LSOI2018 NOIP 初赛模拟赛

普及组 C++语言试题

竞赛时间：2018 年 9 月 10 日 12:30~2:00

选手注意：

- 试题纸共有 14 页，满分 100 分。
- 不得使用任何电子设备（如计算器、手机、电子词典等）或查询任何书籍资料。

一、单项选择题（共 20 题，每题 1.5 分，共计 30 分；每题有且仅有一个正确选项）

1．用二进制表示从 0 到 256 的所有整数（包括 0 和 256），最少需要（　C　）位。

A．7　　B．8　　C．9　　D．256

分析：注意是 0 到 256，一共有 257 个整数。

在二进制下用 x 位可以表示最多 2^x 个整数，所以可得最小需要 9 位。

2．1MB=（　D　）Byte。

A．2^{10}　　B．1000　　C．1000000　　D．2^{20}

分析：计算机常识。

3．冯·诺依曼型计算机由运算器、控制器、（　A　）、输入设备、输出设备五大部件组成。

A．储存器　　B．散热器　　C．缓存器　　D．风扇

分析：计算机常识。

4．在体育课上，若干同学无序排成一列。当数学老师看到有一个矮的同学站在一个高的同学后面时，就让这两位同学交换。在若干次交换后，所有同学完成了从矮到高的排序。这种方法类似于（　D　）。

A．选择排序　　B．插入排序　　C．计数排序　　D．冒泡排序

分析：

选择排序：每次选择 1 个最小/最大的数，然后放到序列的前面。

插入排序：将数组中的每个数分别插入到前面序列中正确的位置。

计数排序：开一个数组 c[]，c[i]表示数值 i 出现了多少次，统计后循环输出。

冒泡排序：每次比较相邻的两个数，如果大小顺序不符合要求则交换。

根据以上分析，结合题意，可得这道题选 D。

5．以下不属于 Linux 操作系统的是（　B　）。

A．Ubuntu　　B．Windows 1.0　　C．ReactOS　　D．Red Hat

分析：Ubuntu、ReactOS、Red Hat 均为比较有名的 Linux 操作系统。

如果不知道，也可以采用排除法：一般带有 Windows 字样的均为 Windows 操作系统。

6．以下为图片的文件名后缀的是（　A　）。

A．.gif　　B．.ppt　　C．.docx　　D．.rar

分析：计算机常识题。

7．以下关于计算机的说法错误的是（　C　）。

A．计算机可以在不安装 Windows 或 Linux 操作系统的情况下开机

B．计算机病毒可能能摧毁 BIOS 系统

C．ROM 储存的数据通常会在正常断电后丢失，但 RAM 通常不会

D．32 位机器和 64 位机器的寻址空间不同

分析：

A．正确。计算机可以使用内置的 BIOS 系统。

B．正确。病毒可以通过获取主板权限等方式烧坏主板或强行篡改可读文件的内容。

C．错误。ROM（Read-Only Memory）是只读的存储器，起到保存文件的作用，因而通常不会丢失。RAM（Random Access Memory）是可写可读的高速存储器，起到高速访问某些地址数据的作用，因而通常在断电后丢失。

D．正确。32 位机器通常有 4G 的寻址空间，64 位机器则有极大的寻址空间。

8．设 n 为目前链表内元素个数，那么链表可以在（　A　）的时间内在链表头新建一个元素。

A．O(1)　　B．O(n log n)　　C．O(log n)　　D．O(n)

分析：由于链表可以在 O(1)的时间内完成插入元素，因此我们只需要记录链表头，就可以做到 O(1)在链表头新建一个元素了。

9．在 NOIP 系列竞赛中，以下不能使用的是（　B　）。

A．#define　　B．多线程　　C．algorithm 库　　D．pb_ds 库

分析：蒙。NOIP 初赛经常出这种无厘头的题，一般 1 道左右，不会有太大的影响。

10．一个有 16 条边的无向连通图，最多可以有（　C　）个点。

A．15　　B．16　　C．17　　D．无数

分析：由于图必须联通，所以最优情况就是 1->2->3->4->...->16->17，共有 16 条边，17 个点。

11．对于操作序列{a，b，c，d，e，f，g}，以下为不合法的出栈序列的是（　D　）。

A．a b c d e f g　　B．a b c g f e d

C．c b g f e d a　　D．c b f e a d g

分析：模拟即可。

12．一个数在十六进制下有 100 位，则它在二进制下至少有（　A　）位。

A．397　　B．400　　C．403　　D．800

分析：我们可以把一个十六进制的数，通过按位转换来转换成二进制数。

e.g.

2	5	F
0010	0101	1111

所以$(25F)_{16}=(1001011111)_2$。

为了让位数最小，该十六进制数的最高位必定为 1，这样才能去掉尽可能多的前导 0。

所以可以发现，如果一个数在十六进制下有 n 位，那么它在二进制下就至少有(4n-3)位。

13．以下排序算法最坏需要$O(n^2)$的是（　B　）。

A．基数排序　　B．快速排序　　C．堆排序　　D．归并排序

分析：

基数排序的最坏/最优/平均复杂度为 O(d(n+r))，其中 d 为关键字数量，n 为序列长度，r 为关键字值域。

快速排序的最优/平均复杂度为 O(n log n)，但是无论是使用随机游标、三位取中还是同项合并的快速排序，都存在一个输入使其退化为$O(n^2)$。尽管该概率极小，但仍然是不可忽略的。因而快速排序的最坏复杂度为$O(n^2)$。

堆排序/归并排序的最坏/最优/平均复杂度为 O(n log n)。

14．NOIP 的全称是（　A　）。

A．National Olympiad in Informatics in Provinces

B．National Olympiad in Informatic Provinces

C．National Olympiad in Informatics Plus

D．Naive Olympiad in Informatic Provinces

分析：计算机常识题。

15．设根节点的深度为 1，那么一个有 100 个节点的树深度至少为（　B　）。

A．1　　B．2　　C．7　　D．100

分析：注意到题目没有说明该树是二叉树，所以为了让深度最小，我们可以构造出一棵树，使得根节点有 99 个孩子。这样，这棵树的深度就是 2。

16．看下面的程序：

```
    res = 0;
    for (i = 1; i <= n; i++)
        if(i % 2 == 0)
            res += i;
```

其中 n 是一个不超过 100 的正整数。

与这个程序片段同等的语句是（ C ）。

A. `res = (2 + n / 2 * 2) * (n / 2) / 2;`

B. `res = (2 + n) * (n / 2) / 2;`

C. `res = (2 + n * 2 / 2) * (n / 2) / 2;`

D. `res = (2 + n / 2 * 2) * (n / 2);`

分析：使用求和公式就可以得出答案是 C 了。

注意 n/2*2 不一定等于 n，所以不能选择 B 项。

17. 以下关于字符串的说法错误的是（ C ）。

A. 在任意字符串中，字典序最小的子串长度一定不超过 1

B. 字符串是一种特殊的线性表

C. 字符串不可以为空

D. 字符串需要的储存空间与字符串长度线性相关

分析：计算机常识题。

A 题是最近的百度之星原题，给出结论后就很容易证明了。

18. 书架上放有 3 本语文书、5 本数学书和 2 本英语书，书本两两不同。要从中分别取出 1 本语文书、数学书和英语书，则有（ A ）种不同的取法。

A. 30　　B. 10　　C. 8　　D. 1

分析：根据“书本两两不同”，我们就可以使用乘法原理求出答案了。

19．1 种彩票，中奖的几率为 20%。买了 5 张彩票，没有一张中奖的概率最接近于（ C ）。

A．0　　B．22.5%　　C．32%　　D．20%

分析：

这个涉及到概率期望相关的知识。

如果只买 1 次彩票，没有中奖的概率为 80%；

如果只买 2 次彩票，没有中奖的概率为 80%×80%=64%；

同样地，买 5 次彩票没有一张中奖的概率为$(80\%)^5$=32.768%，近似于 C 选项中的 32%。

20．以下不是输入设备的是（　D　）。

A．鼠标

B．键盘

C．扫描仪

D．显示屏

分析：计算机常识题。

二、问题求解（共 2 题，每题 5 分，共计 10 分）

1．在 10000 以内，能被 3 或 5 整除的数共有__4667__个。

分析：小学奥数题。

答案为(10000 以内 3 的倍数)+(10000 以内 5 的倍数)-(10000 以内 15 的倍数)，因为 15 的倍数被计算了 2 次，所以要减去。

也就是 floor(10000/3)+floor(10000/5)-floor(10000/15)，其中 floor(x)表示小于等于 x 的最小整数，如 floor(3.5)=3，floor(3)=3。

2．设根节点的深度为 1，f(x)为有 x 个节点的二叉树的最小深度，则令 f(x)=10 的 x 的不同取值共有__511__个。

分析：在最小深度的情况下，该二叉树是一个完全二叉树。根据二叉树的计算公式，可以得出答案为 511。

三、阅读程序写结果（共 4 题，每题 8 分，共计 32 分；第一题至第三题答对各得 8 分，第四题每空 4 分）

1.
```
#include <iostream>
using namespace std;
int main() {
    int sum = 0, count = 0;
    int tmp;

    cin >> tmp;
    while (tmp != 0) {
        count++;
        sum += tmp % 2;
        tmp /= 2;
    }
    cout << "count: " << count << ", sum: " << sum << endl;
    return 0;
}
```

输入：13

输出：count: 4, sum: 3

分析：这个程序对于输入的一个整数，输出它在二进制下的位数和它在二进制下，数位上是 1 的位数个数。

输入数据较小，直接模拟即可。

2.
```
#include <iostream>
using namespace std;
int main() {
    string a = "ZINCSABIAN";
    string b = "zincsablan";
    int i, n;
    int sum = 0;

    n = a.length();
    for (i = 0; i < n; i++) {
        if (a[i] >= 'A' && a[i] <= 'Z')
            a[i] += 'a' - 'A';
        if (b[i] >= 'A' && b[i] <= 'Z')
            b[i] += 'a' - 'A';
        sum += a[i] - b[i];
    }
    cout << sum << endl;
    return 0;
}
```

输出：___-3___

分析：同样，模拟就可以得出答案。

注意字符串 a 中是 SABIAN 而 b 中是 sablan，因而输出的应该是-3。

‘A’..’Z’的 ASCII 码是 65..90;

’a’..’z’的 ASCII 码是 97..122;

‘0’..’9’的 ASCII 码是 48..59。

3.
```
#include <iostream>
using namespace std;
int main() {
    int i, j, k;
    int n, m, ans;
    string a, b;

    cin >> a >> b;
    n = a.length();
    m = b.length();
    ans = 0;
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++){
            k = 0;
            while (i + k < n && j + k < m && a[i+k] == b[j+k])
                k++;
            if (ans < k)
                ans = k;
        }
    }
    cout << ans << endl;
    return 0;
}
```

输入：zinctql znistql

输出：____3____

分析：这个程序通过枚举，求出字符串 a 和 b 中公共字串的长度。

4.
```
#include <iostream>
using namespace std;
const int a[] = {2, 5};
int q[100001];
int vis[100001];
int main() {
    int i, j;
    int ql, qr;
    int n;

    cin >> n;
    ql = qr = 1;
    q[1] = 1;
    vis[1] = 1;
    while (ql <= qr) {
        i = q[ql];
        ql++;
        for (j = 0; j < 2; j++) {
            if (vis[i+a[j]] == 0 && i+a[j] <= 100000){
                vis[i+a[j]] = 1;
                qr++;
                q[qr] = i+a[j];
            }
```

```
            }
        }
        i = 1;
        while (n > 0) {
            n -= vis[i];
            if(n > 0)
                i++;
        }
        cout << i << endl;
        return 0;
    }
```

输入 1：5

输出 1： 7

输入 2：65535

输出 2： 65537

分析：这个程序求出对于输入的 k，求出对于 1<=i<=n，使 i 可以表示为 2x+5y 的 i 的个数等于 k，的最小的 n 是多少。

例如：

1 无法被表示；

2=2；

3 无法被表示；

4=2*2；

5=5；

6=2*3；

7=2*2+5。

在 1..7 中有 5 个数可以被表示，所以 n=7。

四、完善程序（共 2 题，每题 14 分，共计 28 分）

1.（一元二次方程求解）给出 $ax^2+bx+c=0$ 方程中的 a、b 和 c（0<=a, |b|, |c|<=10^9），求出 x 的值（精确到 6 位小数）。保证 x 有且仅有一种取值，且|x|<=10^9。

因为这个题目保证了 x 有且仅有一种取值，且函数 $f(x)=ax^2+bx+c$ 满足单调性，所以我们可以采取二分的方法解决这个问题。

```
#include <iostream>
using namespace std;
double a, b, c;
double f(double x) {
    return a * x * x + b * x + c;
}
int main() {
    double l, r, mid;

    cin >> a >> b >> c;
    l = -1000000000;
    r = 1000000000;
    while (r - l >= 0.0000001) {
        mid = (l + r) / 2;
        if(f(mid) > f(mid + 0.000000001)) l = mid;
        else r = mid;
    }
    cout << l;
    return 0;
```

```
}
```

分析：本题是二分题目的模板，不过有些特别。

注意 while 循环的条件不可以写 l<=r，因为 l 和 r 是浮点数，浮点数计算可能有误差，因此可能会出现死循环的情况。

2．（激光手雷）在一个 n×m（1≤n，m≤1000）的棋盘上，每个格子上都写有一个数字 x（1≤x≤100）。你可以选择第 i 行第 j 列的格子投掷激光手雷，则总得分为该列和该行上所有格子数字之和。给出棋盘，请你计算出可能的最大总得分。

```
#include <iostream>
using namespace std;
int a[1001][1001]; // 棋盘每个格子上的数字
int sa[1001][1001]; // sa[i][j]=a[i][1]+a[i][2]+...+a[i][j]
int sb[1001][1001]; // sb[i][j]=a[1][j]+a[2][j]+...+a[i][j]
int main() {
    int i, j;
    int n, m;
    int res, sum;

    cin >> n >> m;
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= m; j++){
            cin >> a[i][j];
        }
    }
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= m; j++){
            sa[i][j] = sa[i][j-1] + a[i][j];
```

```
            sb[i][j] = sb[i-1][j] + a[i][j];
        }
    }
    res = 0;
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= m; j++){
            sum = sa[i][m] + sb[j][n] - a[i][j];
            if (res < sum)
                res = sum;
        }
    }
    cout << res << endl;
    return 0;
}
```

分析：本题是前缀和普通题，思路源自近年 NOIP 普及组初赛求最大子矩阵。

注意要读懂题意：该手雷影响的区域是整行和整列，是十字形的，而不是 sa[i][j]+sb[i][j]-a[i][j]。