习题一

编写一个程序，它将以米为单位的长度换算成相应的以英寸和英尺为单位的长度。你需要的换算关

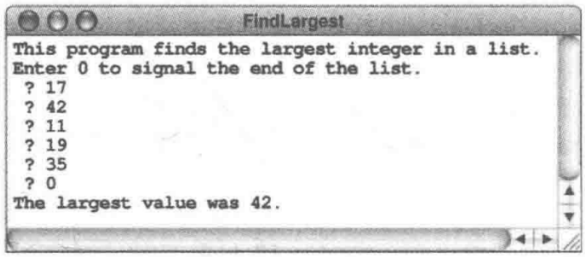
系为：

- 1英寸 = 0.0254米
- 1英尺 = 12英寸

```
void convertMetersToFeetAndInches(double meter, int &feet, double &inches) {
    int ft = feet;
    double is = inches;

    is = meter / 0.0254;
    ft = (int)(is / 12);
    is -= ft * 12;

    feet = ft;
    inches = is;
}
```

习题二

5. 编写一个程序，从用户处读取数据直到用户输入了 0 信号量为止。当信号量出现时，程序需要显示读取数据的最大值，如以下运行结果：



确保你的信号量是个常量而且易于改变。并且确保你的程序当输入数据均为负数时也能正确运行。

[illegible]

在测试终止条件前必须执行某些操作时，程序员碰到了称之为**循环和一半问题**（loop-and-a-half problem）的情况。C++ 提供的解决该问题的策略是使用 `break` 语句，除了 `break` 语句在 `switch` 语句中的应用外，它同样可以立即结束最内层的循环。采用 `break`，可能以下面遵循问题的本来结构的形式来编写循环结构，这种循环编程模式称为**读直到信号量模式**（read-until-sentinel pattern）：

42

```
while (true) {
   Prompt user and read in a value.
   if (value == sentinel) break;
   Process the data value.
}
```

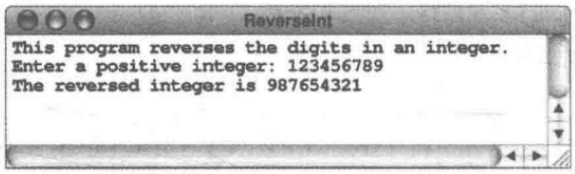
```cpp
#include <iostream>
using namespace std;


const int SENTINEL = 0; // 常量并且易于修改。

int findLargest() {
    cout << "This program finds the largest integer in a list. \n"
         << "Enter 0 to signal the end of the list."
         << endl;
    int total_num;
    int max = 0;
    while (true) {
        cout << "? ";
        cin >> total_num;
        if (total_num == SENTINEL) break;
        if (total_num > max) {
            max = total_num;
            // cout << max << endl;
        }
    }
    return max;
}


```

```cpp
int main() {
    int largest = findLargest();
    cout << "The largest number is: " << largest << endl;
    return 0;
}
```

习题三

```cpp
int digitSum(int n) {
    int sum = 0;
    while (n > 0) {
        sum += n % 10;
        n /= 10;
    }
    return sum;
}
```

8. 使用“while 语句”那节的 digitSum 函数作为模板，编写一个程序，读取一个整数，然后逆序输出该整数中的各位数，如以下示例运行结果：




```cpp
int reverseInt(int n) {
    int ri = 0;
    while (n > 0) {
        ri = ri * 10 + n % 10;
        n /= 10;
    }

    return ri;
}

int main() {

    int digit;

    cout << "This program reverses the digits in an integer." << endl;
    cout << "Enter a positive integer: ";
    cin >> digit;

    cout << "The reversed integer is " << reverseInt(digit) << endl;

}
```