

第二十四届全国青少年信息学奥林匹克联赛初赛

提高组 C++ 语言试题

竞赛时间：2018 年 10 月 13 日 14:30~16:30

选手注意：

- 试题纸共有 10 页，答题纸共有 2 页，满分 100 分。请在答题纸上作答，写在试题纸上的一律无效。
- 不得使用任何电子设备（如计算器、手机、电子词典等）或查阅任何书籍资料。

一、单项选择题（共 10 题，每题 2 分，共计 20 分；每题有且仅有一个正确选项）

1. 下列四个不同进制的数中，与其它三项数值上不相等的是（ ）。
A. $(269)_{16}$
B. $(617)_{10}$
C. $(1151)_8$
D. $(1001101011)_2$
2. 下列属于解释执行的程序设计语言是（ ）。
A. C
B. C++
C. Pascal
D. Python
3. 中国计算机学会于（ ）年创办全国青少年计算机程序设计竞赛。
A. 1983
B. 1984
C. 1985
D. 1986
4. 设根节点深度为 0，一棵深度为 h 的满 k ($k > 1$) 叉树，即除最后一层无任何子节点外，每一层上的所有结点都有 k 个子结点的树，共有（ ）个结点。
A. $(k^{h+1} - 1) / (k - 1)$
B. k^{h-1}
C. k^h
D. $(k^{h-1}) / (k - 1)$
5. 设某算法的时间复杂度函数的递推方程是 $T(n) = T(n - 1) + n$ (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. $O(n^2)$

6. 表达式 $a * d - b * c$ 的前缀形式是 ()。
- A. $a d * b c * -$
 - B. $- * a d * b c$
 - C. $a * d - b * c$
 - D. $- * * a d b c$
7. 在一条长度为 1 的线段上随机取两个点，则以这两个点为端点的线段的期望长度是 ()。
- A. $1/2$
 - B. $1/3$
 - C. $2/3$
 - D. $3/5$
8. 关于 Catalan 数 $C_n = (2n)! / (n+1)! / n!$ ，下列说法中错误的是 ()。
- A. C_n 表示有 $n+1$ 个结点的不同形态的二叉树的个数。
 - B. C_n 表示含 n 对括号的合法括号序列的个数。
 - C. C_n 表示长度为 n 的入栈序列对应的合法出栈序列个数。
 - D. C_n 表示通过连接顶点而将 $n+2$ 边的凸多边形分成三角形的方法个数。
9. 假设一台抽奖机中有红、蓝两色的球，任意时刻按下抽奖按钮，都会等概率获得红球或蓝球之一。有足够多的人每人都用这台抽奖机抽奖，假如他们的策略均为：抽中蓝球则继续抽球，抽中红球则停止。最后每个人都把自己获得的所有球放到一个大箱子里，最终大箱子里的红球与蓝球的比例接近于 ()。
- A. $1:2$
 - B. $2:1$
 - C. $1:3$
 - D. $1:1$
10. 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.  $x <=<= 1$

二、不定项选择题（共 5 题，每题 2 分，共计 10 分；每题有一个或多个正确选项，多选或少选均不得分）

1. NOIP 初赛，选手可以带入考场的有（ ）。
  - A. 笔
  - B. 橡皮
  - C. 手机（关机）
  - D. 草稿纸
2. 2-3 树是一种特殊的树，它满足两个条件：
  - （1）每个内部结点有两个或三个子结点；
  - （2）所有的叶结点到根的路径长度相同。如果一棵 2-3 树有 10 个叶结点，那么它可能有（ ）个非叶结点。
  - A. 5
  - B. 6
  - C. 7
  - D. 8
3. 下列关于最短路算法的说法正确的有（ ）。
  - A. 当图中不存在负权回路但是存在负权边时，Dijkstra 算法不一定能求出源点到所有点的最短路。
  - B. 当图中不存在负权边时，调用多次 Dijkstra 算法能求出每对顶点间最短路径。
  - C. 图中存在负权回路时，调用一次 Dijkstra 算法也一定能求出源点到所有点的最短路。
  - D. 当图中不存在负权边时，调用一次 Dijkstra 算法不能用于每对顶点间最短路径计算。
4. 下列说法中，是树的性质的有（ ）。
  - A. 无环
  - B. 任意两个结点之间有且只有一条简单路径
  - C. 有且只有一个简单环
  - D. 边的数目恰是顶点数目减 1
5. 下列关于图灵奖的说法中，正确的有（ ）。
  - A. 图灵奖是由电气和电子工程师协会（IEEE）设立的。
  - B. 目前获得该奖项的华人学者只有姚期智教授一人。
  - C. 其名称取自计算机科学的先驱、英国科学家艾伦·麦席森·图灵。

D. 它是计算机界最负盛名、最崇高的一个奖项，有“计算机界的诺贝尔奖”之称。

### 三、问题求解（共 2 题，每题 5 分，共计 10 分）

1. 甲乙丙丁四人在考虑周末要不要外出郊游。  
已知①如果周末下雨，并且乙不去，则甲一定不去；②如果乙去，则丁一定去；③如果丙去，则丁一定不去；④如果丁不去，而且甲不去，则丙一定不去。如果周末丙去了，则甲\_\_\_\_\_（去了/没去）（1 分），乙\_\_\_\_\_（去了/没去）（1 分），丁\_\_\_\_\_（去了/没去）（1 分），周末\_\_\_\_\_（下雨/没下雨）（2 分）。
2. 方程  $a*b = (a \text{ or } b) * (a \text{ and } b)$ ，在  $a, b$  都取  $[0, 31]$  中的整数时，共有\_\_\_\_\_组解。（\*表示乘法；or 表示按位或运算；and 表示按位与运算）

### 四、阅读程序写结果（共 4 题，每题 8 分，共计 32 分）

```
1. #include <cstdio>
int main() {
 int x;
 scanf("%d", &x);
 int res = 0;
 for (int i = 0; i < x; ++i) {
 if (i * i % x == 1) {
 ++res;
 }
 }
 printf("%d", res);
 return 0;
}
```

输入：15

输出：\_\_\_\_\_

```
2. #include <cstdio>
int n, d[100];
bool v[100];
int main() {
 scanf("%d", &n);
 for (int i = 0; i < n; ++i) {
 scanf("%d", &d[i]);
 v[d[i]] = false;
 }
}
```

```

 }
 int cnt = 0;
 for (int i = 0; i < n; ++i) {
 if (!v[i]) {
 for (int j = i; !v[j]; j = d[j]) {
 v[j] = true;
 }
 ++cnt;
 }
 }
 printf("%d\n", cnt);
 return 0;
}

```

输入: 10 7 1 4 3 2 5 9 8 0 6

输出: \_\_\_\_\_

3. #include <iostream>  
using namespace std;  
string s;  
long long magic(int l, int r) {  
 long long ans = 0;  
 for (int i = l; i <= r; ++i) {  
 ans = ans \* 4 + s[i] - 'a' + 1;  
 }  
 return ans;  
}  
int main() {  
 cin >> s;  
 int len = s.length();  
 int ans = 0;  
 for (int l1 = 0; l1 < len; ++l1) {  
 for (int r1 = l1; r1 < len; ++r1) {  
 bool bo = true;  
 for (int l2 = 0; l2 < len; ++l2) {  
 for (int r2 = l2; r2 < len; ++r2) {  
 if (magic(l1, r1) == magic(l2, r2) && (l1 !=  
12 || r1 != r2)) {  
 bo = false;  
 }  
 }  
 }  
 if (bo) {

```

 ans += 1;
 }
}
cout << ans << endl;
return 0;
}

```

输入: abacaba

输出: \_\_\_\_\_

4. #include <cstdio>  
using namespace std;  
const int N = 110;  
bool isUse[N];  
int n, t;  
int a[N], b[N];  
bool isSmall() {  
 for (int i = 1; i <= n; ++i)  
 if (a[i] != b[i]) return a[i] < b[i];  
 return false;  
}  
bool getPermutation(int pos) {  
 if (pos > n) {  
 return isSmall();  
 }  
 for (int i = 1; i <= n; ++i) {  
 if (!isUse[i]) {  
 b[pos] = i; isUse[i] = true;  
 if (getPermutation(pos + 1)) {  
 return true;  
 }  
 isUse[i] = false;  
 }  
 }  
 return false;  
}  
void getNext() {  
 for (int i = 1; i <= n; ++i) {  
 isUse[i] = false;  
 }  
 getPermutation(1);  
 for (int i = 1; i <= n; ++i) {

```

 a[i] = b[i];
 }
}
int main() {
 scanf("%d%d", &n, &t);
 for (int i = 1; i <= n; ++i) {
 scanf("%d", &a[i]);
 }
 for (int i = 1; i <= t; ++i) {
 getNext();
 }
 for (int i = 1; i <= n; ++i) {
 printf("%d", a[i]);
 if (i == n) putchar('\n'); else putchar(' ');
 }
 return 0;
}

```

输入 1: 6 10 1 6 4 5 3 2

输出 1: \_\_\_\_\_ (3 分)

输入 2: 6 200 1 5 3 4 2 6

输出 2: \_\_\_\_\_ (5 分)

## 五、完善程序（共 2 题，每题 14 分，共计 28 分）

1. 对于一个 1 到  $n$  的排列  $P$ （即 1 到  $n$  中每一个数在  $P$  中出现了恰好一次），令  $q_i$  为第  $i$  个位置之后第一个比  $P_i$  值更大的位置，如果不存在这样的位置，则  $q_i = n + 1$ 。举例来说，如果  $n = 5$  且  $P$  为 1 5 4 2 3，则  $q$  为 2 6 6 5 6。

下列程序读入了排列  $P$ ，使用双向链表求解了答案。试补全程序。（第二空 2 分，其余 3 分）

数据范围  $1 \leq n \leq 10^5$ 。

```

#include <iostream>
using namespace std;
const int N = 100010;
int n;
int L[N], R[N], a[N];
int main() {
 cin >> n;
 for (int i = 1; i <= n; ++i) {
 int x;

```

```

 cin >> x;
 (1);
 }
 for (int i = 1; i <= n; ++i) {
 R[i] = (2);
 L[i] = i - 1;
 }
 for (int i = 1; i <= n; ++i) {
 L[(3)] = L[a[i]];
 R[L[a[i]]] = R[(4)];
 }
 for (int i = 1; i <= n; ++i) {
 cout << (5) << " ";
 }
 cout << endl;
 return 0;
}

```

2. 一只小猪要买  $N$  件物品 ( $N$  不超过 1000)。

它要买的所有物品在两家商店里都有卖。第  $i$  件物品在第一家商店的价格是  $a[i]$ ，在第二家商店的价格是  $b[i]$ ，两个价格都不小于 0 且不超过 10000。如果在第一家商店买的物品的总额不少于 50000，那么在第一家店买的物品都可以打 95 折(价格变为原来的 0.95 倍)。

求小猪买齐所有物品所需最少的总额。

输入：第一行一个数  $N$ 。接下来  $N$  行，每行两个数。第  $i$  行的两个数分别代表  $a[i]$ ， $b[i]$ 。

输出：输出一行一个数，表示最少需要的总额，保留两位小数。

试补全程序。(第一空 2 分，其余 3 分)

```

#include <cstdio>
#include <algorithm>
using namespace std;

const int Inf = 1000000000;
const int threshold = 50000;
const int maxn = 1000;

int n, a[maxn], b[maxn];
bool put_a[maxn];
int total_a, total_b;

```



```

double ans;
int f[threshold];

int main() {
 scanf("%d", &n);
 total_a = total_b = 0;
 for (int i = 0; i < n; ++i) {
 scanf("%d%d", a + i, b + i);
 if (a[i] <= b[i]) total_a += a[i];
 else total_b += b[i];
 }
 ans = total_a + total_b;
 total_a = total_b = 0;
 for (int i = 0; i < n; ++i) {
 if ((1)) {
 put_a[i] = true;
 total_a += a[i];
 } else {
 put_a[i] = false;
 total_b += b[i];
 }
 }
 if ((2)) {
 printf("%.2f", total_a * 0.95 + total_b);
 return 0;
 }
 f[0] = 0;
 for (int i = 1; i < threshold; ++i)
 f[i] = Inf;
 int total_b_prefix = 0;
 for (int i = 0; i < n; ++i)
 if (!put_a[i]) {
 total_b_prefix += b[i];
 for (int j = threshold - 1; j >= 0; --j) {
 if ((3) >= threshold && f[j] != Inf)
 ans = min(ans, (total_a + j + a[i]) * 0.95
 + (4));
 f[j] = min(f[j] + b[i], j >= a[i] ? (5) : Inf);
 }
 }
 printf("%.2f", ans);
 return 0;
}

```