# 实验：简单计算

樊子谦

软件学院 软件工程 2024级1班

【1】

编写程序将摄氏温度C转换为相应的华氏温度F。使用公式：F=9/5C+32；

要求输出格式，比如输入0摄氏度，输出32华氏度时：

Enter a Celsius temperature: 0

Fahrenheit equivalent: 32

非常简单的题目。只要注意公式中9/5会因为整形运算变成向下取整的除法导致错误结果。改成9.0/5或者1.0\*9/5即可。

#include <iostream>

using namespace std;

int main() {

double celsius, fahrenheit;

cout << "Enter a Celsius temperature: ";

cin >> celsius;

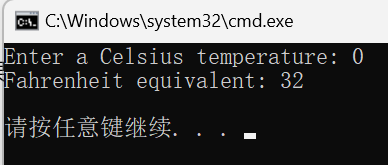
fahrenheit = 9.0 / 5 \* celsius + 32;

cout << "Fahrenheit equivalent: " << fahrenheit << '\n';

return 0;

}

结果：



【2】

编写程序求三个数的平均值；

要求输出格式，求5，7，11的平均值时，输出

Enter the first number: 5

Enter the second number: 7

Enter the third number: 11

Average is : 7.66667

直接累加，然后除以3即可。

#include <iostream>

using namespace std;

int main() {

double sum=0;

string message[]= {"first", "second", "third"};

for(int i = 0; i < 3; i++) {

double num;

cout << "Enter the " << message[i] << " number: ";

cin >> num;

sum += num;

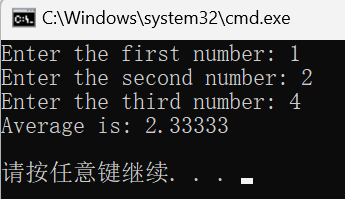
}

double average = sum / 3;

cout << "Average is: " << average << endl;

return 0;

}



【3】

编写数字提取应用程序。用户输入一个5位整数，程序将该数拆分为独立的数字并分行显示，数字前还有说明文字。例如输入12345时：

Enter five-difit number: 12345

Ones difit is: 5

Tens digit is: 4

Hundreds digit is: 3

Thousands digit is: 2

Ten-thousands digit is: 1

因为是从低位输出到高位，所以不断输出模10的结果然后除以10直到这个数字为0即可。

#include <iostream>

using namespace std;

int main() {

int number;

cout << "Enter a five-digit number: ";

while (cin >> number) {

if (number >= 10000 && number <= 99999)

break;

else

cout << "Invalid input. Please enter a five-digit number: ";

}

string messages[] = {"Ones", "Tens", "Hundreds", "Thousands", "Ten-thousands"};

for (int i = 0; i < 5; i++) {

int digit = number % 10;

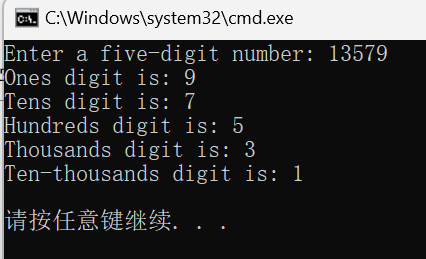
cout << messages[i] << " digit is: " << digit << '\n';

number /= 10;

}

return 0;

}



【4】

改写第三题程序，程序可以拆分任意位的整数，输出简化为：

Enter number：456

[1]：6

[2]：5

[3]：4

可以按照【3】的思路去写，但是这样就不符合“任意位整数”的要求，因为有存储上限。所以用字符串处理，逆序输出。题意没有规定输入的数字必然为正整数，所以需要处理输入负数（前面有负号）的情况。

#include <iostream>

#include <vector>

using namespace std;

int main() {

string number;

cout << "Enter number: ";

cin >> number;

if (number[0] == '-') {

cout << "Negative number\n";

number.erase(0, 1);

}

int n = number.size();

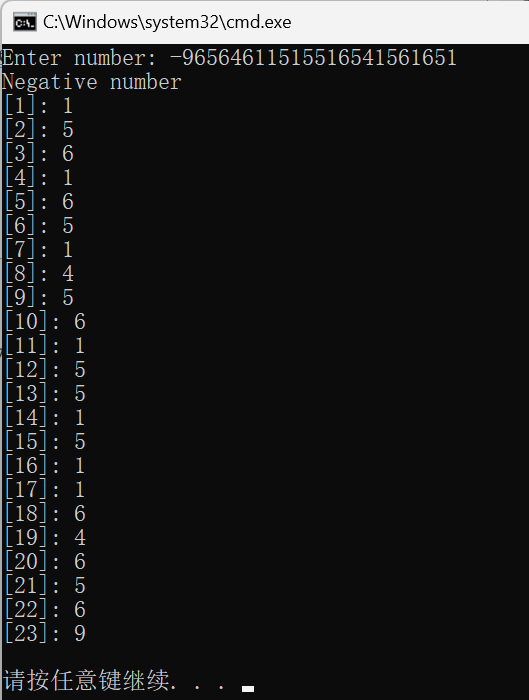
for (int i = 1; i <= n; i++) {

cout << "[" << i << "]: " << number[n - i] << "\n";

}

return 0;

}



程序不仅能分离位数，还能对负数正确处理。

心得：

实验有很多在控制台输出重复的内容的要求。为了简化操作，可以多使用循环和预处理提示来简化操作。【4】的出题意图可能并没有复杂，但鉴于写程序对严谨的需求，还是需要用更复杂的算法，来增加程序的稳定性。