5 阅读程序与完善程序

洛谷网校
kkksc03
2023年8月4日

上课纪律

1. 按时参加各项活动。
2. 上课认真听讲，不要做其他事情，比如玩游戏。
3. 上课时，不能刷屏（短时间多次发言），不能发课程无关内容，不能发影响助教和其他同学的内容，更不能开玩笑和辱骂。
4. 提问区禁止发送非学术提问的内容。
5. 违反以上纪律的人，有可能会被禁言。
6. 禁言后要手写书面检讨才能解开。

课前说明

我个人一直认为，第一轮是无法速成的。

需要C++能力、算法能力、数学分析的能力比较强。

许多技巧也只是纸上谈兵。

你可能还是看不懂代码。

即使学完本课，也不保证有突飞猛进的进步。

但了解一些基本形式，总的来说还是好的。

阅读程序题型

阅读程序

题型介绍

- 给定一个C++代码。
- 阅读这个代码，并且尝试理解程序在做什么。
- 完成题目，包括判断题和选择题。

题数

- 三个大题，每个大题5-7个小题。总分40 分
- 前面是若干个判断题，后面是若干个选择题。
- 最大的一个部分!

例子

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断题正确填√，错误填×；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

(1)

```
01 #include <iostream>
02
03 using namespace std;
04
05 int main()
06 {
07     unsigned short x, y;
08     cin >> x >> y;
09     x = (x | x << 2) & 0x33;
10     x = (x | x << 1) & 0x55;
11     y = (y | y << 2) & 0x33;
12     y = (y | y << 1) & 0x55;
13     unsigned short z = x | y << 1;
14     cout << z << endl;
15     return 0;
16 }
```

例子

假设输入的 x、y 均是不超过 15 的自然数，完成下面的判断题和单选题：

- 判断题

16. 删去第 7 行与第 13 行的 unsigned，程序行为不变。（ ）

17. 将第 7 行与第 13 行的 short 均改为 char，程序行为不变。（ ）

18. 程序总是输出一个整数“0”。（ ）

19. 当输入为“2 2”时，输出为“10”。（ ）

20. 当输入为“2 2”时，输出为“59”。（ ）

- 单选题

21. 当输入为“13 8”时，输出为（ ）。

A. “0”　　B. “209”　　C. “197”　　D. “226”

程序的类型

程序类型非常的多样。可能包括:

- 没啥实际意义的语言性质练习题;
- 利用某些算法和数据结构的简单题目;
- 有一点复杂的模拟题;
- 动态规划题
- 考察C++各种语言特性的题目; （C++14标准）
- 等等……

题型 – 复杂度分析题

设问方式

程序（或者是其中的一部分）的时间复杂度是?

例子

25.算法 g(n,m)最为准确的时间复杂度分析结果为（ ）。

A. $O(n^{3/2}m)$　　B. $O(nm)$　　C. $O(n^2m)$　　D. $O(nm^2)$

解题思路

- 先确定是否是熟悉的算法，根据算法的时间复杂度。
- 确定循环层数，和每层的规模。
- 注意调和级数、二分等特殊情况。
- 极少的递归函数，可能需要用上主定理。

题型 – 程序改动题

设问方式

程序某语句改成另外一种，程序的结果不变。一般是判断题

例子

29. 删除第 23 行的强制类型转换，程序的行为不变。（ ）

17. 将第 7 行与第 13 行的 short 均改为 char，程序行为不变。（ ）

解题思路

- 先确定相应语句的作用。
- 两个语句进行比较，比较差异。尝试构造引起差异的数据。
- 代入数据，判断是否一致。

题型 – 阅读程序写结果题

设问方式

当输入是xxx的时候，应当输出什么？选择判断都有

例子

31.当输入为“100 7”时，输出为（ ）。

A. 202　　B. 1515　　C. 244　　D. 1754

解题思路

- 先确定程序的整体功能。
- 根据功能，处理输入数据。
- 注意经常会出现边界情况，需要注意。

题型 – 中间结果题

设问方式

程序某部分运行了多少次?

某个变量在某个情况下的值是多少?

例子

22. 当输入为“7 3”时，第 19 行用来取最小值的 min 函数执行了 449 次。（ ）

31. 该程序有存在缺陷。当输入的 n 过大时，第 12 行的乘法有可能溢出，因此应当将 mid 强制转换为 64 位整数再计算。（ ）

解题思路

- 确定程序的功能，和变量的功能。
- 根据上下文，模拟步骤，计算这个变量相关数据。

题型 – 计算细节题

设问方式

程序某部分是什么功能?

计算或者构造数据，使得满足一定的需求。

例子

20. f(a, b)与下列（ ）语句的功能最**类似**。

A. a.find(b)　　B. a.rfind(b)

C. a.substr(b)　　D. a.compare(b)

解题思路

- 理解程序的功能;
- 观察测试数据和输入输出的性质。

破题手段

可以按照这样的步骤完成题目：

- 拿到阅读程序题以后，尽可能首先以宏观的角度分析程序。
- 看一看题目和数据范围，看看是否有提示功能的地方。
- 阅读程序，观察有没有熟悉的算法，猜测功能。
- 对于一些确定的功能，可以做笔记。
- 对小的数据进行模拟分析（看程序写结果题）。
  使用草稿本模拟。

例子

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断题正确填√，错误填×；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

数字常量:

0x33（16进制）；013（8进制）；1LL（long long）

(1)

```
01 #include <iostream>
02
03 using namespace std;
04
05 int main()
06 {
07     unsigned short x, y;
08     cin >> x >> y;
09     x = (x | x << 2) & 0x33;
10     x = (x | x << 1) & 0x55;
11     y = (y | y << 2) & 0x33;
12     y = (y | y << 1) & 0x55;
13     unsigned short z = x | y << 1;
14     cout << z << endl;
15     return 0;
16 }
```

假设输入的 x、y 均是不超过 15 的自然数，完成下面的判断题和单选题：

判断题

16. 删去第 7 行与第 13 行的 unsigned，程序行为不变。（ ）

17. 将第 7 行与第 13 行的 short 均改为 char，程序行为不变。（ ）

18. 程序总是输出一个整数“0”。（ ）

19. 当输入为“2 2”时，输出为“10”。（ ）

20. 当输入为“2 2”时，输出为“59”。（ ）

单选题

21. 当输入为“13 8”时，输出为（ ）。

A. “0”　　B. “209”　　C. “197”　　D. “226”

所涉及的知识

优先级	运算符	说明	结合性
1	（） [] .(成员选择(对象)) ->(成员选择(指针))		从左到右
2	! +(正) -(负) ~ ++ --	单目运算符	从右到左
3	*(乘) /(除) %(取余)	双目运算符	从左到右
4	+(加) -(减)		从左到右
5	<<(左移) >>(右移) >>>		从左到右
6	>(大于) >=(大于等于) <(小于) <=(小于等于)		从左到右
7	==(等于) !=(不等于)		从左到右
8	&(按位与)		从左到右
9	^(按位异或)		从左到右
10	|(按位或)		从左到右
11	&&(逻辑与)		从左到右
12	||(逻辑或)		从左到右
13	?:(条件运算符)	三目运算符	从右到左
14	=(赋值运算符) /=(除后赋值) *=(乘后赋值) %=(取余后赋值) +=(加后赋值) -=(减后赋值) >>=(右移后赋值) <<=(左移后赋值) &=(按位与后赋值) |=(按位或后赋值) ^=(按位异或后赋值)	复合运算符	从右到左
15	,(逗号运算符)	从左向右顺序运算	从左到右

符号

- signed：有符号，作为默认可以省略。
- unsigned：无符号，只能表示非负整数。

变量类型

- int：4字节
- short：2字节
- char：1字节
- long long：8字节
- long：4或者8字节

进制换算、位运算、溢出

```cpp
#include <iostream>
using namespace std;
int n, k;
int solve1()
{
    int l = 0, r = n;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (mid * mid <= n) l=mid+1;
        else r = mid - 1;
    }
    return l - 1;
}
double solve2(double x)
{
    if (x == 0) return x;
    for (int i = 0; i < k; i++)
        x = (x + n / x) / 2;
    return x;
}

int main()
{
    cin >> n >> k;
    double ans = solve2(solve1());
    cout << ans << ' ' << (ans * ans == n) << endl;
    return 0;
}
```

假设 int 为 32 位有符号整数类型，输入的 n 是不超过 47000 的自然数、k 是不超过 int 表示范围的自然数，完成下面的判断题和单选题：

- 判断题

28. 该算法最准确的时间复杂度分析结果为$O(\log n + k)$。（ ）

29. 当输入为“9801 1”时，输出的第一个数为“99”。（ ）

30. 对于任意输入的 n，随着所输入 k 的增大，输出的第二个数会变成“1”。（ ）

31. 该程序有存在缺陷。当输入的 n 过大时，第 12 行的乘法有可能溢出，因此应当将 mid 强制转换为 64 位整数再计算。（ ）

- 单选题

32. 当输入为“2 1”时，输出的第一个数最接近（ ）。

A. 1　　B. 1.414　　C. 1.5　　D. 2

33. 当输入为“3 10”时，输出的第一个数最接近（ ）。

A. 1.7　　B. 1.732　　C. 1.75　　D. 2

34. 当输入为“256 11”时，输出的第一个数（ ）。

A. 等于 16　　B. 接近但小于 16

C. 接近但大于 16　　D. 前三种情况都有可能

如何提升能力

听起来容易，实际还是挺难的。

所以相比于临时抱佛脚，扎实提升算法水平更重要。

- 学习各种算法。知道常见的写法。
- 学习更多的C++语言特性。
- 知道如何分析算法复杂度。
- 参考题解时，需要看懂别人在说什么。
  但不要照抄题解!

课间休息

完善程序题型

完善程序

题型介绍

- 给定一个问题描述，和一个有6个空缺部分的C++代码。
- 可能带有解法提示，也可能没有。
- 阅读这个代码，并且尝试理解程序在做什么。
- 完成选择题，从选项中选出最适合的以一个。

题数

- 两个大题，每个大题5个小题。
- 总分30分

例子

（1）（Josephus 问题）有 n 个人围成一个圈，依次标号 0 至 $n-1$。从 0 号开始，依次 $0,1,0,1,\ldots$ 交替报数，报到 1 的人会离开，直至圈中只剩下一个人。求最后剩下人的编号。试补全模拟程序。

```
01 #include <iostream>
03 using namespace std;
05 const int MAXN = 1000000;
06 int F[MAXN];
08 int main() {
09     int n;
10     cin >> n;
11     int i = 0, p = 0, c = 0;
12     while (①) {
13         if (F[i] == 0) {
14             if (②) {
15                 F[i] = 1;
16                 ③;
17             }
18             ④;
19         }
20         ⑤;
21     }
22     int ans = -1;
23     for (i = 0; i < n; i++)
24         if (F[i] == 0)
25             ans = i;
26     cout << ans << endl;
27     return 0;
28 }
```

例子

34. ①处应填（ ）

A. `i < n`　　B. `c < n`　　C. `i < n - 1`　　D. `c < n - 1`

35. ②处应填（ ）

A. `i % 2 == 0`　　B. `i % 2 == 1`　　C. `p`　　D. `!p`

36. ③处应填（ ）

A. `i++`　　B. `i = (i + 1) % n`

C. `c++`　　D. `p ^= 1`

37. ④处应填（ ）

A. `i++`　　B. `i = (i + 1) % n`

C. `c++`　　D. `p ^= 1`

38. ⑤处应填（ ）

A. `i++`　　B. `i = (i + 1) % n`

C. `c++`　　D. `p ^= 1`

出题点 – 流程控制

考察重点

进行循环、条件判断时，对条件或者控制流程的选择。

解题技巧

- 了解整个循环/条件判断的作用；
- 知道选项中不同的变量作用，和语句的功能；
- 猜测需要的条件；
- 特别注意选项之间的差异，尤其是边界问题。

出题点 – 状态变化

考察重点

对于一些模拟问题，或者动规，搜索，递推问题进行变量的状态转移。

解题技巧

- 了解附近代码的作用，需要求什么；
- 知道选项中不同的变量作用，和语句的功能；
- 根据上下文，决定变量或者状态的转移。

出题点 – 初始化

考察重点

对一个变量进行初始化。

或者在递推、动规中，进行搜索。

解题技巧

- 首先明确上下文的意思。
- 了解什么变量应当被初始化。
- 根据后文分析，应当被初始化为何值。

破题技巧 – 对称性

不同题目之间的对称性

- 题目可能会两段相似（多出现在搜索、DP中）
- 那么选项也可能会相似，但是注意变量的细节。

单独题目内的对称性

- 选项之间比较一下差异。
- 根据差异，得到可能需要注意的地方。
- 边界条件？哪些条件需要考虑，哪些不要？
- 看起来等价的写法，是否真的等价？
- 顺序是否有关？

破题技巧 – 变量使用

所有变量都是有用的。

- 变量在使用前，一定会进行初始化。
- 变量赋值计算后，之后一定会被使用。

技巧

- 空格后面出现了没出现过的变量，则是本空是初始化。
- 空格后面没出现选项涉及的变量，则可能被输出。
- 前面初始化或者赋值读入后，应当会被使用。
- 使用：赋值给其他变量；或者输出。

例题讲解

（1）（**Josephus 问题**）有 n 个人围成一个圈，依次标号 0 至 $n-1$。从 0 号开始，依次 $0,1,0,1,\ldots$ 交替报数，报到 1 的人会离开，直至圈中只剩下一个人。求最后剩下人的编号。试补全模拟程序。

```
01 #include <iostream>
03 using namespace std;
05 const int MAXN = 1000000;
06 int F[MAXN];
08 int main() {
09     int n;
10     cin >> n;
11     int i = 0, p = 0, c = 0;
12     while (①) {
13         if (F[i] == 0) {
14             if (②) {
15                 F[i] = 1;
16                 ③;
17             }
18             ④;
19         }
20         ⑤;
21     }
22     int ans = -1;
23     for (i = 0; i < n; i++)
24         if (F[i] == 0)
25             ans = i;
26     cout << ans << endl;
27     return 0;
28 }
```

34. ①处应填（ ）

A. `i < n`　　B. `c < n`

C. `i < n - 1`　　D. `c < n - 1`

35. ②处应填（ ）

A. `i % 2 == 0`　　B. `i % 2 == 1`

C. `p`　　D. `!p`

36. ③处应填（ ）

A. `i++`　　B. `i = (i + 1) % n`

C. `c++`　　D. `p ^= 1`

37. ④处应填（ ）

A. `i++`　　B. `i = (i + 1) % n`

C. `c++`　　D. `p ^= 1`

38. ⑤处应填（ ）

A. `i++`　　B. `i = (i + 1) % n`

C. `c++`　　D. `p ^= 1`

如何提升能力

本质上，和阅读程序的能力是相同的。

需要有一种“Hack”的意识。

遇到代码，需要思考细节是否正确。

实在不行，也可以猜，但是要往靠谱的方向猜。

没有什么立刻提升的办法，只能多学算法，多写题。

随堂测试

答案

FTFTBC

?FFTDB

CABDA

第7题是错题，意义不明，因此选什么都算对。