CSP-J 初赛模拟卷（第 13 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．生物特征识别，是利用人体本身的生物特征进行身份认证的一种技术。目前，指纹识别、虹膜识别、人脸识别等技术已广泛应用于政府、银行、安全防卫等领域。以下不属于生物特征识别技术及其应用的是（ ）。

A．指静脉验证　　B．步态验证　　C．ATM机密码验证　　D．声音验证

2．一个正整数在二进制下有100位，则它在十六进制下有（ ）位。

A．7　　B．13　　C．25　　D．不能确定

3．以下哪个是面向对象的高级语言（ ）。

A．汇编语言　　B．C++　　C．Fortran　　D．Basic

4．Wi-Fi 主要用于构建哪类无线网络？（ ）

A．无线广域网　　B．无线城域网　　C．无线局域网　　D．无线路由器

5．以下逻辑表达式的值恒为真的是（ ）。

A．$P \vee (\neg P \wedge Q) \vee (\neg P \wedge \neg Q)$　　B．$Q \vee (\neg P \wedge Q) \vee (P \wedge \neg Q)$

C．$P \vee Q \vee (P \wedge \neg Q) \vee (\neg P \wedge Q)$　　D．$P \vee \neg Q \vee (P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$

6．把 64 位非零浮点数强制转换成32 位浮点数后，不可能（ ）。

A．大于原数　　B．小于原数　　C．等于原数　　D．与原数符号相反

7．前缀表达式 + 3 * 2 + 5 12的值是（ ）。

A．23　　B．25　　C．37　　D．65

8．计算机界的最高奖是（ ）。

A．菲尔兹奖　　B．诺贝尔奖　　C．图灵奖　　D．普利策奖

9．甲、乙、丙三位同学选修课程，从 4 门课程中，甲选修 2 门，乙、丙各选修3门，则不同的选修方案共有（ ）种。

A．36　　B．48　　C．96　　D．192

10. 如果一棵二叉树的中序遍历是 BAC，那么它的先序遍历不可能是（ ）。

A. ABC　　B. CBA　　C. ACB　　D. BAC

11. 将（2,6,10,17）分别存储到某个地址区间为0~10的哈希表中，如果哈希函数 $h(x) = (\ )$，将不会产生冲突，其中 $a \mod b$ 表示 a 除以 b 的余数。

A. $x \mod 11$　　B. $x^2 \mod 11$

C. $(2x) \mod 11$　　D. $\lfloor sqrt(x) \rfloor \mod 11$，其中 $\lfloor sqrt(x) \rfloor$ 表示 $\sqrt{x}$ 下取整

12. 向一个栈顶指针为 hs 的链式栈中插入一个指针 s 指向的结点时，应执行（ ）。

A. hs->next = s;　　B. s->next = hs; hs = s;

C. s->next = hs->next; hs->next = s;　　D. s->next = hs; hs = hs->next;

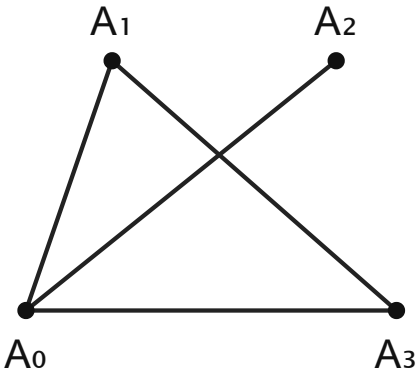
13. 使用冒泡排序对序列进行升序排列，每执行一次交换操作系统将会减少 1 个逆序对，因此序列 5，4，3，2，1 需要执行（ ）次操作，才能完成冒泡排序。

A. 0　　B. 5　　C. 10　　D. 15

14. 设 A 和 B 是两个长为 n 的有序数组，现在需要将 A 和 B 合并成一个排好序的数组，任何以元素比较作为基本运算的归并算法在最坏情况下至少要做（ ）次比较。

A. n^2　　B. nlogn　　C. 2n　　D. 2n - 1

15. 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

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 **37** 分，各小题分值见题目）

(1)

```
01 #include <cstdio>
02 #include <cstring>
03 using namespace std;
04 char st[100001];
05 int main() {
06     scanf("%s", st);
07     int n = strlen(st);
08     for (int i = 1; i <= n; ++i) {
09         if (n % i == 0) {
10             char c = st[i - 1];
11             if (c >= 'a')
12                 st[i - 1] = c - 'a' + 'A';
13         }
14     }
15     printf("%s", st);
16     return 0;
17 }
```

判断题

16. (1.5分) 输入的字符串只能由小写字母或大写字母组成。（ ）

17. (1.5分) 若将第 8 行的 `i = 1` 改为 `i = 0`，程序运行时会发生错误。（ ）

18. (1.5分) 若将第 8 行的 `i <= n` 改为 `i * i <= n`，程序运行结果不会改变。（ ）

19. (1.5分) 若输入的字符串全部由大写字母组成，那么输出的字符串就跟输入的字符串一样。（ ）

单选题

20. (3分) 若输入的字符串长度为 18，那么输入的字符串跟输出的字符串相比，至多有（ ）个字符不同。

A. 18　　B. 6　　C. 10　　D. 1

21. (3分) 若输入的字符串长度为（ ），那么输入的字符串跟输出的字符串相比，至多有 36 个字符不同。

A. 36　　B. 100000　　C. 1　　D. 128

(2)

```
01 #include<iostream>
02 using namespace std;
03 int n,m,i,j,p,k;
04 int a[100],b[100];
05 int main()
06 {
07     cin>>n>>m;
08     a[0]=n;i=0;p=0;k=1;
09     do{
10         for(j=0;j<i;j++)
11             if(a[i]==a[j])
12             {
13                 p=1;k=j;break;
14             }
15         if(p) break;
16         b[i]=a[i]/m;
17         a[i+1]=a[i]%m*10;
18         i++;
19     }while(a[i]!=0);
20     cout<<b[0]<<".";
21     for(j=1;j<k;j++) cout<<b[j];
22     if(p) cout<<"(";
23     for(j=k;j<i;j++) cout<<b[j];
24     if(p) cout<<") ";
25     cout<<endl;
26     return 0;
27 }
```

判断题

22. (1.5分) 程序输入的n和m不能相等。（ ）

23. (1.5分) 程序输入的m不能等于0。（ ）

24. (1.5分) 第9~19行的do…while循环一共有2个出口。（ ）

25. (1.5分) 数组a和b中的数值都小于或等于n。（ ）

单选题

26. (3分) 若输入数据为“11 8”，则输出结果为（ ）

A. 0.375　　B. 0.(375)　　C. 1.375　　D. 1.(375)

27. (3分) 若输人数据为“5 13”，则输出结果为（ ）。

A. 0.386514　　B. 0.(386514)　　C. 0.384615　　D. 0.(384615)

(3)

```
01 #include <iostream>
02 #include <cstdlib>
03 using namespace std;
04
05 int n;
06 int d[10000];
07
08 int find(int L, int R, int k) { if (L == R) return d[L];
09     int x = rand() % (R - L + 1) + L;
10     swap(d[L], d[x]);
11     int a = L + 1, b = R;
12     while (a < b) {
13         while (a < b && d[a] < d[L])
14             ++a;
15         while (a < b && d[b] >= d[L])
16             --b;
17         swap(d[a], d[b]);
18     }
19     if (d[a] < d[L])
20         ++a;
21     if (a - L == k)
22         return d[L];
23     if (a - L < k)
24         return find(a, R, k - (a - L));
25     return find(L + 1, a - 1, k);
26 }
27
28 int main() {
29     int k;
30     cin >> n;
31     cin >> k;
32     for (int i = 0; i < n; ++i)
```

```
33          cin >> d[i];
34      cout << find(0, n - 1, k);
35      return 0;
36  }
```

假设 $1 \le k \le n \le 10000$，输入数组元素为不超过 10000 的正整数，且 rand() 产生均匀随机数。复杂度选项均指渐进数量级。

判断题

28. (1.5分) 第 9 行中 x 的数值范围为 $[L+1, R]$。（ ）

29. (1.5分) 将第 19 行中的 d[a] 改为 d[b]，程序不会发生运行错误。（ ）

单选题

30. (2.5分) 当输入数组严格单调递增时，此程序的平均时间复杂度为（ ）。

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

31. (2.5分) 当输入数组严格单调递减时，此程序的平均时间复杂度为（ ）。

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

32. (2.5分) 当输入数组严格单调递增时，此程序①平均时间复杂度与②最坏时间复杂度分别为（ ）。

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

33. (2.5分) 若所有数组元素相等，且 $k = n$，此程序的时间复杂度为（ ）。

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

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)

```
01  const int SIZE = 100;
02  int a[SIZE], n;
```

它记录着一个长度为n的序列 $a[1], a[2], \cdots, a[n]$ 。 现在需要一个函数，以整数 $p(1 \le p \le n)$ 为参数，实现如下功能:将序列a的前p个数与后n-p个数对调，且不改变这p个数（或n-p个数）之间的相对位置。例如，长度为5的序列1,2,3,4,5，当p=2时重排结果为

3,4,5,1,2。 有一种朴素的算法可以实现这一需求，其时间复杂度为 $O(n)$ 、空间复杂度为 $O(n)$ ：

```
01 void swap1( int p )
02 {
03 int i, j, b[SIZE];
04 for ( i = 1; i <= p; i++ )
05 b[__(1)__] = a[i];
06 for ( i = p + 1; i <= n; i++ )
07 b[i - p] = __(2)__;
08 for ( i = 1; i <= __(3)__; i++ )
09 a[i] = b[i];
10 }
```

我们也可以用时间换空间，使用时间复杂度为 $O(n^2)$ 、空间复杂度为 $O(1)$ 的算法：

```
01 void swap2(int p)
02 {
03     int i, j, temp;
04     for ( i = p + 1; i <= n; i++ )
05     {
06         temp = a[i];
07         for ( j = i; j >= __(4)__; j-- )
08             a[j] = a[j - 1];
09         __(5)__ = temp;
10     }
11 }
```

34. (3分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (2分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (3分) (5)处应填：_____

(2)

（交朋友）根据社会学研究表明，人们都喜欢找和自己身高相近的人做朋友。现在有n名身高两两不相同的同学依次走入教室，调查人员想预测每个人在走入教室的瞬间最想和已经进入教室的哪个人做朋友。当有两名同学和这名同学的身高差一样时，这名同学会更想和高的那个人做朋友。比如一名身高为1.80米的同学进入教室时，有一名身高为1.79米的同学和一名

身高为1.81米的同学在教室里，那么这名身高为1.80米的同学会更想和身高为1.81米的同学做朋友。对于第一个走入教室的同学我们不做预测。由于我们知道所有人的身高和走进教室的次序，所以我们可以采用离线的做法来解决这样的问题，我们用排序加链表的方式帮助每一个人找到在他之前进入教室的并且和他身高最相近的人。（第一空2分，其余3分）

```
01 #include <iostream>
02 using namespace std;
03 #define MAXN 200000
04 #define infinity 2147483647
05 int answer[MAXN], height[MAXN], previous[MAXN], next_pos[MAXN];
06 int order[MAXN];
07 int n;
08 void sort(int l, int r)
09 {
10     int x = height[order[(l + r) / 2]], i = l, j = r, temp;
11     while (i <= j)
12     {
13         while (height[order[i]] < x)
14             i++;
15         while (height[order[j]] > x)
16             j--;
17         if (__(1)__)
18         {
19             temp = order[i];
20             order[i] = order[j];
21             order[j] = temp;
22             i++;
23             j--;
24         }
25     }
26     if (i < r)
27         sort(i, r);
28     if (l < j)
29         sort(l, j);
30 }
31 int main()
32 {
33     cin >> n;
34     int i, higher, shorter;
35     for (i = 1; i <= n; i++)
36     {
37         cin >> height[i];
```

```
38          order[i] = i;
39      }
40      sort(1, n);
41      for (i = 1; i <= n; i++)
42      {
43          previous[order[i]] = order[i - 1];
44          __(2)__;
45      }
46      for (i = n; i >= 2; i--)
47      {
48          higher = shorter = infinity;
49          if (previous[i] != 0)
50              shorter = height[i] - height[previous[i]];
51          if (next_pos[i] != 0)
52              __(3)__;
53          if (__(4)__)
54              answer[i] = previous[i];
55          else
56              answer[i] = next_pos[i];
57          next_pos[previous[i]] = next_pos[i];
58          __(5)__;
59      }
60      for (i = 2; i <= n; i++)
61          cout << i << ":" << answer[i];
62      return 0;
63  }
```

39. (2.8分) (1)处应填：_____

40. (2.8分) (2)处应填：_____

41. (2.8分) (3)处应填：_____

42. (2.8分) (4)处应填：_____

43. (2.8分) (5)处应填：_____