

期中习题课

期中考试分析

- ▶ 简答题：考核的是基本的核心概念，程序设计必须要掌握的知识点。
- ▶ 分析程序题：读懂别人的代码
 - ▶ 分析错误原因：首先分析题目的编程要求，然后分析代码中哪些地方无法达到编程要求。
 - ▶ 分析程序结果：从main函数开始阅读，分析程序的框架（流程），模拟调试，程序单步执行结果。
- ▶ 编程题：
 - ▶ 分析题目的编程要求
 - ▶ 分析程序的流程，和OJ不同，程序框架有对应的分值。
 - ▶ 编写代码：注意缩进、{ }、==、.....

程序设计的思想和原理

- ▶ 通过学习教材上的例题，实践课后习题来理解程序设计的基本思想和原理。——知其然
- ▶ 在什么情况下使用什么样的求解方案，理解这些思想背后的原理。——知其所以然
- ▶ 掌握了设计思想和原理，就能将已学的知识推广到解决新的问题。——举一反三
- ▶ 程序设计思想可以适用于不同的程序设计语言的，因此掌握好程序设计方法和思想有助于今后学习其他编程语言。

程序设计的过程

- ▶ 程序设计本身：问题求解的一种描述形式。
- ▶ 程序设计的过程：
 - ▶ 分析问题：程序需要完成什么、人机交互方式是怎样的
 - ▶ 程序设计：程序的流程图，功能分解（自顶向下）
 - ▶ 编码实现：对照框架图，给出功能的实现（逐步求精）
 - ▶ 调试+测试：输入测试数据，观察运行结果
 - ▶ 运行+维护：在运行过程会遇到前期设计未考虑到的情况，需要维护升级

处理程序设计中的错误

▶ 学会处理程序中的错误:

- ▶ 课程网站上有调试方法介绍

- ▶ 加入输出标签

▶ 初学者容易发生的错误:

- ▶ 语法错误: 编译可以发现

- ▶ 逻辑错误: 运行测试可以发现

▶ 如何减少错误的发生:

- ▶ 把问题分析清楚, 理顺程序的处理逻辑, 合理设计框架

- ▶ 语法和数据类型的正确使用

- ▶ 良好的程序设计风格

- ▶ 编程能力的提高没有秘籍, 唯有千锤百炼

第三章 知识点

- ▶ 顺序执行：依次执行语句
- ▶ 选择执行：根据条件判断结果，执行语句
- ▶ 循环执行：满足条件的前提下，循环执行语句
- ▶ 无条件转移：break continue 往往是和条件判断结合在一起进行流程转移。
- ▶ 学会分析程序流程，逐步精化。
 - ▶ 用文字，分步描述流程
 - ▶ 用流程图描述

选择语句（一）

- ▶ if语句：if(条件) 语句
- ▶ 两路分支：if(条件) 语句1 else 语句2
 - ▶ 类比：三目运算符 条件？表达式1：表达式2
- ▶ else是和最近的一个没有配对的if配对

选择语句（二）

► 多路分支：switch

switch语句是语法最为复杂的语句

switch case break default

switch和第五章的枚举结合

编写一个点单程序，假设客户一次只点一种食品，根据用户的选择，输出餐费的总额。

| | | | |
|--------|-------|----|--------|
| 劲脆鸡腿堡 | 29元/个 | 蛋挞 | 3.5元/个 |
| 新奥尔良烤翅 | 11元/对 | 薯条 | 5.5元/袋 |
| 可乐 | 3元/杯 | 蛋筒 | 3.5元/个 |

思考：如果用户一次点多种食品，如何处理？

循环语句： while do-while for

- ▶ while和do-while：循环次数不确定，while可能执行0次；do-while至少执行1次
- ▶ for：循环次数确定，或者循环变量参与循环体计算，优先用for

for (表达式1; 表达式2; 表达式3)

循环体

请注意for的执行流程(教材上有流程图)

无条件转移：continue break

- ▶ break：有两种用法(switch、循环)
- ▶ continue 和 break 在循环体中的区别

```
int main()
{ for (int j = 1; j <= 10; j++)
  { cout << "j = " << j << endl;
    if (j == 3)
    { j = 8;
      continue;
    }
  }
  return 0;
}
```

练习：单词个数

- 输入一个英文句子，两个单词之间只有一个空格，句首句尾无空格。编程实现输出该句子中的单词个数。

思路一：用数组

英文句子==>包含空格的字符串 ==> 字符数组

```
char sen[100];
```

```
cin>>sen; //不行，因为cin以空白符作为输入结束
```

```
cin.getline(sen,100); // 教材323页有说明
```

```
用while(sen[i]!='\0') 循环，sen[i]==' ' 计数器+1;
```

思路二：不用数组

英文句子==>输入字符序列，回车作为输入结束

```
char ch;
```

```
while((ch=cin.get())!='\n') ch==' ' 计数器+1;
```

练习：放礼炮

- ▶ 在某典礼上，有甲，乙，丙三门军舰要同时开始鸣放礼炮各 n 响。已知甲礼炮每3秒放一次，乙礼炮每4秒放一次，丙礼炮每5秒放一次。请编写程序计算，当A,B,C三门礼炮都鸣放完毕时，观众总共可以听到几次礼炮声。注意：多门礼炮同时鸣放只能听到一声。

例如：各放3响，观众可以听到7声，

分别发生在第0，3，4，5，6，8，10秒。

- ▶ 解题思路：

- ▶ 放礼炮的时间：从0秒~？

- ▶ 每秒能否听到炮声：和什么有关

练习：自守数

- ▶ 自守数：一个自然数的平方的尾数等于该数自身

例如： $25 \times 25 = 625$

编程实现输出10000以内的自守数

- ▶ 思考程序框架

对10000以内的每一个数进行判断

符合循环语句的应用场景，并且是计数循环

循环条件和步长：`for(int i= 0;i<=10000;i++)`

循环体：根据自守数的规则进行判断

符合条件，则输出

自守数判断循环体：思考一

► 思考一：

平方数的尾数和原来的数相同，可以考虑用%

$$25 * 25 = 625 \implies 625 \% 100 = 25$$

计算25的位数：2位 \implies 平方数对10的2次幂求余数

问题：

整型数是有范围的，大整数的平方数会超过范围

如何处理？

自守数判断循环体：思考二

► 思考二：如何解决大整数的平方数？

$$\begin{array}{r} 25 \\ * 25 \\ \hline 125 \rightarrow 5 * 25 \\ 50 \rightarrow 2 * 25 \\ \hline 625 \end{array}$$

$$\begin{array}{r} 625 \\ * 625 \\ \hline 3125 \rightarrow 5 * 625 \\ 12500 \rightarrow 2 * 625 \\ 37500 \rightarrow 6 * 625 \\ \hline 625 \end{array}$$

自守数判断循环体实现

- ▶ 计算自然数的位数 K
- ▶ 自然数的每个数字*自然数的乘积：
 - ▶ 个位数相乘的乘积结果取 K 位尾数；
 - ▶ 十位数相乘的乘积结果取 $K-1$ 位尾数*10；
 - ▶ 百位数相乘的乘积结果取 $K-2$ 位尾数*100；
 - ▶

练习：统计交通流量

1、路边设置一车辆探测器，探测器用线路连接到计算机。

(1) 当有车辆通过时，探测器传送信号1给计算机。

(2) 探测器中有一计时器，每秒钟发出一个数字信号2传给计算机。

(3) 该计算机从开始探测时计时，探测结束时传递一个数字信号0给计算机。

编写程序读入这一系列的信号并输出：

1 2 1 1 2 2 1 2 1 2 0

①进行了多长时间的统计调查；

②记录到的车辆数；

③在车辆之间最长的时间间隔是多少。

5秒内，有5辆车通过，
最长隔2秒有车通过

分析：

输入信号(sign) {
1(车辆通过)
2(秒钟信号)
0(探测结束)

输出结果 {
①进行统计的时间(seconds)
②记录到的车辆数(nums)
③在车辆之间最长的时间间隔(longest)
由此派生出的车辆间隔变量(inter)

根据以上数据特性，可确定数据类型为整型(int)

设计：（自顶向下，逐步求精）

一级流程

```
begin
    1.数据定义及初始化
    2.读入探测信号sign
    3.while( sign != 0 )
        对sign进行处理
    4.输出结果
end
```

二级流程

```
while( sign != 0 )
{   if ( sign == 1 )
    3.1处理车辆信号;
    else if ( sign == 2 )
    3.2处理计时信号;

    3.3读入下一个sign;
}
```

三级流程

```
begin
    3.1.1 车辆计数nums++;
    3.1.2 处理最大间隔
        if(longest<inter)
            longest=inter;
    3.1.3 为下一个间隔计数作准备
        inter=0;
end
```

```
begin
    3.2.1 总时间累加器seconds++
    3.2.2 间隔计数器inter++;
end
```

第四章 函数基本概念

- ▶ 子程序：完成一个独立的子功能。子程序用函数实现。调用函数，可以减少代码重复，以及过程抽象。
- ▶ 学会功能分解，用函数实现某个功能。
- ▶ 函数的定义：返回值类型、函数名、形参和函数体。

练习：输出n以内的所有完数（将完数的判断定成函数）

- ▶ 完数：一个整数等于其所有除自身之外的因子之和，比如
 $6=1+2+3$

程序流程：

- (1) 输入n;
- (2) 循环1~n次，调用函数判断是不是完数，是则输出。

完数的判断函数：

- (1) 返回类型：bool
- (2) 函数名：见名知意
- (3) 形参：对整数进行判断 int x
- (4) 函数体：求因子和，判断是否等于x

多模块程序设计

- ▶ 一个C++模块一般包含两个部分：
 - ▶ 接口（.h文件）：给出在本模块中定义的、提供给其它模块使用的一些程序实体的声明（如：函数、全局变量等）和定义（如类型、常量等）。
 - ▶ 实现（.cpp文件）：给出了模块中的程序实体的定义。
- ▶ 在模块A中要用到模块B中定义的程序实体时，可以在A的.cpp文件中用文件包含命令（#include）把B的.h文件包含进来，这样就起到声明的作用。

练习：我的字符串处理函数（多模块编程，不考虑溢出）

```
#include <iostream>
#include "mystring.h"
using namespace std;
int main()
{ char str1[20] = "hello ";
  char str2[4] = "NJU";
  str_cat(str1,str2);
  cout<<str1<<endl; //输出: hello NJU
  int len = str_len(str1);
  char str3[len+1];
  str_cpy(str3,str1,len); //往str3中拷贝str1的前len个字符;
  cout<<str3<<endl; //输出: hello NJU
  char str4[4] = "PKU";
  if(str_cmp(str2,str4)>0) cout<<str2;
  else cout<<str4; //输出: PKU
```

```
str_replace(str1,str2,str4); //注意存在多个str2
//在str1中找到str2，并用str4替换
cout<<str1<<endl; //输出: hello PKU
char str5[3] = "my";
int index = str_find(str1,str4);
//str4的第一个字符在str1中的位置
str_insert(str1,index,str5);
//在下标index处插入str5;
cout<<str1<<endl; //输出: hello myPKU
to_lower(str1); //将str1中的所有大写字母改为小写
cout<<str1<<endl; //输出: hello mypku
return 0;
}
```

如何设计mystring.h mystring.cpp

- ▶ mystring.h 中给出函数声明
 - ▶ mystring.cpp 中给出函数的具体实现
 - ▶ 第六次作业：课后完成
 - ▶ VS、Dev c++ 要建立项目，编码编译。
 - ▶ vs code 根据终端类型，在命令行输入下面的命令
- powershell终端窗口：
- ```
g++ main.cpp mystring.cpp -o string.exe ; .\string.exe
```
- cmd终端窗口：
- ```
g++ main.cpp mystring.cpp -o string.exe && string.exe
```


讨论：打字练习小程序

- ▶ 分析问题：用户对于打字练习有什么需求
- ▶ 程序设计：根据需求分解出哪些功能
- ▶ 编码实现：根据设计，完成编码编辑、编译
- ▶ 调试+测试：输入测试数据，观察运行结果是否符合预期
 - ▶ 不符合，找到原因，修正代码
- ▶ 运行+维护：在运行过程会遇到前期设计未考虑到的情况，需要维护升级

用户对于打字练习有什么需求

- ▶ 练习分难度：字母练习、句子、段落练习。
- ▶ 计时：记录完成一个练习的时间；限时练习。
- ▶ 练习结果：反馈练习的正确率、打字速度、易错字符统计、排行榜。
- ▶ 用户注册：用户注册（用户名、密码），注册是否成功（用户名是否别占用，密码是否有效）。
- ▶ 用户登录：如果用户名和密码输入错误，允许再次输入（多少次）；登录成功后，选择练习难度进行练习；练习完，可以查看练习记录。
- ▶ 管理员：打字练习文本的增加、修改、删除等；帮助用户重置密码。
- ▶ 程序相关文件的输入输出：启动程序的时候，自动从给定的文件中读入信息，初始化各项信息；当用户关闭系统时，系统自动保存所有信息到文件中，以便下一次启动系统时能够初始化系统。
- ▶