

复习题

题目:

1. 入口条件循环和出口条件循环之间的区别是什么? 各种 C++ 循环分别属于其中的哪一种?
2. 如果下面的代码片段是有效程序的组成部分, 它将打印什么内容?

```
int i;
for (i = 0; i < 5; i++)
    cout << i;
    cout << endl;
```

3. 如果下面的代码片段是有效程序的组成部分, 它将打印什么内容?

```
int j;
for (j = 0; j < 11; j += 3)
    cout << j;
cout << endl << j << endl;
```

4. 如果下面的代码片段是有效程序的组成部分, 它将打印什么内容?

```
int j = 5;
while ( ++j < 9)
    cout << j++ << endl;
```

5. 如果下面的代码片段是有效程序的组成部分, 它将打印什么内容?

```
int k = 8;
do
    cout << " k = " << k << endl;
while (k++ < 5);
```

6. 编写一个打印 1、2、4、8、16、32、64 的 for 循环, 每轮循环都将计数变量的值乘以 2。
7. 如何在循环体中包括多条语句?
8. 下面的语句是否有效? 如果无效, 原因是什么? 如果有效, 它将完成什么工作?

```
int x = (1,024);
```

下面的语句又如何呢?

```
int y;
y = 1,024;
```

9. 在查看输入方面, `cin >> ch` 同 `cin.get(ch)` 和 `ch=cin.get()` 有什么不同?

1.

入口循环指的是: 当满足条件的时候才 "进入循环", 即: 先判断, 再循环 出口循环: 先执行一次循环体, 之后马上进入判断阶段, 当满足条件的时候 "才执行第二次循环"

- 入口循环: while, for
- 出口循环: do-while

2.

循环体语句只有 `cout i;`, 所以最后打印了 01234, 然后换行.

3.

打印 0369, 换行后再输出12, 最后再换行一次

4.

```
// ++j 是先将j自增, 然后才进行本行语句中关于j的其他操作
第一轮: 打印6 // 此时由于j++, j变成了7, 然后换行
第二轮: 打印8 // ++j 把j从7变成了8, 判断 j<9 成立, 打印8, 然后 j++ 把j从8变成了9, 然后换行
```

5.

```
k = 8 // do
// 由于是 k++, k先和5比较, k<5 此时结果为 false, 不再执行循环体, 但是k最后自增到9, 程序结束
```

6.

```
for (int i = 1; i < 65; i *= 2){
    cout << i << "\、 ";
}
```

7.

使用 `{}` 即可在循环中添加多条语句

8.

```
int = (1,1024); // 语句合法. 分析: 用括号括起来的部分先执行, 也就是先执行 "1,024", 这个语句返回 024, 以0
开头, 表示8进制数, int默认使用十进制数表示, 所以最后 int 存放的是十进制的20.
```

```
int y;  
y = 1,024;    // 语句是合法有效的。分析：这个语句与上面语句的区别是没有括号，此时，y=1优先执行，y被赋值，  
然后，024被执行，整条语句最后的结果是024，但是这个语句没有任何意义。
```

9.

```
cin >> ch;    // 输入的时候会忽略所有的空白字符(空格，制表符，enter)  
  
cin.get(char);  
// 和  
ch = cin.get();  
// 上面这两个语句的效果是相同的，它们都读取所有字符，并转换成ASCII码字符存放到ch中，包括空格，制表符，enter
```

编程题部分

5.9 编程练习

1. 编写一个要求用户输入两个整数的程序。该程序将计算并输出这两个整数之间（包括这两个整数）所有整数的和。这里假设先输入较小的整数。例如，如果用户输入的是 2 和 9，则程序将指出 2~9 之间所有整数的和为 44。

2. 使用 `array` 对象（而不是数组）和 `long double`（而不是 `long long`）重新编写程序清单 5.4，并计算 100! 的值。

3. 编写一个要求用户输入数字的程序。每次输入后，程序都将报告到目前为止，所有输入的累计和。当用户输入 0 时，程序结束。

4. Daphne 以 10% 的单利投资了 100 美元。也就是说，每一年的利润都是投资额的 10%，即每年 10 美元：

$$\text{利息} = 0.10 \times \text{原始存款}$$

而 Cleo 以 5% 的复利投资了 100 美元。也就是说，利息是当前存款（包括获得的利息）的 5%，：

$$\text{利息} = 0.05 \times \text{当前存款}$$

Cleo 在第一年投资 100 美元的盈利是 5%——得到了 105 美元。下一年的盈利是 105 美元的 5%——即 5.25 美元，依此类推。请编写一个程序，计算多少年后，Cleo 的投资价值才能超过 Daphne 的投资价值，并显示此时两个人的投资价值。

5. 假设要销售《C++ For Fools》一书。请编写一个程序，输入全年中每个月的销售量（图书数量，而不是销售额）。程序通过循环，使用初始化为月份字符串的 `char *` 数组（或 `string` 对象数组）逐月进行提示，并将输入的数据储存在一个 `int` 数组中。然后，程序计算数组中各元素的总数，并报告这一年的销售情况。

6. 完成编程练习 5，但这一次使用一个二维数组来存储输入——3 年中每个月的销售量。程序将报告每年销售量以及三年的总销售量。

7. 设计一个名为 `car` 的结构，用它存储下述有关汽车的信息：生产商（存储在字符数组或 `string` 对象中的字符串）、生产年份（整数）。编写一个程序，向用户询问有多少辆汽车。随后，程序使用 `new` 来创建一个由相应数量的 `car` 结构组成的动态数组。接下来，程序提示用户输入每辆车的生产商（可能由多个单词组成）和年份信息。请注意，这需要特别小心，因为它将交替读取数值和字符串（参见第 4 章）。最后，程序将显示每个结构的内容。该程序的运行情况如下：

```
How many cars do you wish to catalog? 2
Car #1:
Please enter the make: Hudson Hornet
Please enter the year made: 1952
Car #2:
Please enter the make: Kaiser
Please enter the year made: 1951
Here is your collection:
1952 Hudson Hornet
1951 Kaiser
```

8. 编写一个程序，它使用一个 `char` 数组和循环来每次读取一个单词，直到用户输入 `done` 为止。随后，该程序指出用户输入了多少个单词（不包括 `done` 在内）。下面是该程序的运行情况：

```
Enter words (to stop, type the word done):
anteater birthday category dumpster
envy finagle geometry done for sure
You entered a total of 7 words.
```

您应在程序中包含头文件 `cstring`，并使用函数 `strcmp()` 来进行比较测试。

9. 编写一个满足前一个练习中描述的程序，但使用 `string` 对象而不是字符数组。请在程序中包含头文件 `string`，并使用关系运算符来进行比较测试。

10. 编写一个使用嵌套循环的程序，要求用户输入一个值，指出要显示多少行。然后，程序将显示相应行数的星号，其中第一行包括一个星号，第二行包括两个星号，依此类推。每一行包含的字符数等于用

户指定的行数，在星号不够的情况下，在星号前面加上句点。该程序的运行情况如下：

```
Enter number of rows: 5
....*
...**
..***
.****
*****
```

第一题

```
# include <iostream>

int main(void) {
    using std::cout;
    using std::endl;
    using std::cin;

    int min,max,sum=0;
    cout << "enter the min number:";
    cin >> min;
    cout << "enter the max number:";
    cin >> max;

    for (int i = min; i <= max; i++) {
        sum += i;
    }
    cout << "sum = " << sum << endl;

    return 0;
}
```

第二题：

```
#include <iostream>
#include <array>

const int ArrSize = 16;

int main(void){
    using namespace std;
    // long long factorials[ArrSize];    // 原来的代码
    array<long long, ArrSize> factorials;

    factorials[0] = factorials[1] = 1LL; // long long 类型

    for (int i=2; i<ArrSize; i++){
        factorials[i] = i * factorials[i-1];
    }

    for (int i =0; i < ArrSize; i++)
        std::cout << i << "! = " << factorials[i] << std::endl;
    return 0;
}
```

第三题：

```
#include <iostream>

int main(void){
    using namespace std;

    double num = 0, sum = 0;

    do {
        cout << "Please enter a number: ";
        cin >> num;
        sum += num;
    }
    while (num != 0);        // while 后面别忘了加分号!!

    cout << "The sum of previous num is: " << sum << endl;
    return 0;
}
```

第四题

```
#include <iostream>

const int DEPOSIT_BASE = 100;    // 初始存款是一样的

int main(void){

    using namespace std;

    double daphne_deposit = DEPOSIT_BASE;
    double Cleo_deposit = DEPOSIT_BASE;

    int year = 0;

    while (daphne_deposit >= Cleo_deposit){
        daphne_deposit += 10;
        Cleo_deposit += Cleo_deposit * 0.05;
        year++;
    }

    cout << "After " << year << " years, Cleo's has more money than daphne.\n";

    cout << "Daphne has " << daphne_deposit << " dollars deposit.\n";
    cout << "Cleo has " << Cleo_deposit << " dollars deposit.\n";
    return 0;
}
```

第五题

```
#include <iostream>
#include <string>

int main(void){
```

```

using namespace std;

// const string months[] = {"1","2","3","4","5","6","7","8","9","10","11","12"}; // 数组别忘了[s]
const string Month [] =
{"January","February","March","April","May","June","July","August","September","October","November","December"};

int sale_num[12], sum = 0;

for (int i = 0; i < 12; i++)
{
    cout << "Enter the sale number of " << Month[i] << ": ";
    cin >> sale_num[i];
}

cout << "Input done.";

for (int i = 0; i < 12; i++){
    sum += sale_num[i];
}
cout << "The total sale number is " << sum << endl;
return 0;
}

```

第六题

```

#include <iostream>
#include <string>

int main(void){

    using namespace std;

    const string Month [] =
{"January","February","March","April","May","June","July","August","September","October","November","December"};
    int sale_num[3][12], sum = 0;

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

        cout << "Starting " << i+1 << " year's data" << endl;
        for (int j = 0; j < 12; j++){
            cout << "Enter the sale number of " << Month[j] << ": ";
            cin >> sale_num[i][j];
        }
    }
    cout << "Input Done! \n";

    for (int i = 0; i < 3; i++){
        for (int j = 0; j < 12; j++){
            sum += sale_num[i][j];
        }
    }
    cout << "Total number of sale books for 3 years is " << sum << endl;

    return 0;
}

```

第七题(考察结构体)

- 由于数字和字符串交替输入, 此时, 数字后的回车会在输入缓冲区中, 我们需要使用 `cin.get()` 清空这个回车, 再读取字符串
- 使用了 `new` 创建数组, 一定记得释放内存!!!

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

struct car {
    string manufacture;    // 生产生
    int date;              // 生产日期
};                          // 别忘记分号!!!

int main(void){

    int car_number;
    cout << "How many cars do you wish to catalog? ";
    (cin >> car_number).get();    // 记得用 get() 吃掉回车

    car * pcar = new car[car_number];

    for (int i = 0; i < car_number; i++){
        cout << "Car #" << i+1 << ": \n";
        cout << "Please enter the maker: ";
        getline(cin, (pcar+i)->manufacture);    // 指针名字在一定程度上可以和数组名互换使用
        cout << "Please enter the year made: ";
        (cin >> pcar[i].date).get();              // 记得吃掉数字后的回车
    }

    cout << "Here is your collection: \n";
    for (int i = 0; i < car_number; i++){
        cout << (pcar+i)->date << " " << (pcar+i)->manufacture << endl;
    }

    delete [] pcar;

    return 0;
}
```

第八题

```
#include <iostream>
#include <cstring>

using namespace std;

const char DONE[] = "done";

int main(void){
```



```

char word[20];    // 可接收19个英文字母+1个空白字符
int count = 0;

cout << "Enter words (to stop, type done ): \n";

do {
    cin >> word;
    cin.get();    // 干掉回车
    count++;      // 在这里count++的时候，第一次输入就是done也会变成1，所以最后输出的时候要-1

}while (strcmp(word, DONE)) ;

cout << "You enter a total of " << count-1 << " words.\n";    // 这里要-1，因为最后一次输入的是"done"，也被算进去了
return 0;
}

```

第九题

```

#include <iostream>
#include <string>

using namespace std;

const string DONE = "done";

int main(void){
    string word;    // 可接收19个英文字母+1个空白字符
    int count = 0;

    cout << "Enter words (to stop, type done ): \n";

    do {
        cin >> word;
        cin.get();    // 干掉回车
        count++;      // 在这里count++的时候，第一次输入就是done也会变成1，所以最后输出的时候要-1

    }while (word != DONE);

    cout << "You enter a total of " << count-1 << " words.\n";    // 这里要-1，因为最后一次输入的是"done"，也被算进去了
    return 0;
}

```

第十题

- 思路:
 - 嵌套循环里 有两个 for, 一个打印 . 另一个打印 *

```

#include <iostream>
using namespace std;

int main(void){
    int row;

```

```
cout << "Please enter the number of rows: ";
cin >> row;

// 按行打印
for (int i = 0; i < row; i++){
    // 打印点
    for(int j = 0; j < row - i -1; j++){
        cout << ". ";
    }
    for (int j = 0; j <= i; j++){
        cout << "* ";
    }
    cout << endl;
}
return 0;
}
```