

信息学竞赛初赛笔试练习卷（六）

（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、单选题（20 题，每题 1 分，多选无分，共 20 分）

1. 计算机内部存储、处理信息时之所以要采用二进制，是因为（ ）。
A、制造两个稳定状态的电路要比制造有多个稳定状态的电路容易得多，并且很容易实现高速处理
B、二进位的运算规则很简单
C、使用二进位不仅能表示数值信息，还能表示文字、符号、图像、声音等多种不同形式的信息。由于均使用比特作为其基本单位，因此，它们可以毫不费力地相互混合，形成多媒体
D、信息使用比特表示以后，可以通过多种方法进行“数据压缩”，即使用更少的比特来表示同样的文字、声音或影像信息，从而大大降低信息传输和存储的成本
下面哪一个选择是正确的？
A、都对 B、1 和 2 C、1、2 和 3 D、2、3 和 4
2. 美籍匈牙利数学家 冯·诺依曼 对计算机科学发展所做出的贡献是（ ）。
A、提出理想计算机数学模型，成为计算机科学理论基础 Alan Turing
B、是世界上第一个编写计算机程序的人 Ada
C、提出存储程序工作原理，并设计出第一台具有存储程序功能的计算机 EDVAC
D、指出计算机性能将以每两年翻一番的速度向前发展
3. 第二代计算机所采用的主要元器件是（ ）。
A、电子管 B、晶体管 C、中小规模集成电路 D、大规模、超大规模集成电路
4. 计算机内存中的每个基本单元都被赋予了一个唯一的序号，称为（ ）。
A、地址 B、字节 C、编号 D、容量
5. 下面关于比特（即二进位）的叙述中，错误的是（ ）。
A、比特是组成数字信息的最小单位
B、比特只有“0”和“1”两个符号
C、比特既可以表示数值和文字，也可以表示图像和声音
D、比特“1”大于比特“0”
6. 微机销售广告中“P4 2.4G/256M/80G”中的 2.4G 是表示：
A、CPU 的运算速度为 2.4GIPS
B、CPU 为 Pentium 4 的 2.4 代
C、CPU 的时钟主频为 2.4GHZ
D、CPU 与内存间的数据交换速率为 2.4Gbps

7. 一台微机地址总线的位长为 32 位, 则其最大的寻址空间为 ()。
- A、4GB B、3MB C、2KB D、32B
8. 中央处理器 (CPU) 能访问的最大存储器容量取决于 ()。
- A. 地址总线 B. 数据总线 C. 控制总线 D. 实际内存容量
9. 64KB 的存储器用十六进制表示, 它的最大的地址码是 ()。
- A、10000 B、FFFF C、1FFFF D、EFFFF
10. 至少需要多大存储容量才能存放一幅分辨率为 1024×768 的黑白图像 (不压缩) ()。
- A、96K B、100M C、100K D、1G
11. 下断电之后仍能保存数据的有 ()。
- A. 寄存器 B. ROM C. RAM D. 高速缓存
12. 一个文本屏幕有 25 列及 80 行, 屏幕的左上角以 (1, 1) 表示, 而右下角则以 (80, 25) 表示, 屏幕上每一个字符占用两字节 (byte), 整个屏幕则以线性方式存储在电脑的存储器内, 内屏幕左上角开始, 位移为 0, 然后逐列存储。求位于屏幕 (X, Y) 的第一个字节的位移是 ()。
- A. $(Y * 80 + X) * 2 - 1$ B. $((Y - 1) * 80 + X - 1) * 2$
C. $(Y * 80 + X - 1) * 2$ D. $((Y - 1) * 80 + X) * 2 - 1$
13. 操作系统的作用是 ()。
- A. 把源程序译成目标程序 B. 便于进行数据管理
C. 控制和管理系统资源 D. 实现硬件之间的连接
14. 磁盘缓冲区是 ()。
- A、在 ROM 存储器中建立的一个保留区域
B、读写磁盘文件时用到的内存中的一个区域
C、磁盘上存放暂存数据的存储空间
D、上述三者都不对
15. 虚拟存储器是 ()。
- A、磁盘上存放数据的存储空间
B、读写磁盘文件数据时用到的内存中的一个区域
C、操作系统为用户作业提供的、比计算机中的实际内存大得多的外存区域
D、将部分 ROM 存储器作为磁盘驱动器
16. 关于剪切和删除的叙述中, () 是正确的。
- A. 剪切和删除的本质相同
B. 不管是剪切还是删除, Windows 都会把选定的内容从文档中删除掉
C. 剪切的时候, Windows 只是把文字存在了文件名为“剪贴板”的硬盘里
D. 删除了的文字不再恢复

17. 因特网上的每一台主机都有一个唯一的标识, 它是 ()。
- A. IP 地址 B. 定位器 C. 计算机名 D. 用户名
18. WWW 浏览器使用的网络协议是 ()。
- A. HTTP B. TCP/IP C. FTP D. Telnet
19. 超级链接不可以是文件中的 ()。
- A. 一个词 B. 一个词组 C. 一种颜色 D. 一幅图像
20. Cable MODEM 是常用的宽带接入方式之一。下面关于 Cable MODEM 的叙述中错误的是 ()。
- A. 它利用现有的有线电视电缆线作为传输介质
- B. 它的带宽很高, 数据传输速度很快
- C. 用户可以始终处于连线状态, 无需像电话 MODEM 那样拨号后才能上网
- D. 在上网的同时不能收看电视节目

二、问题求解 (三题, 每题 4 分, 共 12 分)

1. 把 S 集合中的 n 个元素 a_1, a_2, \dots, a_n 放入 k 个 ($0 < k \leq n < 30$) 无标号的盒子中, 使得没有一个盒子为空。请你确定当 $n=4, k=3$ 时, 将 n 个元素 a_1, a_2, \dots, a_n 放入 k 个无标号盒子中去的划分数 $S(n, k)$ 。
2. 平面上有三条平行直线, 每条直线上分别有 7, 5, 6 个点, 且不同直线上三个点都不在同一条直线上。问用这些点为顶点, 能组成多少个不同四边形?

3、**LZW 编码**是一种自适应的词典编码。在编码的过程中, 开始时只有一部基础构造元素的编码词典, 如果在编码的过程中遇到一个新的词条, 则该词条及一个新的编码会被追加到词典中, 并用于后继信息的编码。

举例说明, 考虑一个待编码的信息串: “xyx yy yy xyx”。初始时词典中只有 3 个条目, 第一个为 x , 编码为 1; 第二个为 y , 编码为 2; 第三个为空格, 编码为 3。于是, 串 “xyx” 的编码为 1-2-1 (其中 “-” 为编码分隔符), 加上后面的一个空格就是 1-2-1-3。但由于有了一个空格, 我们就知道前面的 “xyx” 是一个单词, 而由于该单词没有出现在词典中, 我们就可以自适应地把这个词条添加到词典里, 编码为 4。然后, 按照新的词典, 对后继信息进行编码, 依此类推。于是, 最后得到编码: 1-2-1-3-2-2-3-5-3-4。

现在，已知初始词典的 3 个条目如上述，则信息串 “yyxy xx yyxy xyx xyx” 的编码是__

三、阅读程序（5 题，每题 7 分，共 35 分）

```
1 .
#include <iostream>
using namespace std;

int main()
{
    int a[12], k, i;

    a[1] = 1;
    a[2] = 1;

    for (k=2; k<=10; k++) {
        a[k+1] = 1;
        for (i=k; i>=2; i--) {
            a[i] += a[i-1];
        }
    }

    for (i=1; i<=11; i++) cout << a[i] << " ";
    return 0;
}
输出:
```

```
2.
#include <iostream>
using namespace std;

int main()
{
    string n;
    int s, i;

    cin >> n >> s;

    while (s > 0) {
```

```

        i = 0;
        while (i<n.size()-1 && n[i] <= n[i+1]) i++;
        n = n.substr(0,i) + n.substr(i+1);
        s--;
    }

    while (n.size()>1 && n[0]==48) n = n.substr(1);

    cout << n;
    return 0;
}
输入: 178543, 4    输出:

```

3.

```

#include <iostream>
using namespace std;

int main()
{
    int i, j, a;

    cin >> i;
    a = 1;

    for (j=1; j<=i+1; j++) {
        cout << a << " ";
        a = a * (i - j + 1) / j;
    }
    return 0;
}
输入: 6            输出:

```

4.

```

#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    float x, n;
    int i, j;

```

```

cin >> x;
if (x < 0) cout << "-";
n = fabs(x);

do {
    i++;
    n = n / 2;
} while (n != 0);

for (j=1; j<=i; j++) {
    cout << int(n * 2);
    n = n * 2 - int(n * 2);
}

if (x != int(x)) {
    cout << ".";
    n = fabs(x) - int(fabs(x));
    do {
        cout << int(n * 2);
        n = n * 2 - int(n * 2);
    } while (n != 0);
}

return 0;
}

```

输入: -1234.6875 输出:

5.

```

#include <iostream>
using namespace std;

int main()
{
    int a[51], x, y, z, i, j;

    x = 6;
    y = 2345;
    x = 9;

    i = 21;
    while (y > 0) {
        i--;
    }
}

```

```

        a[i] = y % 10;
        y = y / 10;
    }

    s = 0;
    for (j=i; j<=20; j++)
        s = s * x + a[j];

    cout << s;

    i = 21;                //i--;
    do {
        i--;
        a[i] = s % z;
        s = s / z;
    } while ( s > 0);

    for (j=i; j<=20; j++)
        cout << a[j] << " ";

    return 0;
}

```

输出：

五、完善程序（18 个空，每空 2 分，共 36 分）

1. 删除重复的数据 (del.cpp)

【问题描述】

本程序对随机产生的 100 个 0 到 50 之间的随机整数用一个数组存放后进行排序，然后再将其中重复出现的数进行删除，只保留一个，使得剩下的数中任何两个都不相同且连续存储在原数组中。

【程序清单】

```

#include <iostream>
using namespace std;

int main()
{
    int maxn=100;
    int a[maxn+1], i, j;

    for (i=1; i<=maxn; i++)
        a[i] = rand(timer(0))*5+1;
}

```

```

for (i=1; i<= ① ; i++)
    for (j= ② ; j<=maxn; j++)
        if (a[i] < a[j]) swap(a[i], a[j]);

for (i=2; i<=maxn; i++)

    if ( ③ ) a[i-1]=-a[i-1];

tail = 0;
current = 1;

while ( ④ )
    while (a[current] < 0) current++;
    tail++;
    a[tail] = ⑤ ;
    current++;
}

if ( ⑥ ) a[++tail] = 0;

for (i=1; i<=tail; i++) cout << a[i] << " ";
return 0;
}

```

2. 添加括号 (bracket.bas)

【问题描述】

输入 N 个整数（可正可负）及一个整数 M ，在这 N 个数中间添加一对括号，使它们的代数和等于 M ，并打印此等式（若无相等可能，则打印“NO”）。

例： $N=6$ (1, -2, -3, 4, -5, 6) $M=9$

则有 $1 - (2 - 3 + 4 - 5) + 6 = 9$ ‘P=1 : S = (-3+4-5)=-4

【程序清单】

```

#include <iostream>
using namespace std;

int main()
{
    int n, m, i, p, j, s, x;
    cin >> n >> m;
    int a[n+1];

    for (i=1; i<=n; i++) {
        cin >> a[i];
        p += a[i];
    }
}

```



```

    }

    for (i=1; i<=n-1; i++) {
        ① _____ ;
        if (a[i] > 0 ) continue;
        for (j=i+1; j<=n; j++) {
            s += a[j];

            if ( ② _____ ) goto 100;
        }
    }

    cout << "NO!";

    ③ _____ ;

100
    for (x=1; x<=n; x++) {
        if (a[x] > 0 && ④ _____ ) cout << "+";

        if ( ⑤ _____ ) {
            cout << "- (";
            a[x] = - a[x];
        }
        cout << a[x];

        if ( ⑥ _____ ) cout << ")";
    }

    cout << "=" << m;
    return 0;
}

```

3. 素数乘法算式 (prime_mul.bas)

【问题描述】

下面的竖式表示乘法运算，左图中的“*”只能用素数 2、3、5、7 代替（右图是运行结果），因此称为素数乘法竖式。

$ \begin{array}{r} *** \\ \times \quad ** \\ \hline ***** \\ ***** \\ \hline ***** \end{array} $	$ \begin{array}{r} 775 \\ \times \quad 33 \\ \hline 2325 \\ 2325 \\ \hline 25575 \end{array} $
--	--

【程序清单】

```
#include <iostream>
```

```

using namespace std;

int main()
{
    int p_num[5];
    int i, j, k, m, n, a, b, c, d, p, t, x, y ;
    string x_str;

    p_num[1]=2; p_num[2]=3; p_num[3]=5; p_num[4]=7;

    for (i=1; i<=4; i++)
        for (j=1; j<=4; j++)
            for (k=1; k<=4; k++)    //k=2

                for (m= ① _____; m++)

                    for (n= ② _____; n++) {

                        a = ③ _____;

                        b = ④ _____;

                        c = p_num[n] * a; d = p_num[m] * a;
                        x = c; goto 500;
                        if (t == 0) goto 10;
                        x = d; goto 500;
                        if (t != 0) x= ⑤ _____ ; goto 500;
                        if (t != 0) {
                            cout << a;
                            cout<<"*"<<b;
                            cout<<"-----";
                            cout<<c;
                            cout<<d;
                            cout<<"-----";
                            cout<<10*d+c;
                            cout<<endl;
                        }
                        cout<<endl;
                    }
                }
            }
        }
    }
    return 0;
}

```

500

```

x_str = str(x);
for (p=0; p<x_str.size(); p++) {

    y = ⑥;
    if (y!=2 && y!=3 && y!=5 && y !=7) goto 510;
}
t = 1;
return;

510
t = 0; return;

```