CSP-J 初赛模拟卷（第 04 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．Linux下可执行文件的默认扩展名为（　）。

A．exe　　B．com　　C．dll　　D．以上都不是

2．冒泡排序算法的伪代码如下。 输入：长度为 n 的数组 L。输出：按非递减顺序排序的 L。

```
01 FLAG ← n
02 while FLAG > 1 do
03     k ← FLAG - 1
04     FLAG ← 1
05     for j = 1 to k do
06         if L(j) > L(j+1) then
07             L(j) ↔ L(j+1)
08             FLAG ← j
```

对 n 个数用以上冒泡排序算法进行排序，最少需要比较多少次？（ ）

A．n^2　　B．n - 2　　C．n - 1　　D．n

3．地址总线的位数决定了 CPU 可直接寻址的内存空间大小，例如地址总线为 16 位，其最大的可寻址空间为 64KB。如果地址总线是 32 位，则理论上最大可寻址的内存空间为（　）。

A．128KB　　B．1MB　　C．1GB　　D．4GB

4．无向完全图是图中每对顶点之间都恰好有一条边的简单图。已知无向完全图G有7个顶点，则它共有（　）条边。

A．7　　B．21　　C．42　　D．49

5．下列说法正确的是（　）。

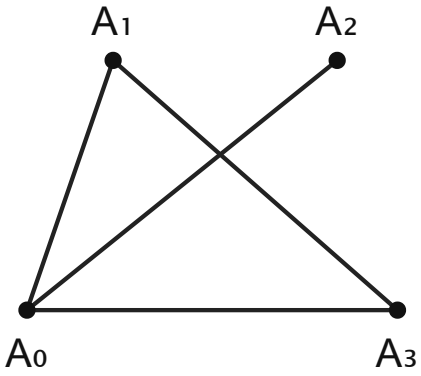
A．CPU的主要任务是执行数据运算和程序控制

B．存储器具有记忆能力，其中信息任何时候都不会丢失

C．两个显示器屏幕尺寸相同，则它们的分辨率必定相同

D．个人用户只能使用 Wifi 的方式连接到 Internet

6．以A_0作为起点，对下面的无向图进行深度优先遍历时，遍历顺序不可能是（　）。



A．A_0, A_1, A_2, A_3　　B．A_0, A_1, A_3, A_2　　C．A_0, A_2, A_1, A_3　　D．A_0, A_3, A_1, A_2

7．完全二叉树的顺序存储方案，是指将完全二叉树的结点从上至下、从左至右依次存放到一个顺序结构的数组中。假定根结点存放在数组的1号位置，则第k号结点的父结点如果存在的话，应当存放在数组的（　）号位置。

A．2k　　B．2k+1　　C．k/2下取整　　D．(k+1)/2下取整

8．下列几个32位IP地址中，书写错误的是（　）。

A．162.105.135.27　　B．192.168.0.1　　C．256.256.129.1　　D．10.0.0.1

9．设 X 、 Y 、 Z 分别代表三进制下的一位数字，若等式 $XY + ZX = XYX$ 在三进制下成立，那么同样在三进制下，等式$XY * ZX = ()$ 也成立。

A．YXZ　　B．ZXY　　C．XYZ　　D．XZY

10．有7个一模一样的苹果，放到3个一样的盘子中，盘子可以为空，一共有（　）种放法。

A．7　　B．8　　C．21　　D．3^7

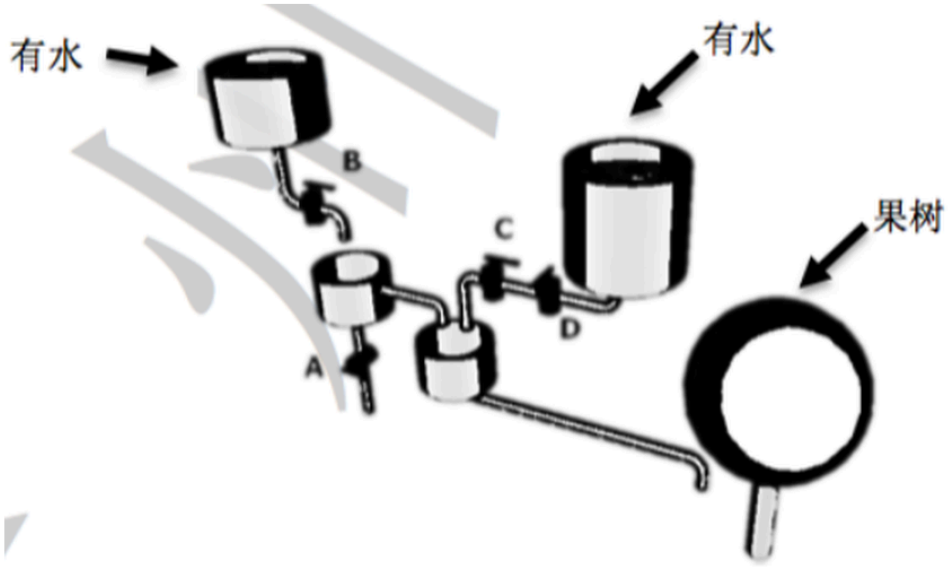
11．现有一段文言文，要通过二进制哈夫曼编码进行压缩。简单起见，假设这段文言文只由4个汉字“之”、“乎”、“者”、“也”组成，它们出现的次数分别为700、600、300、400。若待合并结点权值相同，优先选取由其他结点合并形成的结点。那么，“也”字的编码长度可能是（　）。

A．1　　B．2　　C．3　　D．4

12．某算法的计算时间表示为递推关系式 $T(n) = T(n-1) + n(n$ 为正整数$)$及 $T(0) = 1$，则该算法的时间复杂度为（　）。

A．$O(\log n)$　　B．$O(n \log n)$　　C．$O(n)$　　D．$O(n^2)$

13. 下图表示一个果园灌溉系统，有 A、B、C、D 四个阀门，每个阀门可以打开或关上，所有管道粗细相同，以下设置阀门的方法中，可以让果树浇上水的是（ ）。



A. B 打开，其他都关上　B. AB 都打开，CD 都关上
C. A 打开，其他都关上　D. D 打开，其他都关上

14. 10000 以内，与 10000 互质的正整数有（ ）个。

A. 2000　B. 4000　C. 6000　D. 8000

15. 一些数字可以颠倒过来看，例如0、1、8颠倒过来还是本身，6颠倒过来是9，9颠倒过来看还是6,其他数字颠倒过来都不构成数字。 类似的，一些多位数也可以颠倒过来看，比如106颠倒过来是901。假设某个城市的车牌只由5位数字组成，每一位都可以取0到9。 请问这个城市最多有多少个车牌倒过来恰好还是原来的车牌？（ ）

A. 60　B. 125　C. 75　D. 100

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 ×；共 18 小题，共计 42 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 int main()
04 {
05     const int SIZE=10;
06     int height[SIZE],num[SIZE],n,ans;
07     cin>>n;
```

```
08      for(int i=0;i<n;i++){
09          cin>>height[i];
10          num[i]=1;
11          for(int j=0;j<i;j++){
12              if((height[j]<height[i])&&(num[j]>=num[i]))
13                  num[i]=num[j]+1;
14          }
15      }
16      ans=0;
17      for(int i=0;i<n;i++){
18          if(num[i]>ans) ans=num[i];
19      }
20      cout<<ans<<endl;
21      return 0;
22  }
```

判断题

16. (1.5分) 如果height数组中的输入有负数，则程序会出错。（ ）

17. (1.5分) 程序输出的 ans 小于或等于n。（ ）

18. (1.5分) 将12行num[j]>=num[i]“改为”num[j]>num[i]“，程序的输出结果不会改变。（ ）

19. (1.5分) 将18行”num[i]>ans”改为”num[i]>=ans”，程序的输出结果不会改变。（ ）

单选题

20. (3分) 若输入的数据为 10 1 1 1 1 1 1 1 1 1 则程序的输出结果是（ ）。

A. 4　　B. 3　　C. 2　　D. 1

21. (3分) 若输人的数据为 10 3 2 5 11 12 7 4 10 15 6 因程序的输出结果是（ ）。

A. 4　　B. 5　　C. 6　　D. 7

(2)

```
01 #include <iostream>
02 using namespace std;
03 const int maxn = 10000;
04 int n;
05 int a[maxn];
06 int b[maxn];
07 int f(int l, int r, int depth) {
08     if (l > r)
09         return 0;
10     int min = a[l], mink = l;
11     for (int i = l; i <= r; ++i) {
12         if (min > a[i]) {
13             min = a[i];
14             mink = i;
15         }
16     }
17     int lres = f(l, mink - 1, depth + 1);
18     int rres = f(mink + 1, r, depth + 1);
19     return lres + rres + depth * b[mink];
20 }
21 int main() {
22     cin >> n;
23     for (int i = 0; i < n; ++i)
24         cin >> a[i];
25     for (int i = 0; i < n; ++i)
26         cin >> b[i];
27     cout << f(0, n - 1, 1) << endl;
28     return 0;
29 }
```

判断题

22. (1.5分) 如果a数组有重复的数字，则程序运行时会发生错误。（ ）

23. (1.5分) 如果b数组全为0，则输出为0。（ ）

单选题

24. (3分) 当 $n = 100$ 时，最坏情况下，与第12行的比较运算执行的次数最接近的是:（ ）。

A. 5000　　B. 600　　C. 6　　D. 100

25. (3分) 当n=100时，最好情况下，与第12行的比较运算执行的次数最接近的是:（ ）。

A. 100　　B. 6　　C. 5000　　D. 600

26. (3分) 当 $n = 10$ 时，若 b 数组满足，对任意 $0 <= i < n$, 都有 $b[i] = i + 1$, 那么输出最大为（ ）。

A. 386　　B. 383　　C. 384　　D. 385

27. (4分) 当n=100时，若b数组满足，对任意 $0 <= i < n$ ，都有 $b[i] = 1$ ，那么输出最小为（ ）。

A. 582　　B. 580　　C. 579　　D. 581

(3)

```
01 #include<bits/stdc++.h>
02 using namespace std;
03 #define MOD 19260817
04 #define MAXN 1005
05 long long A[MAXN][MAXN]={0},sum[MAXN][MAXN]={0};
06 int n, m, q;
07 int main(){
08     A[1][1]=A[1][0]=1;
09     for(int i=2;i <= 1000; i++) {
10         A[i][0]=1;
11         for(int j = 1; j <= i; j++)
12             A[i][j]=(A[i-1][j]+A[i-1][j-1])%MOD;
13     }
14     for(int i =1;i <= 1000; i++)
15         for(int j=1;j<=1000;j++)
16             sum[i][j]=(sum[i-1][j]+sum[i][j-1]
17             -sum[i-1][j-1]+A[i][j]+MOD)%MOD;
18             int q;
19     cin >> q;
20     while(q--){
21         int n, m;
22         cin >> n >> m;
23         cout << sum[n][m] << endl;
24     }
25     return 0;
26 }
```

判断题

28. (1分) 当i<=j时，A[i][i]的值是0。（ ）

29. (2分) 当i>j时，A[i][j]的值相当于从i个不同元素中取出j个元素的排列数。（ ）

30. (2分) sum[i][j]的值(1<j<i≤1000)不小于sum[i-1][j-1]的值。（ ）

31. (2分) 若将第12行改为“A[i][j]=(A[i-1][j]+A[i-1][j-1]+MOD)%MOD;”，程序的运行的结果不变。（ ）

单选题

32. (4分) A[i][j] (1≤i≤10, 1≤j≤10)的所有元素中，最大值是（ ）。

A. 126　　B. 276　　C. 252　　D. 210

33. (3分) 若输入数据为 1/5 3(其中“/”为换行符)，则输出为（ ）。

A. 10　　B. 35　　C. 50　　D. 24

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **29** 分，各小题分值见题目）

(1)

（切割绳子）有 n条绳子，每条绳子的长度已知且均为正整数。绳子可以以任意正整数长度切割，但不可以连接。现在要从这些绳子中切割出 m 条长度相同的绳段，求绳段的最大长度是多少。 输入：第一行是一个不超过 100 的正整数 n，第二行是 n 个不超过10^6的正整数，表示每条绳子的长度，第三行是一个不超过10^8的正整数m。 输出：绳段的最大长度，若无法切割，输出Failed。

```
01 #include<iostream>
02 using namespace std;
03 int n, m, i, lbound, ubound, mid, count;
04 int len[100]; // 绳子长度
05 int main()
06 {
07     cin >> n;
08     count = 0;
09     for (i = 0; i < n; i++)
10     {
11         cin >> len[i];
```

```
12          __(1)__;
13      }
14      cin >> m;
15      if (__(2)__)
16      {
17          cout << "Failed" << endl;
18          return 0;
19      }
20      lbound = 1;
21      ubound = 1000000;
22      while (__(3)__)
23      {
24          mid = __(4)__;
25          count = 0;
26          for (i = 0; i < n; i++)
27              __(5)__;
28          if (count < m)
29              ubound = mid - 1;
30          else
31              lbound = mid;
32      }
33      cout << lbound << endl;
34      return 0;
35  }
```

34. (2.5分) (1)处应填：_____

35. (2.5分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (3分) (5)处应填：_____

(2)

完善程序

（排列数）输入两个正整数 n ， $m(1<n<20,1<m<n)$ ，在 $1\sim n$ 中任取 m 个数，按字典序从小到大输出所有这样的排列。

例如:输入:32

输出：

```
#include <iostream>
#include <cstring>
using namespace std;
const int SIZE =25;
bool used[SIZE];
int data[SIZE];
int n,m,i,j,k;
bool flag;
int main()
{
    cin>>n>>m;
    memset(used,false,sizeof(used));
    for(i=1;i<=m;i++)
    {
        data[i]=i;
        used[i]=true;
    }
    flag=true;
    while(flag)
    {
        for(i=1;i<=m-1;i++) cout<<data[i]<<" ";
        cout<<data[m]<<endl;
        flag= __(1)__ ;
        for(i=m;i>=1;i--)
        {
            __(2)__;
            for(j=data[i]+1;j<=n;j++)
                if(!used[j])
                {
                    used[j]=true;
                    data[i]=__(3)__ ;
                    flag=true;
                    break;
                }
            if(flag)
            {
                for(k=i+1;k<=m;k++)
                    for(j=1;j<= __(4)__;j++)
                        if(!used[j])
                        {
                            data[k]=j;
```

```
42                              used[j]=true;
43                              break;
44                          }
45                  __(5)__;
46              }
47          }
48      }
49      return 0;
50  }
```

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (3分) (4)处应填：_____

43. (3分) (5)处应填：_____