入门级 C++语言试卷（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. 如果 128 种颜色用二进制编码来表示，至少需要（　　）位。
A. 5　　B. 6　　C. 7　　D. 8

2. 若 12*25=311 成立，则用的是（　　）进制。
A. 11　　B. 8　　C. 7　　D.9

3. 一个袋子里装了 100 个苹果，100 个香蕉，100 个橘子，100 个梨子。从袋子中取出一个水果需要 1 分钟，那么需要分钟就能肯定至少已经拿出 1 打（12 个）相同种类的水果（　　）种。
A. 12　　B. 13　　C. 45　　D. 101

4. 下列程序段的时间复杂度是（　　）。

```
cnt=0;
    for(int i=1; i<n; i*=2)
        for(int j=1; j<n; j++)
            cnt++;
```

A. O(log2(n))　　B. O(n)　　C. O(n*log2(n))　　D.$O(n^2)$

5. 以下是 32 位机器与 64 位机器区别的是（　　）。
A. 硬盘大小不同
B. 显示器分辨率不同
C. 操作系统版本号不同
D. 寻址空间不同

6. 如果一棵二叉树的先序遍历是 ACDBEFG，中序遍历是 DCAEBFG，那么它的后序遍历是（　　）。
A. DCEGFBA
B. DCAEFGB
C. DCEFGBA
D. DCAEGFB

7. 如果开始时计算机处于小写输入状态，现在有一只小狗反复按照 CapsLock、字母键 A、字母键 B 、字母键 C 、字母键 D 的顺序来回按键，即 CapsLock、A、B、C、D、CapsLock、A、B、C、D、CapsLock、A、B、C、D、……，屏幕上输出的第 2020 个字符是字母（　　）。
A. 大写 A　　B. 大写 B　　C. 大写 C　　D. 大写 D

8. 如果进栈序列位 e1,e2,e3,e4，则不可能的出栈序列是：（ ）。
A. e2, e4, e3, e1　　B. e4, e3, e2, e1　　C. e1, e2, e3, e4　　D. e3, e1, e4, e2

9. 有如下程序:

```
#include <iostream>
using namespace std;
int a[3][3];
int main(){
   for(int i=0;i<3;i++)
         for(int j=0;j<3;j++)
               a[i][j]=i*3+j+1;
   for(int i=1;i<3;i++)
         for(int j=0;j<2;j++)
               cout<< a[j][i];
   return 0;
}
```

运行后的输出结果是（ ）。
A. 2536　　B. 2356　　C. 4758　　D. 4578

10. 8 名男生和 4 名女生围绕圆桌就坐,任意两个女生不相邻的坐法有（ ）种。
A. 67737600　　B. 8467200　　C. 40320　　D. 352800

11. 关于下面说法，错误的一项是（ ）。
A. ASCII 码是一种字符编码,常用 7 位码
B. 程序设计的三种基本结构是顺序、选择、循环
C. 刷新率、CPI、DPI 均可以用于描述鼠标性能
D. 目前世界上最大的计算机互联网络是 IBM 网

12. 小红课桌上有 5 本书，每本书都放有一个写着书名的书签。调皮的小王把小红的书签全部打乱了。那么，每个书签都没有插到对应的书中的可能性总共有（ ）种。
A. 38　　B. 40　　C. 44　　D. 48

13. 设 G 是有 7 个结点的完全图，要得到一棵生成树，需要从 G 中删去（ ）条边。
A. 10　　B. 12　　C. 15　　D. 18

14. 46! 的计算结果，尾数总共有（ ）个零。
A. 9　　B. 10　　C. 11　　D.12

15. 定义语句“double * array[8]”的含义正确的是（ ）。
A. array 是一个指针，它指向一个数组，数组的元素是双精度浮点型。
B. array 是一个数组，数组的每一个元素是指向双精度浮点型数据的指针
C. C++语言中不允许这样的定义语句

D. 以上都不对

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 main() {
4       int max, min, sum, cnt = 0;
5       int temp;
6       cin >> temp;
7       if (temp == 0) return 0;
8       max = min = sum = temp;
9       cnt++;
10      while (temp != 0) {
11          cin >> temp;
12          if (temp != 0) {
13              sum += temp;
14              cnt++;
15              if (temp > max) max = temp;
16              if (temp < min) min = temp;
17          }
18      }
19      cout << cnt << " " << min << " " << max << " " << sum/cnt <<endl;
20      return 0;
21  }
```

判断题：

1. 本程序统计了输入数字个数、最大值、最小值、平均值，输入遇-1 结束。（　　）

2. 对任意的整数值 x，只需要将第 7 行中 if 条件判断修改为 temp==x，就可以将本程序功能变成为输入遇值 x 结束。（　　）

3. 去掉第 8 行的连续赋值语句，或者忘记对第 4 行的整型变量 cnt 初始化为 0，这两种情况都可能影响程序最终输出结果。（　　）

4. 将 15 行 if 条件判断语句改成 temp>=max，并且把 16 行 if 条件判断语句改成 temp<=min，对程序结果没有任何影响。（　　）

选择题：

5. 若输入 0 1 2 3 4 5，程序运行到第（　　）行结束。

A. 6　　B. 7　　C. 10　　D. 20

6. 若输入 1 2 3 4 6 0 8 9，程序最终输出结果是（　　）
A. 5 1 6 3.2　　B. 8 1 9 4　　C. 6 1 6 3　　D. 5 1 6 3

2.

```
#include <stdio.h>
int gcd(int a, int b){
    if(b == 0) return a;
    return gcd(b,a%b);
}
void work(int a, int b) {
    int s[10009], t[10009], i = 0, d = 1, j;
    while(1) {
        if(a == 0) break;
        a *= 10; //除法借位
        t[i] = a;
        s[i] = a / b;
        a %= b;
        for(j = 0; j < i; ++j)
            if(t[j] == t[i]) {
                d = 0;
                break;
            }
        if(d == 0) break;
        printf("%d", s[i]);
        ++i;
    }
}
int main() {
    int a, b, g;
    scanf("%d%d", &a, &b);
    if(a > b) g = gcd(a, b);
    else g = gcd(b, a);
    a /= g;
    b /= g;
    printf("%d.", a/b); //格式化输出一个整数和一个小数点
    a %= b;
    work(a, b);
    return 0;
}
```

本程序输入一个分数的正整数分子 a 和正整数分母 b，将其转换为小数形式输出。

判断题：

1. 函数 gcd 使用了递归的方法，返回参数 a 和 b 的最小公倍数。（　　）

2. 33 行作为 work 函数的参数传入 a 和 b，一定满足 a<b 且 a≠0(　　)

3. 去掉 14-19 行的代码，将增加程序发生运行时错误的风险。(　　)

4. 去掉 27-30 行，程序输出结果一定发生改变。(　　)

选择题:
5. 若输入 91 13，输出结果是(　　)。
A. 7　　B. 7.　　C. 7.0　　D. 7.00

6. 以下哪个结果是程序可能输出的（　　）
A. 10.00　　B. 1.571428571428　　C. 1.5714285　　D. 1.571428

3.

```
1   #include <iostream>
2   using namespace std;
3   int solve(int n, int m){
4       int i, sum;
5       if (m == 1) return 1;
6       sum = 0;
7       for (i = 1; i < n; i++)
8           sum += solve(i, m - 1);
9       return sum;
10  }
11  int main(){
12      int n, m;
13      cin>>n>>m;
14      cout<<solve(n, m)<<endl;
15      return 0;
16  }
```

判断题:
1. 程序第三行改写成 solve(int m, int n)，程序运行结果不变。(　　)

2. 去掉程序第 6 行后，程序运行结果不变。(　　)

选择题:
3. 若输入的 n<m，则输出结果为（　　)。
A. 0　　B. n　　C. m　　D. m+(m+1)+...+(n-1)

4. 输入 4 3，程序第 8 行共循环（　　）次。
A. 3　　B. 6　　C. 9　　D. 12

5. 输入 7 4，输出是（　　）
A. 35　　B. 10　　C. 20　　D. 15

6.（4 分）以下四组输出数据中，输出最大的是（　　）。
A. 6 3　　B. 5 3　　C. 7 5　　D. 8 7

三、完善程序（单选题，每小题 3 分，共计 30 分）

1. （活动选择）学校在最近几天有 n 个活动，这些活动都需要使用学校的大礼堂，在同一时间，礼堂只能被一个活动使用。由于有些活动时间上有冲突，学校办公人员只好让一些活动放弃使用礼堂。现给出 n 个活动使用礼堂的起始时间 begini 和结束时间 endi（begini<endi）。现程序在第一行输入一个整数 n（n<=1000），接下来 n 行，每行两个整数，第 1 个是 begini，第 2 个是 endi（begini<endi<=32767），程序输出最多能安排的活动个数。

```
#include<iostream>
using namespace std;
int n, begin[1009], end[1009];
void init() {
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> begin[i] >> end[i];
}
void qsort(int x, int y) {
    int i, j, mid, t;
    i = ____(1)____;
    j = y;
    mid = ____(2)____;
    while(i <= j) {
        while(end[i] < mid) i++;
        while(____(3)____)  j--;
        if(i <= j) {
            t = end[j];
            end[j] = end[i];
            end[i] = t;
            t = begin[j];
            ____(4)____;
            begin[i] = t;
            i++;
            j--;
        }
    }
    if(x < j) qsort(x, j);
    if(i < y) qsort(i, y);
}
```

```
void solve() {
    int ans = 0;
    for(int i = 1, t = -1; i <= n; i++)
        if(begin[i] >= ____(5)____) {
            ans++;
            t = end[i];
        }
    cout << ans << endl;
}
int main() {
    init();
    qsort(1, n);
    solve();
    return 0;
}
```

1. （1）处应填（　　）。
A. 0　　B. x　　C. mid　　D. x

2. （2）处应填（　　）。
A. end[(x+y)/2]　　B.begin[(x+y)/2]　　C. (x+y)/2　　D. begin[(x+y)/2]+ end[(x+y)/2]

3. （3）处应填（　　）。
A. end[j]<mid　　B. end[j]>mid　　C. begin[j]<mid　　D. begin[j]>mid

4. （4）处应填（　　）。
A. begin[j]=t　　B. t=0　　C. begin[i]=begin[j]　　D. begin[j]=begin[i]

5. （5）处应填（　　）。
A. ans　　B. end[i]　　C. n+1　　D. t

2. （细胞问题）一个矩形阵列，由数字 0 到 9 组成。数字 1 到 9 是代表细胞。细胞的定义是，沿细胞数字上下左右四个方向，只要还是细胞数字，则被视为同一个细胞，求给出的矩形阵列的细胞个数。
如：阵列
4　10
0234500067
1034560500
2045600671
0000000089
有四个细胞。

```
#include<iostream>
#include<string>
using namespace std;
```

```
int direction_x[4] = _____;
int direction_y[4] = {0, 1, 0, -1};
int flag[100][100];
int num_of_cells = 0
int m,n;
void find(int start_x, int start_y) {
    int x, y, t, w, i;
    int queue[1000][2];
    num_of_cells++;
    flag[start_x][start_y] = 0;
    t = 0;
    w = 1;
    queue[1][1] = start_x;
    queue[1][2] = start_y;
    do {
        t++;
        for (i = 0; i <= 3; i++) {
            x = queue[t][1] + direction_x[i];
            y = _____(2)_____;
            if((x >= 0)&&(x < m)&&(y >= 0)&&(y <=m)&&(flag[x][y])) {
                w++;
                queue[w][1] = x;
                queue[w][2] = y;
                flag[x][y] = 0;
            }
        }
    }while(____(3)____);
}
int main() {
    int i, j;
    string s;
    cin >> m >> n;
    for(i = 0; i <= m - 1; i++)
        for(j = 0; j <= n - 1; j++)
            flag[i][j] = 1;
    for(i = 0; i <= m - 1; i++) {
        cin >> s;
        for(j = 0; j <= n - 1; j++)
            if(s[j] == '0')
                ____(4)____;
    }
    for(i = 0; i <= m - 1; i++)
        for(j = 0; j <= n - 1; j++)
            if(flag[i][j])
```

```
                ____(5)____;

    cout<<"NUMBER of cells is "<< num_of_cells << endl;
    return 0;
}
```

1. （1）处应填（　　）。

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

2. （2）处应填（　　）。

A. queue[t][2] + direction_y[i]　　B. queue[t][1] + direction_y[i]

C. queue[t][2] + direction_x[i]　　D. queue[1][t] + direction_y[i];

3. （3）处应填（　　）。

A. w!=0　　B.t>w　　C. t!=0　　D. t<w

4. （4）处应填（　　）。

A. flag[j]=0　　B. flag[i][j]=0

C. flag[i][j]='0'　　D. flag[i][j]=1

5. （5）处应填（　　）。

A. num_of_cells++　　B. flag[i][j]=0

C.find(queue[i][1],queue[j][2])　　D. find(i,j)