CSP-J 初赛模拟卷（第 07 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．计算机应用的最早领域（　）

A．数值计算　　B．人工智能　　C．机器人　　D．过程控制

2．二进制数11 1011 1001 0111和01 0110 1110 1011进行逻辑与运算的结果是（　）。

A．01 0010 1000 1011　　B．01 0010 1001 0011

C．01 0010 1000 0001　　D．01 0010 1000 0011

3．在内存储器中每个存储单元都被赋予一个唯一的序号，称为（　）

A．地址　　B．序号　　C．下标　　D．编号

4．如果开始时计算机处于小写输入状态，现在有一只小老鼠反复按照CapsLock、 字母键A、字母键 S、字母键D、字母键 F 的顺序循环按键，即 CapsLock、A、 S、D、F、CapsLock、A、S、D、F、……，屏幕上输出的第 81 个字符是字母（　）。

A．A　　B．S　　C．D　　D．a

5．分辨率为 800x600、16 位色的位图，存储图像信息所需的空间为（　）。

A．937.5KB　　B．4218.75KB　　C．4320KB　　D．2880KB

6．以比较作为基本运算，在N个数中找出最大数，最坏情况下所需要的最少的比较次数为（　）。

A．N^2　　B．N　　C．N-1　　D．N+1

7．以下是 32 位机器和 64 位机器的区别是（　）。

A．显示器不同　　B．硬盘大小不同　　C．寻址空间不同　　D．输入法不同

8．由四个没有区别的点构成的简单无向连通图的个数是（　）。

A．6　　B．7　　C．8　　D．9

9. 言泡排序算法的伪代码如下：

```
输入：数组L,n≥k。输出：按非递减顺序排序的L。
算法 BubbleSort:
    1. FLAG ← n //标记被交换的最后元素位置
    2. while FLAG > 1 do
    3. k ← FLAG -1
    4. FLAG ← 1
    5. for j=1 to k do
    6. if L(j) > L(j+1) then do
    7. L(j) ← L(j+1)
    8. FLAG ← j
```

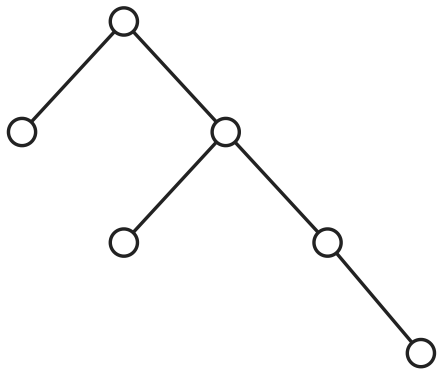
对 n 个数用以上音泡排序算法进行排序，最少需要比较多少次？（ ）。

A. n^2　　B. $n-2$　　C. $n-1$　　D. n

10. 表达式a*(b+c)*d 的后缀表达式为（ ），其中“*”和“+”是运算符。

A. **a+bcd　　B. abc+*d*　　C. abc+d**　　D. *a*+bcd

11. 一棵二叉例如右图所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点（根结点的下标为1，若某结点的下标为i，则其在孩子位于下标2i处、右孩子位于下标2i+1处），则该数组的最大下标至少为（ ）。



A. 6　　B. 10　　C. 15　　D. 12

12. 十进制小数 13.375 对应的二进制数是（ ）。

A. 1101.011　　B. 1011.011　　C. 1101.101　　D. 1010.01

13. 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

14. 为了统计一个非负整数的二进制形式中 1 的个数，代码如下： int CountBit(int x) { int ret = 0; while (x) { ret++; _____; } return ret; } 则空格内要填入的语句是（ ）。

A. x >>= 1　　B. x &= x - 1　　C. x |= x >> 1　　D. x <<= 1

15. 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

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 39 分，各小题分值见题目）

(1)

```
01 #include <cstdio>
02 using namespace std;
03 int n;
04 int a[100];
05
06 int main() {
07     scanf("%d", &n);
08     for (int i = 1; i <= n; ++i)
09         scanf("%d", &a[i]);
10     int ans = 1;
11     for (int i = 1; i <= n; ++i) {
12         if (i > 1 && a[i] < a[i - 1])
13             ans = i;
14         while (ans < n && a[i] >= a[ans + 1])
15             ++ans;
```

```
16          printf("%d\n", ans);
17      }
18      return 0;
19  }
```

判断题

16. (1分) 第16行输出ans时，ans的值一定大于i。（ ）

17. (1分) 程序输出的ans小于等于n。（ ）

18. (1.5分) 若将第12行的 $<$ 改为 $!=$ ，程序输出的结果不会改变。（ ）

19. (1.5分) 当程序执行到第16行时，若ans-i>2,则 $a[i+1] \leq a[i]$ 。（ ）

单选题

20. (3分) 若输入的a数组是一个严格单调递增的数列，此程序的时间复杂度（ ）

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

21. (4分) 最坏情况下，此程序的时间复杂度是（ ）

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

(2)

```
01  #include <stdio.h>
02  #include <string.h>
03
04  char base[64];
05  char table[256];
06  char str[256];
07  char ans[256];
08
09  void init()
10  {
11      for (int i = 0; i < 26; i++) base[i] = 'A' + i;
12      for (int i = 0; i < 26; i++) base[26 + i] = 'a' + i;
13      for (int i = 0; i < 10; i++) base[52 + i] = '0' + i;
14      base[62] = '+', base[63] = '/';
15
16      for (int i = 0; i < 256; i++) table[i] = 0xff;
17      for (int i = 0; i < 64; i++) table[base[i]] = i;
18      table['='] = 0;
19  }
```

```
20
21 void decode(char *str)
22 {
23     char *ret = ans;
24     int i, len = strlen(str);
25     for (i = 0; i < len; i += 4) {
26         (*ret++) = table[str[i]] << 2 | table[str[i + 1]] >> 4;
27         if (str[i + 2] != '=')
28             (*ret++) = (table[str[i + 1]] & 0x0f) << 4 | table[str[i
   + 2]] >> 2;
29         if(str[i + 3] != '=')
30             (*ret++) = table[str[i + 2]] << 6 | table[str[i + 3]];
31     }
32 }
33
34 int main()
35 {
36     init();
37     printf("%d\n", (int)table[0]);
38
39     scanf("%s", str);
40     decode(str);
41     printf("%s\n", ans);
42     return 0;
43 }
```

假设 char 是 8 位有符号类型，输入字符串长度为 4 的倍数，且输入不超过数组范围。

判断题

22. (1.5分) 输出的第二行一定是由小写字母、大写字母、数字和 +、/、= 构成的字符串。（ ）

23. (1.5分) 可能存在输入不同，但输出的第二行相同的情形。（ ）

24. (1.5分) 输出的第一行为 -1。（ ）

单选题

25. (3分) 设输入字符串长度为 n，decode 函数的时间复杂度为（ ）。

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

26. (3分) 当输入为 Y3Nx 时，输出的第二行为（ ）。

A. csp　　B. csq　　C. CSP　　D. Csp

27. (3.5分) 当输入为 Y2NmIDIwMjE= 时，输出的第二行为（ ）。

A. ccf2021　　B. ccf2022　　C. ccf 2021　　D. ccf 2022

(3)

```
#include<cstdio>
#include<cstring>
using namespace std;
int n, m;
int f[101][101];
int F[101][101];
int main(){
    scanf("%d%d",&n,&m);//n的值在1到100之间
    memset(f,-1,sizeof(f));
    for(int i = 1; i <=  m; i++) {
        int u,v,w;//w的值在0到10000之间
        scanf("%d%d%d",&u,&v,&w);
        f[u][v]=f[v][u]=w;
    }
    for(int k = 1; k <= n; k++)
        for(int i=1;i <=n; i++)
            for(int j=1;j<= n; j++)
                if(f[i][k]!=-1&&f[k][j]!=-1)
                    if(f[i][j]==-1||f[i][j]>f[k][j]+f[i][k])
                        f[i][j]=f[i][k]+f[k][j];
    int ans=2147483647;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n; j++){
            for(int x=1;x<=n; x++)
                for(int y=1;y<=n; y++)
                    F[x][y]=f[x][y];
            F[i][j]=F[j][i]=0;
            for(int x=1;x<= n; x++)
                for(int y=1;y<=n; y++)
                    if(F[x][y]==-1||F[x][y]>F[x][i]+F[i][y])
                        F[x][y]=F[x][i]+F[i][y];
            for(int x=1;x<=n; x++)
                for(int y=1;y<=n; y++)
                    if(F[x][y]==-1||F[x][y]>F[x][j]+F[j][y])
                        F[x][y]=F[x][j]+F[j][y];
```

```
36              int res=0;
37              for(int x=1;x<=n; x++)
38                  for(int y=1;y<x; y++)
39                      res +=F[x][y];
40              ans = res < ans ? res : ans;
41          }
42      printf("%d\n",ans);
43      return 0;
44  }
```

- 判断题

判断题

28. (1分) 14到16行，将外层到内层的循环变量依次调整为i、j、k，程序的运行的结果不变。（ ）

29. (2分) 这个程序的时间复杂度和m无关。（ ）

30. (2分) 20行的ans如果初始化为10^7时，可能无法得到正确结果。（ ）

31. (2分) 若将第27到30行的部分和31到34行的两个部分互换，程序的运行的结果不变。（ ）

单选题

32. (3分) 若输入数据为4 5/1 2 3/1 3 6/2 3 4/2 4 7/3 4 2(其中“/”为换行符)，则输出为（ ）。

A. 14　　B. 18　　C. 21　　D. 28

33. (3分) 这个程序使用了（ ）算法。

A. Floyd　　B. Dijkstra　　C. Prim　　D. Kruskal

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)

（计数排序）计数排序是一个广泛使用的排序方法。下面的程序使用双关键字计数排序，将 n 对10000以内的整数，从小到大排序。

例如有三对整数(3,4)、(2,4)、(3,3),那么排序之后应该是(2,4)、(3,3)、(3,4)。

输入第一行为n,接下来n行，第i行有两个数 $a[i]$ 和 $b[i]$ ，分别表示第i对整数的第一关键字和第二关键字。

从小到大排序后输出。

数据范围 $1 < n < 10^7, 1 < a[x], b[x] < 10^4$。提示:应先对第二关键字排序，再对第一关键字排序。数组ord[]存储第二关键字排序的结果，数组res[]存储双关键字排序的结果。试补全程序。

```
01 #include <cstdio>
02 #include <cstring>
03 using namespace std;
04 const int maxn = 10000000;
05 const int maxs = 10000;
06
07 int n;
08 unsigned a[maxn], b[maxn],res[maxn], ord[maxn];
09 unsigned cnt[maxs + 1];
10 int main() {
11     scanf("%d", &n);
12     for (int i = 0; i < n; ++i)
13         scanf("%u%u", &a[i], &b[i]);
14     memset(cnt, 0, sizeof(cnt));
15     for (int i = 0; i < n; ++i)
16         __(1)__; // 利用 cnt 数组统计数量
17     for (int i = 0; i < maxs; ++i)
18         cnt[i + 1] += cnt[i];
19     for (int i = 0; i < n; ++i)
20         __(2)__; // 记录初步排序结果
21     memset(cnt, 0, sizeof(cnt));
22     for (int i = 0; i < n; ++i)
23         __(3)__; // 利用 cnt 数组统计数量
24     for (int i = 0; i < maxs; ++i)
25         cnt[i + 1] += cnt[i];
26     for (int i = n - 1; i >= 0; --i)
27         __(4)__; // 记录最终排序结果
28     for (int i = 0; i < n; i++)
29         printf("%u %u\n", __(5)__);
30
31     return 0;
32 }
```

34. (3分) (1)处应填

A. ++cnt [i]　　B. ++cnt[b[i]]
C. ++cnt[a[i] * maxs + b[i]]　　D. ++cnt[a[i]]

35. (3分) (2)处应填

A. ord[--cnt[a[i]]] = i　　B. ord[--cnt[b[i]]] = a[i]
C. ord[--cnt[a[i]]] = b[i]　　D. ord[--cnt[b[i]]] = i

36. (3分) (3)处应填

A. ++cnt[b[i]]　　B. ++cnt[a[i] * maxs + b[i]]
C. ++cnt[a[i]]　　D. ++cnt[i]

37. (3分) (4)处应填

A. res[--cnt[a[ord[i]]]] = ord[i]　　B. res[--cnt[b[ord[i]]]] = ord[i]
C. res[--cnt[b[i]]] = ord[i]　　D. res[--cnt[a[i]]] = ord[i]

38. (3分) (5)处应填

A. a[i], b[i]　　B. a[res[i]], b[res[i]]
C. a[ord[res[i]]], b[ord[res[i]]]　　D. a[res[ord[i]]], b[res[ord[i]]]

(2)

完善程序

（郊游活动）有n名同学参加学校组织的郊游活动，已知学校给这n名同学的郊游总结题为A元，与此同时第i位同学自己携带了M元。为了方便郊游，活动地点提供 $B(\geq n)$ 钢游行车供人租用，租用第j辆自行车的价格为C1元，每位同学可以使用自己携带的钱或者学校的郊游经验，为了方便账务管理，每位同学只能为自己租用自行车，且不会借钱给他人，他们供图道最多有多少位同学能够租用到自行车。

本题采用二分法。对于区间 $[l,r]$ ，我们取中间点 mid 并判断租用到自行车的人数能否达到mid。判断的过程是利用贪心算法实现的。

```
01 #include <iostream>
02 using namespace std;
03 #define MAXN 1000000
04 int n, B, A, M[MAXN], C[MAXN], l, r, ans, mid;
05 bool check(int nn) {
06     int count = 0, i, j;
07     i = __(1)__;
```

```
    j = 1;
    while (i <= n) {
        if(__(2)__)
            count += C[j] - M[i];
        i++;
        j++;
    }
    return __(3)__;
}
void sort(int a[], int l, int r) {
    int i = l, j = r, x = a[(l + r) / 2], y;
    while (i <= j) {
        while (a[i] < x) i++;
        while (a[j] > x) j--;
        if (i <= j) {
            y = a[i]; a[i] = a[j]; a[j] = y;
            i++; j--;
        }
    }
    if (i < r) sort(a, i, r);
    if (l < j) sort(a, l, j);
}
int main() {
    int i;
    cin >> n >> B >> A;
    for (i = 1; i <= n; i++)
        cin >> M[i];
    for (i = 1; i <= B; i++)
        cin >> C[i];
    sort(M, 1, n);
    sort(C, 1, B);
    l = 0;
    r = n;
    while (l <= r) {
        mid = (l + r) / 2;
        if(__(4)__){
            ans = mid;
            l = mid + 1;
        }else
            r = __(5)__;
```

```
    }
    cout << ans << endl;
    return 0;
}
```

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (2.5分) (4)处应填：_____

43. (2.5分) (5)处应填：_____