

模拟 C++ 语言试卷 4

一、单项选择题（共 15 题，每题 2 分，共计 30 分，每题有且仅有一个正确选项）

- 下列哪个选项（ ）可以正确用来表示八进制数。
A、0x18 B、(0800)₈ C、123O D、(001)₂
- 计算机各个部件传输信息的公共通路被称为总线，一次传输信息的位数称为总线的（ ）。
A、长度 B、宽度 C、粒度 D、深度
- 先于或随着操作系统的系统文件装入内存储器，从而获得计算机特定控制权并进行串扰和破坏的病毒是（ ）。
A、文件型病毒
B、引导区型病毒
C、宏病毒
D、网络病毒
- 在平衡二叉树中，节点的平衡因子的取值不可能是（ ）。
A、1 B、2 C、-1 D、0
- 六个人排成一排，甲乙不相邻，且甲丙不相邻的不同排法数是多少种（ ）。
A. 216
B. 240
C. 288
D. 360
- 已有一个栈，进栈顺序为 ABCD，出栈顺序为 BCDA，若用 I 表示进栈，O 表示出栈，则进操作的顺序是（ ）。
A. IOIOIIIO
B. IIOOIIIO
C. IIOIOIOO
D. IIOIOOIO
- 十六进制数 FFF.CH 相当于十进制数（ ）。
A. 4096.3
B. 4096.25
C. 4095.75
D. 4096.75
- 面向对象程序设计方法的优点不包括（ ）。
A. 可重用性 B. 可扩展性 C. 易于管理和维护 D. 简单易懂

9. 在一个图中，所有顶点的度数之和等于图的边数的（ ）倍。
A. 2 B. 1 C. 1/2 D. 4
10. 线性表(a1,a2,...,an)以链接方式存储时，访问第 i 位置元素的时间复杂性为（ ）。
A. O(i)
B. O(1)
C. O(n)
D. O(i-1)
11. 4 个袋子，15 个球，每个袋子至少放一个球，而且袋子中的球的数量不能重复，问有多少种方式（ ）。
A. 4
B. 6
C. 7
D. 5
12. $3*(5-2)+7$ 的后缀表达式为（ ）。
A. $352-*7+$
B. $532-*7+$
C. $352*-7+$
D. $352-*-7$
13. 下列标识符错误的事（ ）。
A. xly
B. _123
C. 2ab
D. _ab
14. 若是一棵完全二叉树有 768 个节点，则该二叉树中叶节点的个数为（ ）。
A. 257 B. 258 C. 384 D. 385
15. 对于一个具有 n 个顶点的无向图，若是采用邻接表表示，则存放表头节点的数组大小为（ ）。
A. n
B. n+1
C. n-1
D. n+边数

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.

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int n,q,i,j,s=1,maxx;
4  int main()
5  {
6      cin>>n>>q;
7      for(i=2;i<=n;i++)
8      {
9          cin>>j;
10         if(q<j&&j-q==1) s++;
11         else s=1;
12         if(s>maxx) maxx=s;
13         q=j;
14     }
15     cout<<maxx;
16     return 0;
17 }
```

判断题:

1. 本程序时间复杂度为 $O(n)$ 。()
2. maxx 的值必然不可能大于 n 。()
3. maxx 的初始值没有给出。()
4. 12 行 $s > maxx$ 修改为 $s \geq maxx$ 会导致程序结果受到影响。()

选择题:

5. 若输入

10

3 5 6 2 3 4 5 6 8 9

程序输出结果为 ()。

A. 1 B. 6 C. 7 D. 5

6. 若输一串质数, 理论上最大的输出结果为 ()

A. 1 B. 2 C. 0 D. 最大值不确定

2.

```

1  #include<cstdio>
2  #include<queue>
3  using namespace std;
4
5  queue<char>a;
```

```

6  int left,right,t;
7
8  void read(){
9      char ch=getchar();
10     while(ch!='@') a.push(ch),ch=getchar();
11 }
12
13 int main(){
14     read();
15     while(!a.empty()){
16         if(a.front()=='('){
17             ++left;
18             ++t;
19         }
20         if(a.front()==')'){
21             ++right;
22             if(t)--t;
23         }
24         a.pop();
25     }
26     if(left==right&& t==0) printf("YES");
27     else printf("NO");
28     return 0;
29 }

```

判断题:

1. 程序输入直到读到'@'为止。()
2. t 代表读入的左括号数量。()
3. 26 行比较条件仅仅 $left == right$ 程序输出结果不变。()
4. 输入 $((()))) (($, 26 行做判定时 $left$ 和 $right$ 不相等。()

选择题:

5. 若输入

$2*(x+y)/(1-x)@$

输出结果是()。

A. YES B. NO C. YES! D. NO!

6. 若输入

$(25+x)*(a*(a+b+b))@$

输出结果时 ()

A. YES B. NO C. YES! D. NO!

3.

```
1  #include<cstdio>
2  #include<iostream>
3  #include<cstring>
4  using namespace std;
5  void befond(string in,string after){
6      if (in.size()>0){
7          char ch=after[after.size()-1];
8          cout<<ch;
9          int k=in.find(ch);
10         befond(in.substr(0,k),after.substr(0,k));
11         befond(in.substr(k+1),after.substr(k,in.size()-k-1));
12     }
13 }
14 int main(){
15     string inord,aftord;
16     cin>>inord;
17     cin>>aftord;
18     befond(inord,aftord);cout<<endl;
19     return 0;
20 }
```

判断题:

1. 程序蕴藏了递归的思想。()
2. cout<<ch 一共要运行 inord.size()次。()

选择题:

3. 输入

BADC

BDCA

输出是 ()。

- A. ABCD
- B. ABDC
- C. ACBD
- D. ACDB

4. 输入

BADC

BDCA

会执行几次第 10 行的 befond 函数 () 次。

- A. 4
- B. 3
- C. 2
- D. 1

5. 长度为 n 的序列，最差的情况，会迭代多少次 beford 函数 ()

- A. n
- B. $n \log n$
- C. $n+1$
- D. $n/2$

6. 输入

342165897

432698751

输出是 ()。

- A. 123456789
- B. 124357689
- C. 123456879
- D. 156789234

三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

1. (阶乘之和) 用高精度计算出

$S=1!+2!+3!+4!+\cdots+n! (n \leq 50)$

其中! 代表阶乘, 例如 $5!=5*4*3*2*1$

输入:

正整数 N 。

输出:

一个正整数 S , 表示计算结果。

样例:

输入:

3

输出

9

```
1  #include<iostream>
2  #include<cstring>
3  using namespace std;
4  int n,a[90],b[90],c[90],f[90],d=0,len_a,len_b=1,len_c=1,len_ans,m=1;
5  string s;
6  int main(){
7      cin>>n;
8      b[0]=1;
9      for(int i=1;i<=n;i++){
10         len_a=0;
11         int p=i;
12         while(p>0){
13             a[len_a++]=p%10;
```

```

14         ---1---;
15     }
16     for(int j=0;j<len_a;j++)
17         for(int k=0;k<=len_b;k++)
18             c[---2---]+=a[j]*b[k];
19     for(int j=0;j<len_c;j++)
20         if(c[j]>9) c[j+1]+=c[j]/10,c[j]%10;
21     if(c[len_c]) len_c++;
22     len_ans=len_b,len_b=len_c,m=max(m,len_c);
23     for(int k=len_c-1;k>=0;k--) ---3---;
24     len_c=len_a+len_ans;
25     memset(c,0,sizeof(c));
26     for(int j=0;j<m;j++){
27         ---4---;
28         if(f[j]>9) f[j+1]+=f[j]/10,f[j]%10;
29     }
30 }
31 while(!f[m]&& m>0) ---5---;
32 for(int i=m;i>=0;i--) cout<<f[i];
33 return 0;
34 }

```

1. (1) 处应填 ()。

A. p/=10 B. p-- C. p-=10 D. p++
2. (2) 处应填 ()。

A. j+k-1 B. j+k C. k D. k+1
3. (3) 处应填 ()。

A. c[k]=b[k] B. c[k]-- C. a[k]=c[k] D. b[k]=c[k]
4. (4) 处应填 ()。

A. f[j]+=b[j]+a[j] B. f[j]+=a[j] C. f[j]+=c[j] D. f[j]+=b[j]
5. (5) 处应填 ()。

A. m-=2 B. m-- C. f[m]-- D. f[m--]++

2. (乒乓球) 国际乒联现在主席沙拉拉自从上任以来就立志于推行一系列改革，以推动乒乓球运动在全球的普及。其中 11 分制改革引起了很大的争议，有一部分球员因为无法适应新规则只能选择退役。华华就是其中一位，他退役之后走上了乒乓球研究工作，意图弄明白 11 分制和 21 分制对选手的不同影响。在开展他的研究之前，他首先需要对他多年比赛的统计数据进行一些分析，所以需要你的帮忙。

华华通过以下方式进行分析，首先将比赛每个球的胜负列成一张表，然后分别计算在 11 分制和 21 分制下，双方的比赛结果（截至记录末尾）。

比如现在有这么一份记录，（其中 W 表示华华获得一分，L 表示华华对手获得一分）：

WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWLW

在 11 分制下，此时比赛的结果是华华第一局 11 比 0 获胜，第二局 11 比 0 获胜，正在进行第三局，当前比分 1 比 1。而在 21 分制下，此时比赛结果是华华第一局 21 比 0 获胜，正在进行第二局，比分

2 比 1。如果一局比赛刚开始，则此时比分为 0 比 0。直到分差大于或者等于 2，才一局结束。

你的程序就是要对于一系列比赛信息的输入（WL 形式），输出正确的结果。

输入格式

每个输入文件包含若干行字符串，字符串有大写的 W、L 和 E 组成。其中 E 表示比赛信息结束，程序应该忽略 E 之后的所有内容。

输出格式

输出由两部分组成，每部分有若干行，每一行对应一局比赛的比分（按比赛信息输入顺序）。其中第一部分是 11 分制下的结果，第二部分是 21 分制下的结果，两部分之间由一个空行分隔。

样例

输入：

WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW

WWLWE

输出：

11:0

11:0

1:1

21:0

2:1

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int win[62503];
5  int w,l;
6  int main()
7  {
8      char s;
9      for(int i=1;cin>>s&&---1---;i++)
10     {
11         if(s=='W') win[i]=1;
12         else win[i]=2;
13     }
```



```

14     for(int i=1;1;i++)
15     {
16         if(win[i]==1) ---2---;
17         if(win[i]==2) l++;
18         if(win[i]==0)
19         {
20             cout<<w<<": "<<l<<endl<<endl;
21             ---3---;
22         }
23         if(w-l>=2||l-w>=2)
24             if(---4---)
25             {
26                 cout<<w<<": "<<l<<endl;
27                 w=0;
28                 l=0;
29             }
30     }
31     w=0;
32     l=0;
33     for(int i=1;1;i++)
34     {
35         if(win[i]==1)w++;
36         if(win[i]==2)l++;
37         if(win[i]==0)
38         {
39             cout<<w<<": "<<l;
40             break;
41         }
42         if(---5---)
43             if(w>=21||l>=21)
44             {
45                 cout<<w<<": "<<l<<endl;
46                 w=0;
47                 l=0;
48             }
49     }
50     return 0;
51 }

```

1. (1) 处应填 ()。

A. s!='E' B. s!= 'W' C. s!= 'L' D. s!= ''

2. (2) 处应填 ()。

A. l-- B. w++

C. $w+=2$ D. $l-=2$

3. (3) 处应填 ()。

A. continue B. break C. return 0 D. cout<<endl

4. (4) 处应填 ()。

A. $w \geq 11$ B. $l \geq 11$
C. $w \geq 11 || l \geq 11$ D. $w \geq 11 \&\& l \geq 11$

5. (5) 处应填 ()。

A. $w-l \geq 2 \&\& l-w \geq 2$ B. $w-l \geq 2 || l-w \geq 2$
C. $w-l \geq 1 \&\& l-w \geq 1$ D. $w-l \geq 1 || l-w \geq 1$



太戈编程
Tiger AI