

昆明理工大学

课程报告

课程名称：____ 初级程序设计课程设计(0402702) _____

学 院：____ 信息工程与自动化学院 _____

专业年级：____ 2020 级 _____

学 号：____ 202010401137 _____

学生姓名：____ 丁紫嫣 _____

指导教师：____ 钱谦 _____

日 期：____ 2021-9-10 _____

一、 初级程序设计课程报告简介

本次程序设计课程任务是编写 100 个囚犯和灯泡问题和 21 点牌类单机游戏。由于之前没接触过 C++ 所以两个程序使用了 C 语言编写。

二、 课程报告目的

本次课程报告的目的是通过编写指定任务的程序找到学科漏洞和不足，在老师的演示下接触到了 Win32 窗口程序，学习编写更加复杂些的程序，而不仅仅对程序设计停留在十几行代码的低级接触状态，并对代码的简洁性要求有了一定认识。

三、 编程环境

本次程序设计使用了 Microsoft Visual C++ 2010 编写。

四、 报告内容

1. 利用结构化程序设计思想，完成一个解“100 个囚犯和灯泡”问题的程序。

问题描述如下，有 100 个囚犯分别关在 100 间牢房里。牢房外有一个空荡荡的房间，房间里有一个由开关控制的灯泡。初始时，灯是关着的（或者随机）。看守每天随便选择一名囚犯进入房间，但保证每个囚犯都会被选中无穷多次。如果在某一时刻，有囚犯成功断定出所有人都进过这个房间了，所有囚犯都能释放。游戏开始前，所有囚犯可以聚在一起商量对策，但在此之后它们唯一可用来交流的工具就只有那个灯泡。他们应该设计一个怎样的协议呢？请根据此协议设计实现一个程序，把解题过程详细输出并给出解题所需天数。

2. 利用结构化程序设计思想，完成一个 21 点人机对战牌类游戏程序。游戏过程如下，进入程序后玩家根据提示信息选择开始游戏，此时系统对一副牌（52 张）进行洗牌，然后分别给玩家和电脑发一张牌。之后玩家和电脑在每回合中自行选择每次发牌时要牌还是不要牌，双方都力争使自己接近 21 点且不“爆牌”，即不超过 21 点。玩家和电脑在某回合中一旦选择不要牌，则以后都不可以再要牌，只能等待游戏结束（即对方不再要牌或者对方超过 21 点）。如果玩家要牌后超过 21 点，则电脑获胜；如果玩家不要牌，电脑要牌后超过 21 点，则玩家获胜；如果双方都不超 21 点并都不要牌，则点数接近 21 点的获胜，点数一样则为平局。

以上为本次初级程序设计需要解决的两道题目，我根据以上题目对代码进行编写，以下为思路说明。

针对第一题，大致思路已在 PPT 中的流程图中给出，即约定一位计数人来进行计数和关灯操作，（比如 9 月 1 日出来的人被公认为计数人，计数人 id 号可以随意，定下来即可），其余囚犯负责开灯，以开灯作为信号表明“我出来过”。计数人通过观察自己已经关掉的灯被其他囚犯打开了来确定在自己第一次与第二次放风期间有囚犯出来过并进行计数，但是必须与囚犯们在一开始就说好每人无论出来放风多少次都只能在第一次出来的时候开灯一次，其余出来的时候要维持灯的关闭状态。当技术者

计数了 99 次，可以断定所有囚犯都出来放过风。

如果灯初始随机，且囚犯不知道年月日，如果还按照上述规则：就只能指定一位号码的囚犯当计数人，而不能通过规定哪天出来的囚犯当计数人，即起始计数的时刻不能确定，因而也无法确认计数人放风之前有没有囚犯出来过（比如 1 号囚犯当计数人），计数人无论出来的时候灯是什么状态都关掉，从他出来的那一刻进行日期计算，前面出来的人不算，而后出来的囚犯不会关掉其他囚犯开灯传达的信号。那么由于共同协商只能开灯一次，比如在计数人之前 2 号已经开了灯，计数人出来关掉了灯，就不知道这灯开着算初始状态还是 2 号传达的“我出来过”的消息，（2 号以后 1 号以前可能出来了很多 3、4、5 号等人，幸运的是，他们看到 2 号开着的灯不必操作，只需等 1 号计数人关掉灯后正常传达自己的信号即可）那么当计数人数到 98，由于 2 号已经开过一次灯便不会再传递消息，计数人也就永远无法数到 99，也不能确认 2 号是出来过还是从未被放出来过导致自己只能数到 98。

为解决这个问题，如果按照 PPT 中所写：每位囚犯可以开灯两次，我们依旧以开灯为信号表示“我已经出来过了”，计数人计数到 198 时可确定，如果还有人没出来过，那么计数不会超过 197（即 98 人的 196 次+初始灯开着 1），此种方法通过“每位囚犯开灯两次”来确保计数人出来前灯的状态影响到计数判断的结果，就算 2 号在 1 号计数人出来之前开灯过一次，在剩下 98 人开了两次，计数人 1 号计数了 196 次后，计数人可以等到 2 号开 197 次灯让自己关掉，来保证所有囚犯都出来过了，而不用等是否有第 198 次灯让自己关。

针对第二题，由于之前没有学过 C++，不清楚类的使用，该代码参考了一些 C 语言源代码，使用函数引用结构体来设计玩家模型，使用了结构体指针和函数对结构体的引用。在之前的学习中我对结构体的引用还不甚明晰，经过 21 点游戏的改动也习惯了函数和结构体嵌套的使用模式。本代码中对于花色的控制和平局的控制是编写过程中的两大难点。

五、关键代码

100 囚犯与灯代码如下，参考附件 prisoner.cpp

```
1.  #include <stdio>
2.  #include <stdlib.h>
3.  #include <time.h>
4.
5.  int counter,sum,day,sum_average = 0;
6.  int v[100]={0},light;
7.  int frequency[100]={0};
8.
9.  void oper(int id)
10. {
11.     if (counter ==1)    //随机确定一个计数人
12.     {
13.         counter = id;
14.         light =0;
15.         printf("%d 号犯人进行计数\n",id);
```

```

16.     return ;
17. }
18.
19. if (id==counter&&light==1)    //计数人进行关灯操作 ,确定已经有人出来
20. {
21.     light=0;
22.     sum++;
23.     printf("目前除去计数人至少已有%d 人进行放风\n",sum);
24.     frequency[id]++;//如果计数人发现灯关着,则不用进行任何操作
25. }
26. if (id!=counter&&!light&&v[id])    // 非计数人,灯是关着的, 犯人没进行过开灯操作
27. {
28.     v[id] = 1;    //确定该犯人进行开灯灯
29.     light = 1;
30.     printf("%d 号犯人进行放风并开灯\n",id);
31.     frequency[id]++;
32. }
33. }
34.
35. int check(int sum)
36. {
37.     if (sum==99) return 1;
38.     else return 0;
39. }
40.
41. int main()
42. {
43.     srand(time(NULL));    //播随机种子
44.     light =rand()%1;    //产生 0,1 随机数, 0 代表灯关, 1 代表灯开
45.     counter = 1;
46.     day = 0;
47.     sum = 0;
48.     while (true)
49.     {
50.         int id = rand() % 100;    //随机犯人
51.         oper(id);
52.         if (check(sum))    //查看是否全部出来放过风
53.         {
54.             day++;
55.             frequency[id]++;
56.             for(int i=0; i<100;i++){
57.                 printf("%d 号犯人放风%d 次\n",i+1,frequency[i]);

```

```

58.         sum_average +=frequency[i];
59.     }
60.     printf("每个犯人平均放风%d 次\n",sum_average/100);
61.     printf("经过%d 天, 犯人解放\n",day);
62.     break;
63. }
64.     day++;
65.     frequency[id]++;
66. }
67.     system("pause");
68.     return 0;
69. }

```

当犯人们不知道年月日和灯初状态始随机的情况如下，参考附件 az.cpp

```

1.     #include <cstdio>
2.     #include <stdlib.h>
3.     #include <time.h>
4.
5.     int counter,sum,day;
6.     int v[100]={0},light;
7.     int frequency[100]={0};
8.
9.     void oper(int id)
10. {
11.     if (counter ==1)    //随机确定一个计数人，假设囚犯们在商讨时抽签随机决定了这位计数
        人号码，此程序计算的天数是计数人出来的那一天开始算得的天数，在计数人出来以前可能已经过了无
        穷多天
12.     {
13.         counter = id;
14.         light =0;
15.         printf("%d 号犯人进行计数\n",id);
16.         return ;
17.     }
18.     if (id==counter&&light)    //计数人进行关灯操作
19.     {
20.         light=0;
21.         sum++;
22.         printf("目前已关灯%d 次\n",sum);
23.         frequency[id]++;
24.     }
25.     if (id!=counter&&!light&&v[id]<2)    // 非计数人，灯是关着的， 犯人尚未开满两次灯

```

```

26.     {
27.         v[id]++;    //确定该犯人进行开灯,记录该犯人开灯次数
28.         light = 1;
29.         printf("%d 号犯人进行放风并开灯\n",id);
30.         frequency[id]++;
31.     }
32. }
33.
34. int check(int sum)
35. {
36.     if (sum==197) return 1;
37.     else return 0;
38. }
39.
40. int main()
41. {
42.     srand(time(NULL));    //播随机种子
43.     light = 1;    //产生 0,1 随机数, 0 代表灯关, 1 代表灯开
44.     counter = 1;
45.     day = 0;
46.     sum = 0;
47.     while (true)
48.     {
49.         int id = rand() % 100;    //随机犯人
50.         oper(id);
51.         if (check(sum))    //查看是否全部出来放过风, 计数人关了 197 次必定全部放过风了
52.         {
53.             day++;
54.             frequency[id]++;
55.             for(int i=0; i<100;i++){
56.                 printf("%d 号犯人放风%d 次\n",i+1,frequency[i]);
57.             }
58.         }
59.
60.         printf("经过%d 天, 犯人解放\n",day);
61.         break;
62.     }
63.     day++;
64.     frequency[id]++;
65. }
66. system("pause");
67. return 0;

```

```
68. }
```

21 点游戏代码 参考附件 21game.cpp

```
1.  #include<stdio.h>
2.  #include<stdlib.h>
3.  #include<time.h>
4.  #include<string.h>
5.  #include<ctype.h>
6.  #define NUMBER 13
7.  #define SIZE 1000
8.  #define length 100
9.  //玩家模型
10. struct Player{
11.     char name[length];
12.     int points;
13.     int refuse;
14. };
15. int Shuffle_cards(int S_cards[]);
16. int play_game();
17. int Reset_game(struct Player* players[],int bots);
18. struct Player *Create_bot(char name[]);
19. int IS_card(int Is_card[],int h);
20. //重置游戏,角色不释放
21. int Reset_game(struct Player* players[],int bots){
22.     int i;
23.     for(i=0;i<bots;i++){
24.         players[i]->points=0;
25.         players[i]->refuse=1;
26.     }
27.     return 1;
28. }
29. int Read_line(char str[],int n);
30. //字符串输入函数
31. int Read_line(char str[],int n){
32.     int ch,i=0;
33.     while(isspace(ch=getchar()))
34.         ;
35.     while(ch!='\n'&&ch!=EOF){
36.         if(i<n)
37.             str[i++]=ch;
38.         ch=getchar();
```

```

39.     }
40.     str[i]='\0';
41.     return i;
42. }
43. //当前牌在牌库中是否发完 , S_cards 数组是牌库
44. int IS_card(int Is_card[],int h){
45.     if(Is_card[h])
46.         return 1;
47.     else
48.         return 0;
49. }
50. //随机发牌
51. int Shuffle_cards(int S_cards[]){
52.     int a,i,t,h;
53.     while(1){
54.         a=rand()%NUMBER+1;//随机点数 1-13
55.         t=rand()%3;//随机数 0-3 控制花色
56.         h=t*13+a;
57.         if(IS_card(S_cards,h)){
58.             for(i=0;i<52;i++){
59.                 if(h==S_cards[i]){
60.                     S_cards--;
61.                     break;
62.                 }
63.             }
64.             break;
65.         }else{break;}
66.     }
67.     return a;
68. }
69. //创建角色
70. struct Player *Create_bot(void){
71.     struct Player *p;
72.     p=(Player*)malloc(sizeof(struct Player));
73.     if(p==NULL){
74.         printf("创建机器人失败!\n");
75.     }
76.     Read_line(p->name,length);
77.     p->points=0;
78.     p->refuse=1;
79.     return p;
80. }
81. int play_game(){

```



```

82.     int i,bots_num,cards[NUMBER]={4},card,choose,max,maxid,players[4]={0},drawCount=0;
83.     struct Player *plays[5];
84.     printf("欢迎来到 21 点游戏!\n");
85.     printf("下面我将告诉你游戏规则:\n");
86.     printf("1.你要选择不超过 3 个机器人做你的对手\n");
87.     printf("2.你将在每轮选择是否要牌.如果某轮你要牌让你获得所有牌的总点数超过 21, 你将输掉本局.最终, 你将得知那个获得最大点数的赢家的名字.\n");
88.     printf("请输入本局你的对手数目 (不超过 3 个):");
89.     scanf("%d",&bots_num);
90.     printf("请输入你的角色名:");
91.     plays[0]=Create_bot();
92.     i=1;
93.     while(bots_num--){
94.         printf("请输入创建的机器人角色名:");
95.         plays[i++]=Create_bot();
96.     }
97.     bots_num=i;
98.     while(1){
99.         if(plays[0]->refuse){
100.             card=Shuffle_cards(cards);
101.
102.             printf("%s , 你拿走的牌点数是: %d\n",plays[0]->name,card);
103.             if(card==1&&plays[0]->points<=10)
104.                 card=11;
105.             if(card==1&&plays[0]->points>10)
106.                 card=1;//card==1 代表抽取了 A(可能是四个花色之一), A 的值是变化的, 根据是
                是否会爆牌而变化值
107.             plays[0]->points+=card;
108.             printf("你现在牌的总点数是:%d\n",plays[0]->points);
109.             if(plays[0]->points>21){
110.                 printf("对不起, 你所有的牌的点数超过了 21, 输掉了本局!^M^\n");
111.                 printf("%s , 你是否还要开启下一局? (Yes/1 or No/0)",plays[0]->name);
112.                 scanf("%d",&choose);
113.                 if(choose){
114.                     Reset_game(plays,bots_num);
115.                     continue;
116.                 }else{
117.                     return 0;
118.                 }
119.             }
120.
121.         }else{

```

```

122.         max=plays[0]->points;
123.         //寻找胜利者
124.         for(i=0;i<bots_num;i++){
125.             if(max<=plays[i]->points){
126.                 max=plays[i]->points;
127.                 maxid=i;
128.             }
129.         }
130.         //寻找平局的可能
131.         for(i=0,drawCount=0;i<bots_num;i++){
132.             if(plays[i]->points==max){
133.                 players[i]=1;
134.                 drawCount++;
135.             }
136.         }
137.         printf("你最终的点数是:%d\n",plays[0]->points);
138.         for(i=1;i<bots_num;i++){
139.             printf("机器人 %s 最终的点数
是:%d\n",plays[i]->name,plays[i]->points);
140.         }
141.         if((maxid==0)&&(drawCount==1)){//玩家获胜
142.             printf("恭喜你赢得了本局!^v^\\n");
143.             printf(" %s , 你是否还要开启下一局?(Yes/1 or No/0)",plays[0]->name);
144.             scanf("%d",&choose);
145.             if(choose){
146.                 Reset_game(plays,bots_num);
147.                 continue;
148.             }else{
149.                 return 1;
150.             }
151.         }
152.         if((maxid==0)&&(drawCount>1)){//玩家和电脑平局
153.             for(i=1;i<bots_num;i++){
154.                 if(players[i]){
155.                     printf("机器人%s 与您获得了平局 ",plays[i]->name);
156.                 }
157.             }
158.             continue; }
159.             printf(" %s , 你是否还要开启下一局?(Yes/1 or No/0)",plays[0]->name);
160.             scanf("%d",&choose);
161.             if(choose){
162.                 Reset_game(plays,bots_num);
163.                 continue;

```

```

164.         }else{
165.             return 2;
166.         }
167.     }
168.
169.     else{// 电脑获胜
170.         printf("%s 赢得了胜利!^m^\\n",plays[maxid]->name);
171.         printf("%s, 你是否还要开启下一局?(Yes/1 or No/0)",plays[0]->name);
172.         scanf("%d",&choose);
173.         if(choose){
174.             Reset_game(plays,bots_num);
175.             continue;
176.         }else{
177.             return 0;
178.         }
179.     }
180.
181. }
182.
183.
184.
185. if(plays[0]->refuse){
186.     for(i=1;i<bots_num;i++){
187.         if(plays[i]->refuse){
188.             if(plays[i]->points>=10){
189.                 card=Shuffle_cards(cards);
190.                 while(1){
191.                     if(card>3) card=Shuffle_cards(cards);
192.                     else break;//此处可见机器人的出牌选择并不很随机,反而很努力地出牌
求胜
193.                 }
194.             }
195.             plays[i]->points+=card;
196.             if(plays[i]->points>=19) plays[i]->refuse=1;
197.         }else{
198.             continue;
199.         }
200.     }
201.     printf("你是否要继续要牌?(Yes/1 or No/0)");
202.     scanf("%d",&choose);
203.     plays[0]->refuse=choose;
204. }
205. }

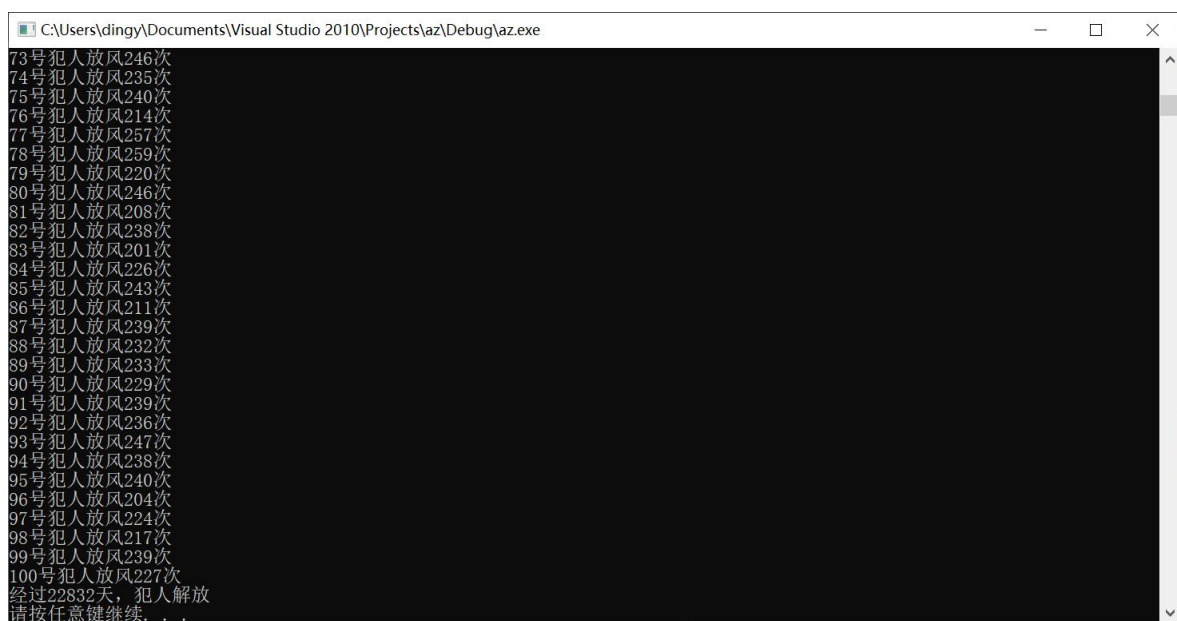
```

```

206. }
207. int main(){
208.     srand(time(NULL));
209.     play_game();
210.     return 0;
211. }

```

六、实验结果截图（关键界面）



```
C:\Users\dingy\documents\visual studio 2010\Projects\21点\Debug\21点.exe
请输入本局你的对手数目（不超过3个）:3
请输入你的角色名:www
请输入创建的机器人角色名:eee
请输入创建的机器人角色名:rrr
请输入创建的机器人角色名:ttt
www，你拿走的牌点数是: 9
你现在牌的总点数是:9
你是否要继续要牌?(Yes/1 or No/0)1
www，你拿走的牌点数是: 2
你现在牌的总点数是:11
你是否要继续要牌?(Yes/1 or No/0)1
www，你拿走的牌点数是: 8
你现在牌的总点数是:19
你是否要继续要牌?(Yes/1 or No/0)0
你最终的点数是:19
机器人 eee 最终的点数是:12
机器人 rrr 最终的点数是:12
机器人 ttt 最终的点数是:13
恭喜你赢得了本局! v
www，你是否还要开启下一局?(Yes/1 or No/0)1
www，你拿走的牌点数是: 4
你现在牌的总点数是:4
你是否要继续要牌?(Yes/1 or No/0)1
www，你拿走的牌点数是: 7
你现在牌的总点数是:11
你是否要继续要牌?(Yes/1 or No/0)1
www，你拿走的牌点数是: 12
你现在牌的总点数是:23
对不起，你所有的牌的点数超过了21，输掉了本局!`M`
www，你是否还要开启下一局?(Yes/1 or No/0)1
```

```
文件 C:\Users\dingy\documents\visual studio 2010\Projects\21点\Debug\21点.exe
欢迎来到21点游戏!
下面我将告诉你游戏规则:
1. 你要选择不超过3个机器人做你的对手
2. 你将在每轮选择是否要牌. 如果某轮你要牌让你获得所有牌的总点数超过21, 你将输掉本局. 最终, 你将得知那个获得最大点数的赢家的名字.
请输入本局你的对手数目（不超过3个）:3
请输入你的角色名:qqq
请输入创建的机器人角色名:www
请输入创建的机器人角色名:eee
请输入创建的机器人角色名:rrr
qqq，你拿走的牌点数是: 3
你现在牌的总点数是:3
你是否要继续要牌?(Yes/1 or No/0)1
qqq，你拿走的牌点数是: 11
你现在牌的总点数是:14
你是否要继续要牌?(Yes/1 or No/0)1
qqq，你拿走的牌点数是: 11
你现在牌的总点数是:25
对不起，你所有的牌的点数超过了21，输掉了本局!`M`
qqq，你是否还要开启下一局?(Yes/1 or No/0)1
qqq，你拿走的牌点数是: 13
你现在牌的总点数是:13
你是否要继续要牌?(Yes/1 or No/0)1
qqq，你拿走的牌点数是: 10
你现在牌的总点数是:23
对不起，你所有的牌的点数超过了21，输掉了本局!`M`
qqq，你是否还要开启下一局?(Yes/1 or No/0)1
qqq，你拿走的牌点数是: 12
你现在牌的总点数是:12
你是否要继续要牌?(Yes/1 or No/0)1
```

七、报告总结体会

经过老师的讲解,我大概明白 C++中的面向对象的含义并非抽象的而是针对生活中某一物的状态而封装的,比如 21 点中的扑克牌就可以作为对象,根据扑克的规则将函数对牌进行封装,在其他地方也可拿来使用。作为自动化转专业到计科来的新同学来说,没有学过 C++是一块短板,对面向对象和类在自学的基础上书上的概念也是模糊的,说不清怎么运用,但是经过这门初级程序设计课程设计,我看到了对类的使用的演示,也第一次见到窗口程序,发现它的源代码和外观表现与控制台程序的不同。而囚徒与灯的问题也第一次让我初步接触到算法,并在询问老师并自己思考后有种顿悟的感觉,对此引发了兴趣。在这种引导下,我乐于搞清楚这门课所涉及的我从未接触到的以上方面,但由于课程时长较短,也只能课下再钻研琢磨。

我对没精通一门语言不能学好其他语言深表认同,并认识到 C 与 C++也有很多相似之处,虽然 C++涵盖内容更广但要说“精通 C”应该是精通二者,尤其是 C++。假期也粗略看过短学期结束后要学习的 Java 相关书籍,明白 C++跟 C 的不同之处,

也明白这不同之处在于 Java 等编程语言恰好又有相同之处。但由于课程时间冲突不能在之后对大一落下的 C&C++ 程序设计重新学习，只能在课下自我补习。通过这次课程的练习，我明白代码量的短板很严重，作为计算机专业的学生更应该勤于练习才能更快理解和运用代码，哪怕不全是自己敲出来的代码，使用参考前人“造的轮子”，将别人的轮子改成我所需要的规格的轮子，debug 时的修改效率也会提高。未来我将从 Leetcode、Github 等网站查找资料练习提升自己，而不是仅羡慕于那些码龄大佬能编写出怎样的小游戏程序，尽快在大三以前达到三到四千行代码量的练习。

八、参考文献

- [1] 方娇莉，潘晟旻，《c 语言程序设计（慕课版）》，北京：电子工业出版社，2018.1。
- [2] 陈春丽 等，《程序设计基础及应用（C&C++语言）》，北京：清华大学出版社，2020.2。