全国青少年信息学奥林匹克联赛初赛模拟考试（三）

（ 普及组　C++　语言　两小时完成 ）

●●　全部试题答案均要求写在答卷纸上，写在试卷纸上一律无效　●●

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、在标准 ASCII 码表中，已知英文字母 D 的 ASCII 码是 01000100，英文字母 B 的 ASCII 码是（　　）

A、0100000　B、01000010　C、01000011　D、01000000

2、在下列各种排序算法中，不是以“比较”作为主要操作的算法是（　　）。

A、选择排序 B、冒泡排序　C、插入排序　D、基数排序

3、在含有 n 个元素的双向链表中查询是否存在关键字为 k 的元素，最坏情况下运行的时间复杂度是（　）。

A、O(1)　B、O(log n)　C、O(n)　D、O(n log n)

4、广度优先搜索时，需要用到的数据结构是（　）。

A、链表　B、队列　C、栈　D、散列表

5、控制器（CU）的功能是（　　）。

A、指挥计算机各部件自动、协调一致地工作　B、对数据进行算术运算或逻辑运算

C、控制对指令的读取和译码　D、控制数据的输入和输出

6、微机中，西文字符所采用的编码是（　　）

A、EBCDIC 码　B、ASCII 码　C、国标码　D、BCD 码

7、Linux 是一种(　　)。

A、 绘图软件 B、 程序设计语言 C、操作系统 D、 网络浏览器

8、如果根结点的深度记为 1，则一棵恰有 2011 个叶结点的二叉树的深度最少是（　　）。

A、10　B、11　C、12　D、13

9、如果 256 种颜色用二进制编码来表示，至少需要（　　）位

A、6　B、7　C、8　D、9

10、下面叙述正确的是

A、算法的执行效率与数据的存储结构无关

B、算法的空间复杂度是指算法程序中指令(或语句)的条数

C、算法的有穷性是指算法必须能在执行有限个步骤之后终止

D、以上三种描述都不对

11、一片容量为 8GB 的 SD 卡能存储大约（　　）张大小为 2MB 的数码照片。

A、1600　B、2000　C、4000　D、16000

12、以下数据结构中不属于线性数据结构的是

A、队列　B、线性表　C、二叉树　D、栈

13、在一棵二叉树上第 5 层的结点数最多是

A、8　B、16　C、32　D、15

14、如果在一个非零无符号二进制整数之后添加一个 0，则此数的值为原数的（）

A、4 倍　B、2 倍　C、1/2　D、1/4

15、以下不是微软公司出品的软件是（　　）

A、Powerpoint　B、Word　C、Excel　D、Acrobat Reader

二、（判断题正确的填√，错误的填×；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

1、

```
1 #include <cstdio>
2 using namespace std;
3 int main(){
4       int a,b;
5       cin >> a >> b ;
6       if((++a<0)&&!(b--<=1))
7               printf("%d,%d\n",a,b);
8       else
9               printf("%d,%d\n",b,a);
10      return 0;
11 }
```

判断题

1、若第 6 行++a 改为 a++结果不会改变（）

2、若 a 为 0，那么将执行第 7 行代码（）

3、若 b 为 1，那么将执行第 9 行代码（）

4、若第 6 行++a<0 为假，那么后面的条件将无效（）

选择题

5、若输入数据 0 2，那么输出的结果为（）

A. 1,1　　B.1,2　　C.2,1　　D.2,0

6、若输入数据 -2 3，那么输出结果为（）

A. -1,3　　B.-2,3　　C.-2,2　　D.-1,2

2、

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 int main(){
5       char str[32] = {0};
6       cin >> str;
7       int len = strlen(str);
8       for(int i = 0 ; i < len; i++){
9               if( str[i] >= 'A' && str[i] <= 'Z')
10                      str[i] += 32;
11              if( str[i] >= 'a' && str[i] <= 'z')
12                      str[i] -= 32;
13      }
14      cout << str << endl;
15      return 0;
16 }
```

判断题

1、若第 10 行 str[i] += 32 改为 str[i] |= 32 结果不会改变（）

2、若第 12 行 str[i] -= 32 改为 str[i] |= 32 结果不会改变（）

3、程序功能为将大写转小写，小写转大写（）

4、若第 8 行 i < len 改为 i < strlen(str)，程序可能会变慢（）

选择题

5、若输入 AbCdEf，那么输出的结果为（）

A. ABCDEF　　B.abcdef　　C.aBcDeF　　D. AbCdEf

6、若输入 HellO，并再 14 行增加 str[3]='\0',那么输出的结果为（）

A. Hello　　B.Hel　　C.hel　　D.HEL

3、

```
1 #include<iostream>
2 using namespace std;
3 int f(int n,int k){
4       int sum=0;
5       if(k<=0)
6               return 1;
7       sum+=f(n-1,k-1);
8       if(n>k)
9               sum+=f(n-1,k);
10      return sum;
11 }
12 int main(){
13      int n,k;
14      cin >> n >> k;
15      cout<<f(n,k);
16      return 0;
17 }
```

判断题

1、 k<=0 为递归的结束条件（）

2、如果俩个参数相等，将会减少递归的次数（）

选择题

3、若输入 6 6，那么输出的结果为（）

A. 6　　B.1　　C.3　　D. 5

4、若输入 3 2，那么输出的结果为（）

A. 3　　B.4　　C.5　　D.6

5、若输入 6 3，那么输出的结果为（）

A. 21　　B.15　　C.18　　D.20

6、（4 分）若输入两个值相同，那么输出结果（）

A. 不确定　　B.两个数的和　　C.两个数的乘积　　D. 等于一

三 完善程序（每小题 3 分，总共 30 分）

1（子矩阵）输入一个 n1*m1 的矩阵 a，和 n2*m2 的矩阵 b，问 a 中是否存在子矩阵和 b 相等。若存在，输出所有子矩阵左上角的坐标；若不存在输出“There is no answer”。

```
#include <iostream>
using namespace std;
const int SIZE = 50;
int n1, m1, n2, m2, a[SIZE][SIZE], b[SIZE][SIZE];
int main() {
        int i, j, k1, k2;
        bool good, haveAns;
        cin >> n1 >> m1;
        for(i = 1; i <= n1; i++)
                for(j = 1; j <= m1; j++)
                        cin >> a[i][j];
        cin >> n2 >> m2;
        for(i = 1; i <= n2; i++)
                for(j = 1; j <= m2; j++)
                        ____①____;
        haveAns = false;
        for(i = 1; i <= n1 - n2 + 1; i++)
                for(j = 1; j <= ____②____; j++) {
                        ____③____;
                        for(k1 = 1; k1 <= n2; k1++)
                                for(k2 = 1; k2 <= ____④____; k2++)
                           {
```

```
                        if(a[i + k1 - 1][j + k2 - 1] != b[k1]
                [k2])
                        good = false;
                        }
                if(good){
                        cout << i << ' ' << j << endl;
                      ____⑤____;
                }
        }
    if(!haveAns)
            cout << "There is no answer" << endl;
    return 0;
}
```

1）①处应填（）

A. b[i][j] = a[i][j]　　B. cin >> b[n2][m2]

C. cin >> b[i][j]　　D. b[i][j] += a[i][j]

2) ②处应填（）

A. m1-m2　　B. m1-m2+1　　C. m1-m2-1　　D. m1+m2-1

3) ③处应填（）

A. good=true　B. good=false C. good=!good D. haveAns=true

4) ④处应填（）

A. m1　　B. n1　　C. m2　　D. n2

5) ⑤处应填（）

A. haveAns=false　B. haveAns=true　C. haveAns=!haveAns　D. good=true

2（全排列）下面程序的功能是利用递归方法生成从 1 到 n(n<10)的 n 个数的全部可能的排列（不一 定按升序输出）。例如，输入 3，则应该输出（每行输出 5 个排列）：　123 132 213　231 321　312

```
#include <iostream>
#include <iomanip>
int n,a[10];  // a[1],a[2],…,a[n]构成 n 个数的一个排列
long count=0;  // 变量 count 记录不同排列的个数，这里用于控制换行
void perm(int k){
        int j,p;
```

```
        if(   ①   ){
                count++;
                for(p=1;p<=n;p++)
                        cout <<setw(1)<<a[p];
                cout <<"  ";
                if(     ②   )
                        cout <<endl;
                return;
        }
        for(j=k;j<=n;j++){
                swap(a[k],a[j]);
                    ③    ;      ④    ;}
}
void main(){
        int i;
        cout <<"Entry n:"<<endl;
        cin >>n;
        for(i=1;i<=n;i++)
                a[i]=i;
        ⑤    ;
}
```

1）①处应填（）

A. k==n　　B. k==0　　C. k>=0　　D. k==n+1

2) ②处应填（）

A. count==5　　B. count/5 == 0

C. count%5 != 0　　D. count%5 == 0

3) ③处应填（）

A. perm(k)　　B. perm(k+1)　　C. perm(i)　　D. perm(i+1)

4) ④处应填（）

A. swap(a[k],a[i])　　B. swap(a[i],a[j])

C. swap(a[k],a[j])　　D. swap(a[k],a[n])

5) ⑤处应填（）

A. perm(1)　　B. perm(n)　　C. perm(0)　　D. perm(n+1)