

入门级 C++ 语言试卷 3（模拟）

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）

1. 计算机网络最大的优点是（ ）。
A、精度高 B、资源共享 C、运行速度快 D、存储容量大
2. 计算机病毒是指（ ）。
A、编制有错误的计算机程序 B、设计不完善的计算机程序 C、计算机的程序已被破坏
D、以危害系统为目的的特殊的计算机程序
3. 下列各数中最大的是（ ）。
A、11010110.0101(二进制) B、D6.53(十六进制) C、214.32(十进制)
D、326.25(八进制)
4. 已知英文字母 a 的 ASCII 代码值是十六进制数 61H, 那么字母 z 的 ASCII 代码值十六进制是（ ）。
A、7BH B、73H C、7AH D、64H
5. 以下哪一个不是视频文件格式（ ）。
A. .rmvb
B. .xv
C. .bmp
D. .mov
6. 如果一棵二叉树的中序遍历是 BCAGEDF, 后序遍历是 CBGEFDA, 那么他的先序遍历是（ ）。
A. ABCDEFG
B. ABCDEGF
C. ACBDEFG
D. ACBDEGF
7. 一个无向图的顶点个数为 x 个, 那么以下选项中哪一个必然不是其边数（ ）。
A. 0 B. $x/2$ C. $x(x-1)/2$ D. x^2
8. 前缀表达式 $- * 5 8 + 4 / 8 2$ 的值是（ ）。
A. 32 B. 30 C. 48 D. 50

9. 将根节点深度记为 1，那么深度为 n 的完全二叉树至少有几个节点 ()。
- A. $2^n - 1$ B. $2^{(n-1)} + 1$ C. $2^{(n-1)}$ D. $2 * n$
10. 在 $16 * 16$ 点阵的字库中调用文字，句子“明天你好”占用的字节数是 ()。
- A. 128 B. 144 C. 784 D. 196
11. 7 个人并排站成一行，如果甲乙两人必须不相邻，那么不同的排法总数是 ()。
- A. 1440
B. 3600
C. 4820
D. 4800
12. 入栈顺序是: $a_1, a_3, a_5, a_2, a_4, a_6$ ，出栈顺序是: $a_5, a_4, a_2, a_6, a_3, a_1$ ，则栈的容量最小是 ()。
- A. 5 B. 3 C. 6 D. 4
13. 下面说法错误的是 ()。
- A. ASCII 码的含义是美国信息交换标准代码
B. $1\text{GB} = 1024\text{MB}$
C. 程序和数据在内存中都是用二进制码表示的
D. 在计算机中，字符编码通常采用补码
14. 在一个顺序表中 $(1, 4, 7, 9, 12, 23, 45, 78, 99, 123, 456)$ 中，用二分法查找关键码 7，需要关键码比较次数为 ()。
- A. 2 B. 3 C. 4 D. 5
15. 当网络中的任何一个工作站发生故障时，都有可能导致整个网络停止工作，这种网络的拓扑结构为 ()。
- A. 星型
B. 环型
C. 总线型
D. 树型

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 <iostream>
2	using namespace std;
3	int n,i,ans;
4	int main()
5	{

```

6      cin>>n;
7      ans=0;
8      for(i=1;i<=n;i++)
9          if(n%i==0) ans++;
10     cout<<ans<<endl;
11     return 0;
12 }
```

判断题：

1. 本程序时间复杂度为 $O(n)$ 。()
2. 程序实现的是求 n 的因数和。()
3. 对于奇数 n ，第 8 行 $i++$ 修改成 $i+=2$ 不会影响结果。()
4. 将 8 行 $i<=n$ 修改成 $i*i<=n$ ，然后第 10 行输出 $ans*2$ ，程序结果一样。()

选择题：

5. 若输入 20，程序输出结果为 ()。
A. 1 B. 6 C. 7 D. 5
6. 若输入一个质数，程序最终输出结果是 ()。
A. 1 B. 2 C. 3 D. 不同的数结果不同

2.

```

1  #include<cstdio>
2  using namespace std;
3  int l,n;
4  int a[201],b[201];
5  void gjc()
6  {
7      int t=0;
8      for (int j=200;j>0;j--)
9          {
10             l=b[j]*2+t;
11             b[j]=l%10;
12             t=l/10;
13         }
14 }
15 void gjj()
16 {
17     int t=0;
18     for (int j=200;j>0;j--)
```

```

19     {
20         l=a[j]+b[j]+t;
21         a[j]=l%10;
22         t=l/10;
23     }
24 }
25 int main()
26 {
27     scanf("%d",&n);
28     b[200]=1;
29     for (int i=1;i<=n;i++)
30     {
31         gjc();
32         gjj();
33     }
34     int k=1;
35     while (a[k]==0&&k<200)
36         k++;
37     for (int i=k;i<=200;i++)
38         printf("%d",a[i]);
39 }

```

判断题:

1. 函数 gjc()对 b 数组中的数做了*2 操作。()
2. 对于储存在 a、b 数组中的数值，个位存储在 a[0]和 b[0]上。()
3. 两个函数中变量 t 中存储的都是进位。()
4. 将 34 行 k 的初始值设为 0，不影响程序。()

选择题:

5. 若输入 3，输出结果是()。
A. 7 B. 14 C. 5 D. 30
6. 以下哪个结果是程序可能输出的 ()
A. 30 B. 32 C. 7 D. 98

3.

```

1  #include<stdio>
2  #include<cstring>
3  #include<iostream>
4  #include<algorithm>
5  #include<cmath>

```

```

6  using namespace std;
7  int main()
8  {
9      int m,s,t,now=0;
10     cin>>m>>s>>t;
11     int s1=0,s2=0;
12     for(int i=1;i<=t;i++)
13     {
14         s1+=17;
15         if(m>=10) {s2+=60;m-=10;}
16         else m+=4;
17         if(s2>s1) s1=s2;
18         if(s1>s){
19             cout<<"Yes"<<endl<<i<<endl;
20             return 0;
21         }
22     }
23     cout<<"No"<<endl<<s1<<endl;
24     return 0;
25 }

```

判断题:

1. 程序输出必然是两行(不包含最后的空格和换行)。()
2. for 循环的循环次数必定为 t。()

选择题:

3. 输入 100 1000 50, 第二行输出是 ()。
 - A. 34
 - B. 917
 - C. 960
 - D. 35
4. 输入 39 200 4, 程序第 17 行的 s1=s2 共运行 () 次。
 - A. 4
 - B. 3
 - C. 2
 - D. 1
5. 一组数据, m 足够大, 输出 ()
 - A. 如果第一行是 No, 第二行一定是大于 s
 - B. 如果第一行是 Yes, 第二行一定大于 t
 - C. 如果第一行是 No, 第二行一定是 60 的倍数
 - D. 如果第一行是 No, 第二行一定是 17 的倍数
6. 以下数据中, 第一行输出为 No 的是 ()。
 - A. 36 355 10

- B. 30 600 10
C. 10000 700 15
D. 0 100 6

三、完善程序（单选题，每小题 3 分，共计 30 分）

1. （斐波那契拆分）已知任意一个正整数都可以拆分为若干个斐波纳契数，现在，让你求出 n 的拆分方法

输入：

一个数 t 表示有 t 组数据

接下来 t 行，每行一个数 n

输出：

t 行，每行一个字符串，表示拆分方法(格式: $n=a_1+a_2+a_3+...+a_n$)，要求从小到大输出

举例：

输入：

1

10

输出

10=2+8

思路：每次都取能取得最大值

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int a[50], b[50], n, t, w, x, y;
5
6  int main() {
7      a[1] = 1;
8      a[2] = 1;
9      for (int i = 3; i <= 45; i++) a[i]= ____ (1) ____;
10     cin >> n;
11
12     for (int i = 1; i <= n; i++) {
13         w = 0;
14         cin >> t;
15         x = t;
16         memset(b, 0, sizeof(b)); // 每次都要初始化 b
17         while (t > 0) {
18             y = ____ (2) ____;
19             while (y > 0) {
20                 if (a[y] <= t) {
21                     ____ (3) ____;

```

```

22         b[w] = a[y];
23         ____ (4) ____;
24     }
25     else y--; //有可能会有两个相同的数
26 }
27 }
28 cout << x << "=";
29 for (int j = w; j > 0; j--) //反着输出
30     if (____ (5) ____ ) cout << b[j] << endl;
31     else cout << b[j] << "+";
32 }
33 }

```

1. (1) 处应填 ()。
A. 0 B. i C. a[i-1]+a[i-2] D. a[i-1]+i

2. (2) 处应填 ()。
A. 45 B. 100 C. 0 D. 1

3. (3) 处应填 ()。
A. y-- B. w++ C. w-- D. y++

4. (4) 处应填 ()。
A. w-- B. t+=a[y] C. y-- D. t-=a[y]

5. (5) 处应填 ()。
A. j==0 B. j==1 C. b[j]==1 D. b[j]==0

2. (数独) 数独是根据 9*9 盘面上已知的数据, 推理出所有剩余空格的数字, 并满足每一行, 每一列, 每一个九宫内的数字均含 1-9, 不重复。

程序要求给一个未完成的数独, 没填的部分用 0 表示, 求出一个填好的数独。

举例说明:

输入:

```

800000000
003600000
070090200
050007000
000045700
000100030
001000068
008500010
090000400

```

输出:

```

812753649

```

```

9 4 3 6 8 2 1 7 5
6 7 5 4 9 1 2 8 3
1 5 4 2 3 7 8 9 6
3 6 9 8 4 5 7 2 1
2 8 7 1 6 9 5 3 4
5 2 1 9 7 4 3 6 8
4 3 8 5 2 6 9 1 7
7 9 6 3 1 8 4 5 2

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int sd[11][11];
4  bool p[11][11], l[11][11], fz[11][11];
5  void _out()
6  {
7      for(int i=1; i<=9; i++)
8      {
9          for(int j=1; j<=9; j++)
10             cout<<sd[i][j]<<" ";
11         cout<<endl;
12     }
13     return;
14 }
15 void dfs(int x, int y)
16 {
17     if(sd[x][y]!=0)
18         if(x==9&&y==9) _ ____ (1) ____;
19         else if(y==9) dfs(____ (2) ____);
20         else dfs(x, y+1);
21     else
22         for(int i=1; i<=9; i++)
23
24         if((!p[x][i])&&(!l[y][i])&&(!fz[(x-1)/3*3+(y-1)/3+1][i]))
25             {
26                 sd[x][y]=i; //填充!
27                 p[x][i]=l[y][i]=fz[(x-1)/3*3+(y-1)/3+1][i]=true; //
28                 打上标记。
29                 if(x==9&&y==9) _out();
30                 else if(y==9) dfs(x+1, 1);
31                 else dfs(x, y+1);
32                 ____ (3) ____
33                 p[x][i]=l[y][i]=fz[(x-1)/3*3+(y-1)/3+1][i]=false;
34             }
35 }

```



```

36 int main()
37 {
38     for(int i=1;i<=9;i++)
39         for(int j=1;j<=9;j++)
40             {
41                 int t;
42                 cin>>t;
43                 if(t!=0)
44                     p[i][t]=1[j][t]=fz[____(4)____][t]=true;
45                 sd[i][j]=t;//填充进数独。
46             }
47         dfs(____(5)____);
48     return 0;
49 }

```

1. (1) 处应填 ()。

- A. out() B. dfs (x+1,y) C. dfs(x+1,y+1) D. dfs(x,y+1)

2. (2) 处应填 ()。

- A. out() B. dfs (x+1,1)
C. dfs(x+1,y+1) D. dfs(x,y+1)

3. (3) 处应填 ()。

- A. sd[x][y]=0; B. sd[x][y]++; C. sd[x][y]- -; D. sd[x][y]=i;

4. (4) 处应填 ()。

- A. x B. y
C. (i-1)/3*3+(j-1)/3 D. (i-1)/3*3+(j-1)/3+1

5. (5) 处应填 ()。

- A. 1,1 B. 0,1
C. 0,0 D. 1,0