CSP-J 初赛模拟卷（第 16 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．在 PC 机中，PENTIUM(奔腾)、酷睿、赛扬等是指（ ）。

A．生产厂家名称　　B．硬盘的型号　　C．CPU 的型号　　D．显示器的型号

2．1TB代表的字节数是（ ）。

A．2的10次方　　B．2的20次方　　C．2的30次方　　D．2的40次方

3．如果一个栈初始时为空，且当前栈中的元素从栈底到栈顶依次为 a,b,c ，另有元素 d 已经出栈，则可能的入栈顺序是（ ）。

A．a,d,c,b　　B．b,a,c,d　　C．a,c,b,d　　D．d,a,b,c

4．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．Y X Z　　B．Z X Y　　C．X Y Z　　D．X Z Y

5．一棵具有5层的满二叉树中结点数为（ ）。

A．31　　B．32　　C．33　　D．16

6．给定含有 n 个互不相同元素的单峰数组 L，满足先严格递增、后严格递减，峰顶位于数组内部。用下面的二分算法寻找峰顶，请将 a、b、c 三行代码依次填入空白处。正确的填空顺序是（ ）。

```
a. return Search(k + 1, r)
b. return Search(l, k - 1)
c. return L[k]

Search(l, r):
    if l == r then return L[l]
    k = floor((l + r) / 2)
    if L[k] > L[k - 1] and L[k] > L[k + 1]: ______
    else if L[k] < L[k + 1]: ______
    else: ______

调用 Search(2, n - 1)
```

A．c,a,b　　B．c,b,a　　C．a,b,c　　D．b,a,c

7．表达式 $a*(b+c)*d$ 的后缀形式是（ ）。

A．$abcd*+*$　　B．$abc+*d*$　　C．$a*bc+*d$　　D．$b+c*a*d$

8．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

9．假设一棵二叉树的后序遍历序列为 DGJHEBIFCA，中序遍历序列为DBGEHJACIF,则其前序遍历序列为（ ）。

A．ABCDEFGHIJ　　B．ABDEGHJCFI　　C．ABDEGJHCFI　　D．ABDEGHJFIC

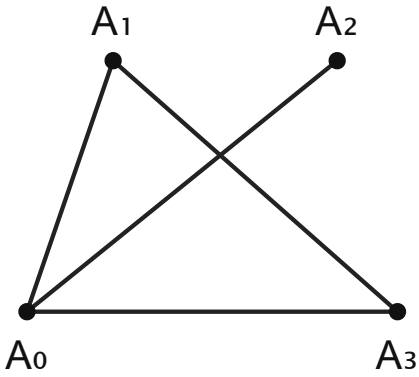
10．若有如下程序段，其中 s 、 a 、 b 、 c 均已定义为整型变量，且 a 、 c 均已赋值（c 大于0）。

$$\begin{aligned}&\mathrm{s=a;}\\&\mathrm{for\ (b=1;b<=c;\ b{+}{+})}\\&\mathrm{s=s+1;}\end{aligned}$$

则与上述程序段修改 s 值的功能等价的赋值语句是（ ）。

A．$s=a+b$　　B．$s=a+c$　　C．$s=s+c$　　D．$s=b+c$

11．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

12．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

13．周末小明和爸爸妈妈三个人一起想动手做三道菜。小明负责洗菜、爸爸负责切菜、妈妈负责炒菜。假设做每道菜的顺序都是:先洗菜10分钟，然后切菜10分钟，最后炒菜10分钟。那么做一道菜需要30分钟。注意:两道不同的菜的相同步骤不可以同时进行。例如第一道菜和第二道的菜不能同时洗，也不能同时切。那么做完三道菜的最短时间需要（　）分钟。

A．90　　B．60　　C．50　　D．40

14．有向图中每个顶点的度等于该顶点的（　）。

A．入度　B．出度　C．入度和出度之和　D．入度和出度之差

15. 干支纪年法是中国传统的纪年方法，由10个天干和12个地支组合成60个天干地支。由公历年份可以根据以下公式和表格换算出对应的天干地支。 天干 = (公历年份) 除以10所得余数 地支 = (公历年份) 除以12所得余数 | 天干 | 甲 | 乙 | 丙 | 丁 | 戊 | 己 | 庚 | 辛 | 壬 | 癸 | | | | – | – | – | – | – | – | – | – | – | – | – | | | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | | | | 地支 | 子 | 丑 | 寅 | 卯 | 辰 | 巳 | 午 | 未 | 申 | 酉 | 戌 | 亥 | | | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 0 | 1 | 2 | 3 | 例如，今年是2020年，2020除以10余数为0，查表为“庚”；2020除以12余数为4，查表为“子”所以今年是庚子年。请问1949年的天干地支息{。

A. 己酉　　B. 己亥　　C. 己丑　　D. 己卯

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
04 int n;
05 int d[1000];
06
07 int main() {
08     cin >> n;
09     for (int i = 0; i < n; ++i)
10         cin >> d[i];
11     int ans = -1;
12     for (int i = 0; i < n; ++i)
13         for (int j = 0; j < n; ++j)
14             if (d[i] < d[j])
15                 ans = max(ans, d[i] + d[j] - (d[i] & d[j]));
16     cout << ans;
17     return 0;
18 }
```

假设输入的n和d[i]都是不超过10000的正整数，完成下面的判断题和单选题:

判断题

16. (1.5分) n必须小于1000，否则程序可能会发生运行错误。（　）

17. (1.5分) 输出一定大于等于0。（ ）

18. (1.5分) 若将第13行的“j = 0”改为“j = i + 1”，程序输出可能会改变。（ ）

19. (1.5分) 将第14行的“d[i] < d[j]”改为“d[i] != d[j]”，程序输出不会改变。（ ）

单选题

20. (3分) 若输入 n 为100，且输出为127，则输入的d[i]中不可能有（ ）。

A. 127　　B. 126　　C. 128　　D. 125

21. (3分) 若输出的数大于0，则下面说法正确的是（ ）。

A. 若输出为偶数，则输入的d[i]中最多有两个偶数
B. 若输出为奇数，则输入的d[i]中至少有两个奇数
C. 若输出为偶数，则输入的d[i]中至少有两个偶数
D. 若输出为奇数，则输入的d[i]中最多有两个奇数

(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;
```

```
    for (int i = 0; i < n; ++i)
        cin >> a[i];
    for (int i = 0; i < n; ++i)
        cin >> b[i];
    cout << f(0, n - 1, 1) << endl;
    return 0;
}
```

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)

```
#include<bits/stdc++.h>
using namespace std;
#define MOD 19260817
#define MAXN 1005
long long A[MAXN][MAXN]={0},sum[MAXN][MAXN]={0};
int n, m, q;
int main(){
    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 **28** 分，各小题分值见题目）

(1)

完善程序:

（读入整数）请完善下面的程序，使得程序能够读入两个int范围内的整数，并将这两个整数分别输出，每行一个。输入的整数之间和前后只会出现空格或者回车。输入数据保证合法。

例如:

输入:

```
123 -789
```

输出:

```
123
-789
```

```
#include <iostream>
using namespace std;
int readint(){
    int num = 0;          // 存储读取到的整数
    int negative = 0;    // 负数标识
    char c;               // 存储当前读取到的字符
    c = cin.get();
    while ((c < '0' || c > '9') && c != '-')
        c = __(1)__;
    if (c == '-')
        negative = 1;
    else
        __(2)__;
    c = cin.get();
    while (__(3)__){
        __(4)__;
        c = cin.get();
    }
    if (negative == 1)
        __(5)__;
    return num;
}
```

```
int main()
{
    int a, b;
    a = readint();
    b = readint();
    cout << a << endl
         << b << endl;
    return 0;
}
```

34. (2.5分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (2.5分) (5)处应填：_____

(2)

完善程序：

(中位数 median) 给定 n(n 为奇数且小于 1000)个整数，整数的范围在 $0-m(0<m<2^{31})$ 之间，请使用二分法求这 n 个整数的中位数。所谓中位数，是指将这 n 个排序之后，排在正中间的数。

```
#include <iostream>
using namespace std;
const int MAXN = 1000;
int n, i, lbound, rbound, mid, m, count;
int x[MAXN];
int main()
{
    cin >> n >> m;
    for (i = 0; i < n; i++)
        cin >> x[i];
    lbound = 0;
    rbound = m;
    while (__(1)__)
    {
        mid = lbound + (rbound - lbound) / 2;
        __(2)__;
```

```
17          for (i = 0; i < n; i++)
18              if (__(3)__)
19                  __(4)__;
20          if (count > n / 2)
21              lbound = mid + 1;
22          else
23              __(5)__;
24          cout << mid << " " << lbound << " " << rbound << " " <<
    count << endl;
25      }
26      cout << rbound << endl;
27      return (0);
28  }
```

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. (2分) (5)处应填：_____