Use a while loop to review the source string until the null character is encountered. The function generates two constant character references, s and o, for each loop iteration. Initial values for these pointers are source and obj.**

**vi. Compare the characters referenced by s and o using an additional while loop. The comparison process reverses if the two characters are equivalent and the variable o reference is not the ending null character.**

**vii. The function uses const-cast to convert the source pointer to a non-constant pointer and return it as a reference to the matching location if the character pointed to by o is the end of the string.**

**viii. If there is no match at the end of the loop, the function moves the source pointer backwards and repeats until it has examined the full source string.**

**iX. When the loop terminates without a match, the function returns the source address as a non-constant pointer using const-cast to indicate that the target string was not found.**

**x. Display the function return value on the screen.**

## 1.6 Implementation

Please see the C++ source code Coursework1.cpp (at the end of this document) with comments.

## 1.7 Testing:

### 1.7.1 ===Test 1===

========== Exercise 1 ==========
Enter your name: Nie Xi
Enter your telephone number (11 digits): 18860900867
Enter a 2-digit decimal number: 17
Enter temperature in degrees Celsius: 21

———— Exercise 1:Q2 Q3 Q4 output ————
Division Result: 188609/867 = 217.542
Decimal: 17
Octal: 21
Hexadecimal: 11
21 °C – 70°F,294°K

———— Exercise 1:Q5 ————
Type three numbers to be the a,b,c in the quadratic equation(e.g.12 13 14)
1 4 4
There is only one real root of this function. The answer is-2
========== Exercise 2 ==========
Please provide a source for research: afghshjdvalj
Please provide a target to research: s
Return the pointer point to the:s

### 1.7.2 ===Test 2===

========== Exercise 1 ==========
Enter your name: haochuan jiang
Enter your telephone number (11 digits): 123123123123123
Formatting error. Please re-enter:
12312312312
Enter a 2-digit decimal number: 12
Enter temperature in degrees Celsius: 46

———— Exercise 1:Q2 Q3 Q4 output ————
Division Result: 123123/12312 = 10.0002
Decimal: 12
Octal: 14
Hexadecimal: c
46 °C – 115°F,319°K

———— Exercise 1:Q5 ————
Type three numbers to be the a,b,c in the quadratic equation(e.g.12 13 14)
1 9 6
There are two real roots of this function. The answer are-0.725083and-8.27492
========== Exercise 2 ==========
Please provide a source for research: dsjhsfbeisdfkj
Please provide a target to research: a
Return the pointer point to the:

### 1.7.3 ===Test 3===

========== Exercise 1 ==========
Enter your name: jiarui yu
Enter your telephone number (11 digits): 89089089089

Enter a 2-digit decimal number: 90
Enter temperature in degrees Celsius: 77

—————— Exercise 1:Q2 Q3 Q4 output ——————
Division Result: 890890/89089 = 10
Decimal: 90
Octal: 132
Hexadecimal: 5a
77 °C – 171°F,350°K

———————— Exercise 1:Q5 ————————
Type three numbers to be the a,b,c in the quadratic equation(e.g.12 13 14)
0 45 39
The answer is -0.866667
========== Exercise 2 ==========
Please provide a source for research: efwbirfkasjhhw23
Please provide a target to research: h
Return the pointer point to the:h

#### 1.7.4 ===Test 4===

========== Exercise 1 ==========
Enter your name: jiarui yu
Enter your telephone number (11 digits): 12345678909
Enter a 2-digit decimal number: 34
Enter temperature in degrees Celsius: 45

—————— Exercise 1:Q2 Q3 Q4 output ——————
Division Result: 123456/78909 = 1.56454
Decimal: 34
Octal: 42
Hexadecimal: 22
45 °C – 113°F,318°K

———————— Exercise 1:Q5 ————————
Type three numbers to be the a,b,c in the quadratic equation(e.g.12 13 14)
6 3 4
There is no real root of this function.
========== Exercise 2 ==========
Please provide a source for research: abcdef
Please provide a target to research: cf
Return the pointer point to the:

#### 1.7.5 ===Test 5===

========== Exercise 1 ==========
Enter your name: zhongying zhu
Enter your telephone number (11 digits): 44422200055
Enter a 2-digit decimal number: 94
Enter temperature in degrees Celsius: 10086

—————— Exercise 1:Q2 Q3 Q4 output ——————
Division Result: 444222/55 = 8076.76
Decimal: 94
Octal: 136
Hexadecimal: 5e

10086 °C -¿ 18187°F,10359°K

———————— Exercise 1:Q5 ————————
Type three numbers to be the a,b,c in the quadratic equation(e.g.12 13 14)
9 42 50
There is no real root of this function.
========== Exercise 2 ==========
Please provide a source for research: wjstfzjkbed
Please provide a target to research: tf
Return the pointer point to the:t

## 2 C++ Code

```{c++}
/*
Name: RBE104_Courseworks1
File Name: Courseworks1.cpp
Author: Xi Nie
Description:
    Exercise 1: Looping structures, two judgement structures, logical operators, forced type conversions, io streams, and basic operators solve most problems.
    Exercise 2: In Part1:Two loop structures, two judgement structures, and a const_free representation complete function internals.
                In Part2:Prompts the user for a string and calls the findC function.
*/
#include <iostream>/* include standard library iostream for use of cin, cout, getline, endl, oct, hex */
#include <cmath>/* include standard library cmath for use of sqrt() */
#include <iomanip>/*include to set the output precision of a floating-point number to n bits.*/
using namespace std;/*Using the std namespace eliminates the need for the full namespace qualifier. */

/*Exercise 2 Part1:
    Define a function.
    Use if loops to initialis pointers.
    Nested while loops and if judgements using while loops with arithmetic operations on pointers to change pointers.
    Use const_cast<>() to eliminate const effects.
*/
char *findC (char const *source, char const *obj){
    if (source == nullptr || obj == nullptr) {
        return nullptr;
    }

    while (*source != '\0') {
        char const * s = source;
        char const * o = obj;

        while (*s == *o && *o != '\0') {
            s++;
            o++;
        }

        if (*o == '\0') {
            return const_cast<char*>(source);
        }

        source++;
    }

    return const_cast<char*>(source);
}
```

```
int main() {
//____________________________Declare variable____________________________
    string name;
    long long phoneNumber,interimRepositoryPhoneNumber,fiveDigit,sixDigit;
    int decimalNumber,interimRepositoryDecimalNumber;
    float temperature;
    double a,b,c,dscrmt;

//____________________________Solution____________________________

//Exercise 1:Q1

cout<<"==================== Exercise 1 ====================="<<"\n"<<endl;
    cout << "Enter your name: ";
    getline(cin, name);
    cout << "Enter your telephone number (11 digits): ";
    cin >> interimRepositoryPhoneNumber;

    /*
    Use the do while structure to nest the if structure to complete the judgement on the number of
        digits.
    Use logical operators ||(or) juxtaposing conditions.
    Assign the correct result to the appropriate variable.
    */
    do{
        if (interimRepositoryPhoneNumber < 10000000000 || interimRepositoryPhoneNumber >
            99999999999)
        {
            cout<<"Formatting error. Please re-enter:"<< endl;
            cin >> interimRepositoryPhoneNumber;
        }
    } while (interimRepositoryPhoneNumber<10000000000 || interimRepositoryPhoneNumber>99999999999);
        phoneNumber = interimRepositoryPhoneNumber;

    cout << "Enter a 2-digit decimal number: ";
    cin >> interimRepositoryDecimalNumber;
    do{
        if (interimRepositoryDecimalNumber<10 || interimRepositoryDecimalNumber>99)
        {
            cout<<"Formatting error. Please re-enter:"<< endl;
            cin >> interimRepositoryDecimalNumber;
        }
    } while (interimRepositoryDecimalNumber<10 || interimRepositoryDecimalNumber>99);
        decimalNumber = interimRepositoryDecimalNumber;

    cout << "Enter temperature in degrees Celsius: ";
    cin >> temperature;


    cout<<"------------- Exercise 1:Q2 Q3 Q4 output -------------"<<"\n"<<endl;


/*Exercise 1:Q2
    Use divide 100000 to get each part of the number;
    Use forced type conversion to convert the long integer sixDigit to an SFP type, which can get
        an accurate solution.
*/
    sixDigit = phoneNumber / 100000;
    fiveDigit = phoneNumber % 100000;
    float divisionResult = static_cast<float>(sixDigit) / fiveDigit;
    cout << "Division Result: " << sixDigit << "/" << fiveDigit << " = " << divisionResult <<"\n"<<
        endl;


/*Exercise 1:Q3
```

```
    Use oct and hex.
*/
    cout << "Decimal: " << decimalNumber << endl;
    cout << "Octal: " << oct << decimalNumber << endl;
    cout << "Hexadecimal: " << hex << decimalNumber <<"\n"<< endl;


/*Exercise 1:Q4
    Compute the temperature into degrees Fahrenheit and Kelvin according to the formula.
    The "fixed" function ensures that floating-point numbers have a set number of decimal places.
    The "setprecision(0)" function generates integer-trimmed floating-point numbers.
    The "round()" method rounds floating-point integers to two decimal places.
*/
    cout<<temperature<<" C -- "<< fixed << setprecision(0)<<round((temperature*9/5+32)*100)/100<<"F
        ,"<<round((temperature+273.15)*100)/100<<"K"<<"\n"<<endl;


/*Exercise 1:Q5
    Use else if structure to discuss the different situations.
    Get the answers by the math calculation.
*/

    cout<<"--------------------- Exercise 1:Q5 ---------------------"<<"\n"<<endl;

    cout<<"Type three numbers to be the a,b,c in the quadratic equation(e.g.12 13 14)"<<endl;
    cin >>a>>b>>c;/*Assign the input values to the three variables a,b,c respectively.*/
    dscrmt = b*b-(4*a*c);/*Calculate the discriminant of equation*/
    if(a==0){
        cout<<"The answer is "<<-c/b<<"\n"<<endl;
    }else if ( dscrmt < 0){
        cout<<"There is no real root of this function. "<<"\n"<<endl;
    }else if(dscrmt==0){
        cout<<"There is only one real root of this function. The answer is"<<-b/(2*a)<<"\n"<<endl;
    }else{
        cout<<"There are two real roots of this function. The answer are"<<(-b+sqrt(dscrmt))/(2*a)
            <<"and"<< (-b-sqrt(dscrmt))/(2*a)<<"\n"<<endl;
    }

cout<<"==================== Exercise 2 ====================="<<"\n"<<endl;

/*Exercise 2 Part 2
    Use a variable to consume the return value
*/
    string source0, target;

    cout<<"Please provide a source for research: ";
    cin.ignore();
    getline(cin, source0);
    cout<<"Please provide a target to research: ";
    getline(cin, target);
    char* result = findC(source0.c_str(), target.c_str());
    cout << "Return the pointer point to the:" << *result << endl;
    return 0;
}
```

---

Notes:
single precision floating point number (SFP)
double precision floating point number (DFP)