

# CSP-J 2019 初赛题解

## (CSP-J) 入门级参考答案

### 一、单项选择题 (共15题, 每题2分, 共计30分)

1	2	3	4	5	6	7	8	9	10
A	D	C	A	A	D	C	C	B	C
11	12	13	14	15					
C	A	C	B	A					

### 二、阅读程序 (除特殊说明外, 判断题1.5分, 单选题3分, 共计40分)

第1题	判断题 (填√或×)				单选题	
	1)	2)	3)	4)	5)	6)
	×	√	×	√	B	B
第2题	判断题 (填√或×)				单选题	
	1)	2)	3)	4)	5)	6)
	√	×	×	×	A	A
第3题	判断题 (填√或×)		单选题			
	1)	2)	3)	4)	5)	6) (4分)
	×	√	A	D	D	B

### 三、完善程序 (单选题, 每小题3分, 共计30分)

第1题					第2题				
1)	2)	3)	4)	5)	1)	2)	3)	4)	5)
C	D	B	B	B	B	D	C	A	B

单项选择题(共 15 题，每题 2 分，共计 30 分;每题有且仅有一个正确选项)

1.中国的国家顶级域名是()

A.cn      B.ch      C.chn      D.china

正确答案:A

考点：计算机基础-计算机网络-域名

解析：典型的国家顶级域名有.cn（中国）、.us（美国）、.uk（英国）、.jp（日本）、.sg（新加坡）等。

典型的通用顶级域名有.edu（教育机构）、.gov（政府部门）、.net（网络组织）、.com（商业组织）、.org（非营机构）、.mil（军事部门）等。

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

正确答案：D

考点：计算机基础-数制与编码-二进制-位运算

解析：逐位进行与运算。对于每一位，0与0得0，1与0得0，0与1得0，1与1得1。

1	1	1	0	1	1	1	0	0	1	0	1	1	1
0	1	0	1	1	0	1	1	1	0	1	0	1	1
0	1	0	0	1	0	1	0	0	0	0	0	1	1

所以答案选 D。

3.一个 32 位整型变量占用()个字节

A. 32    B. 128    C.4    D.8

正确答案:C

解析:一个字节占用 8 位， $32/8 = 4$  字节，正确答案是 C。

4.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)

```
s = a;  
for (b=1;b<=c;b++)  
    s=s-1;
```

则与上述程序段功能等价的赋值语句是()

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

正确答案：A

解析： $s$  一开始等于  $a$ ，从 1 到  $c$ ， $s$  一共执行了  $c$  次自减 1 的操作，所以答案是 A

5. 设有 100 个已排好序的数据元素，采用折半查找时，最大比较次数为()

A. 7      B. 10      C. 6      D. 8

正确答案:A

解析：我们假设二分，每次我们都是右半区区间分的多一点，那么我想搜索到第 100 号元素需要多少次呢？

已知[1~100]一共 100 个数

比较次数	区间数的个数	区间值
1	50	[51~100]
2	25	[76~100]
3	13	[88~100]
4	7	[94~100]
5	4	[97~100]
6	2	[99~100]
7	1	100

每次折半搜索，一共是 7 次。正确答案是 A。

7.链表不具有的特点是()

- A.插入删除不需要移动元素      B.不必事先估计存储空间  
C.所需空间与线性表长度成正比   D.可随机访问任一元素

正确答案： D

解析：在链表中间的元素不能直接随机访问，需要从我们的链表的头节点开始访问。

7.把 8 个同样的球放在 5 个同样的袋子里，允许有的袋子空着不放，问共多少种不同的分法?()提示:如果 8 个球都放在一个袋子里，无论是哪个袋子，都只算同一种分法

- A. 22      B. 24      C.18      D.20

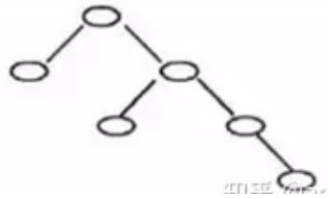
正确答案： C

解析：我们枚举法就可以了，由于允许空着不放，所以放球的袋子{1,2,3,4,5}的情况下都是符合我们条件的，我们只需将他们都加起来

放球的袋子数量	方案数量
1	1 (8)
2	4 qs((1,7),(2,6),(3,5)(4,4))
3	5 qs(1,1,6),(1,2,5)(1,3,4) (2,2,4)(2,3,3)
4	5 (1,1,1,5),(1,1,2,4),(1,1,3,3) (1,2,2,3),(2,2,2,2)
5	3 (1,1,1,1,4),(1,1,1,2,3) (1,1,2,2,2)

一共是 18 种，正确答案是 C。

8.

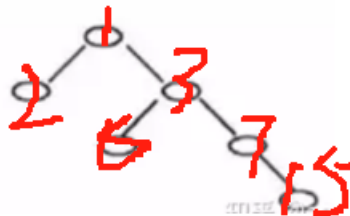


一棵二叉树如右图所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点（根结点的下标为 1,若某结点的下标为  $i$ ,则其左孩子位于下标  $2i$  处、右孩子位于下标  $2i + 1$  处），则该数组的最大下标少为（ ）。

A. 6    B. 10    C. 15    D. 12

正确答案：C

解析：右节点每次比上次大 1 倍+1；



如图，所以答案是 15。

9. 100 以内最大的素数是（ ）。

A. 89    B. 97    C. 91    D. 93

正确答案:B

解析：上述选项质数是 89,97，最大的是 97。

10.319 和 377 的最大公约数是 ( ) 。

A. 27 B. 33 C. 29 D. 31

正确答案：C

解析： $\text{GCD}(319, 377) = \text{GCD}(58, 319) = \text{GCD}(29, 58) = 29$

11. 新学期开学了，小胖想减肥，健身教练给小胖制定了两个训练方案。

方案一：每次连续跑 3 公里可以消耗 300 千卡（耗时半小时）：

方案二：每次连续跑 5 公里可以消耗 600 千卡（耗时 1 小时）。小胖每周周一到周四能抽出半小时跑步，周五到周日能抽出一小时跑步。另外，教练建议小胖每周最多跑 21 公里，否则会损伤膝盖。请问如果小胖想严格执行教练的训练方案，并且不想损伤膝盖，每周最多通过跑步消耗多少千卡？ ( )

A. 3000    B. 2500    C. 2400    D. 2520

正确答案：A

解析：小胖的存量是 21 公里，他的需求是尽可能多的消耗卡路里，1 小时的方案里每公里消耗的卡路里 120 千卡，半个小时的方案里每公里消耗的卡路里是 100 千卡。

周五到周日一共三个小时所以消耗  $600 \times 3 = 1800$  千卡的能



量，剩下的周一到周四一共是  $300 \times 4 = 1200$  千卡。

所以一共 3000 千卡。

12. 一副纸牌除掉大小王有 52 张牌，四种花色，每种花色 13 张。假设从这 52 张牌中随机抽取 13 张纸牌，则至少（ ）张牌的花色一致。

A. 4      B. 2      C. 3      D. 5

正确答案：A

解析：一共是 13 张牌，4 个花色，那么正确答案就会变成

$\left\lceil \frac{13}{4} \right\rceil$ ，答案是 4。

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

正确答案：B

解析：我们用乘法定理写，我们分析一下车牌的模型

1	2	3	4	5
---	---	---	---	---

我们必须保证 1-5，2-4，位的数可以互相通过旋转得到，  
可以通过旋转得到的数字对是 0-0 1-1 6-9 8-8，9-6，一共有 5 对，由于最中间位是一个位置，所以必须是中心对称的  
(0,1,8)一共是 3 种

1	2	3	4	5
5 种	5 种	3 种	5 种	5 种

根据乘法定理，答案是  $5*5*3 = 75$  种，所以答案是 C。

14. 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. ABCDEFGIHJ

B. ABDEGHJCFI

C. ABDEGJHCFI

D. ABDEGHJFIC

正确答案：B

解析：我们说后序遍历是“左右根”，中序遍历是“左根右”

所以后序遍历的最后一个位置是 A,A 就是二叉树的根，那么

中序 “DBGEHJ” 是左子树，“CFI” 是右子树的序列

“DGJHEB” 是我们左子树的后序遍历，那么它的根节点就是 B，那么根据中序遍历，B 的左子树是 D，

右子树是 “GJHE ”，B 的右子树根节点是 E，

E 的左子树是 G,右子树是” JH ”，他的根节点是 H，

中序遍历里 J 在 H 右边，所以是 J 是 H 的左子树

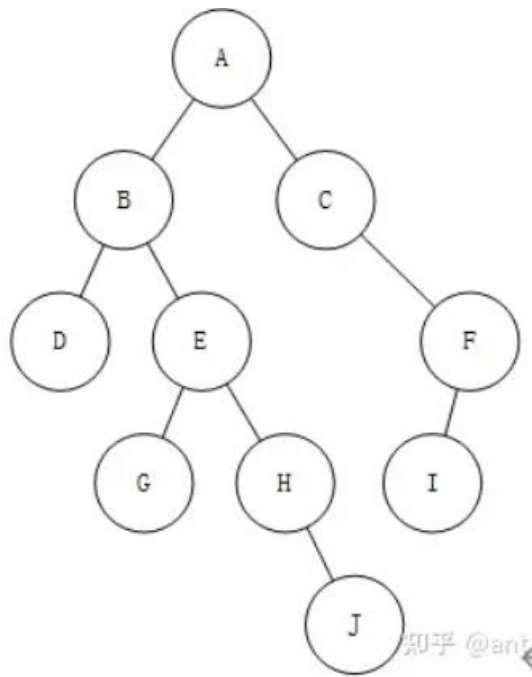
A 的右子树” IFC” ，C 是根节点

C 没有左子树，IF 里 I 是 F 的左子树，

我们现在可以汇总了

A(根)	左子树根 B，右子树根 C
B	左子树根 D，右子树根 E
C	无左子树，右子树根 F
D	无
E	左子树根 G，右子树根 H
F	左子树根 I，右子树根无
G	无
H	左子树根无，右子树根 J
I	无
J	无

所以我们现在可以基于这个表来建树了



我们的前序遍历是“ABDEGHJCFI”,正确答案选 B

15. 以下哪个奖项是计算机科学领域的最高奖？（）

A.图灵奖      B.鲁班奖      C.诺贝尔奖      D.普利策奖

正确答案:A

解析：图灵奖由美国计算机协会于 1966 年设立，其名称取自计算机科学之父图灵，专门奖励对计算机事业作出重要贡献的个人。

**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×；除特殊说明外，判断题 1.5 分，选择题 4 分，共计 40 分）**

```
1  #include <cstdio>↵
2  #include <cstring>↵
3  using namespace std;↵
4  char st[100];↵
5  int main() {↵
6      scanf("%s", st);↵
7      int n = strlen(st);↵
8      for (int i = 1; i <= n; ++i) {↵
9          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(st);↵
16         return 0;↵
17     }↵
```

- 1) 输入的字符串只能由小写字母或大写字母组成。 ( )
- 2) 若将第 8 行的 “i = 1” 改为 “i = 0” , 程序运行时会发生错误。 ( )
- 3) 若将第 8 行的 “i <= n” 改为 “i \* i <= n” , 程序运行结果不会改变。 ( )
- 4) 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。 ( )

•**选择题**

- 5) 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

- 6) 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

解析: 我们来粗略地分析一下这个程序的结构, 这个结构本质上很简单, 先是输入一个字符串进入到字符数组 `st[100]` 里面, 然后我们用一个循环遍历这个字符串, 然后把某些特定的字符改变一下, 最后输出他们的结果

```

int n = strlen(st);
for (int i = 1; i <= n; ++i) {
    if (n % i == 0) {
        char c = st[i - 1];
        if (c >= 'a')
            st[i - 1] = c - 'a' + 'A';
    }
}

```

```

if (c >= 'a')
    st[i - 1] = c - 'a' + 'A';

```

这句话的意思是小写改成大写

那么什么样的情况要触发小写改成大写的操作呢？

```

if (n % i == 0)

```

i 是 n 的因数的时候，字符串第 i 项字符就可以小写改成大写

那么我们开始看各种选项吧！！

### 1. 判断题解析

└

1) 输入的字符串只能由小写字母或大写字母组成。 ( )

正确答案:错误

解析：字符串的形式是 char 型数组,char 型不仅仅可以表示大小写字母，还有数字跟各种标点符号

2) 若将第 8 行的 “ $i = 1$ ” 改为 “ $i = 0$ ” , 程序运行时会发生错误。 ( )

正确答案：正确

解析：若  $i$  可以为 0, 则第 9 行的 if 语句条件 “ $n \% i == 0$ ” 将发生运行时错误 RE。

3) 若将第 8 行的 “ $i \leq n$ ” 改为 “ $i * i \leq n$ ” , 程序运行结果不会改变。 ( )

正确答案：错误

解析：改变之后字符串的结尾有可能不会加工，但是改变之后是一定会加工的(因为自己是自己的因数)

导致输出发生改变

4) 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。 ( )

正确答案：正确

解析：大写字母的 ASCII 码值是 65~90 小于 ‘a’ 的 97 所以不会发生改变



## 2. 选择题解析

5) 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

正确答案: B

解析: 18 的因数有 1, 2, 3, 6, 9, 18. 一共是 6 个字符不一样, 故选 B

6) 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

正确答案: B

解析:  $100000 = 2^5 * 5^5$ , 所以根据约数定理, 它的约数个数是  $(5+1)*(5+1) = 36$ 。一共是 36 种。

## 第二题

```
1  #include <stdio>
2  using namespace std;
3  int n, m;

4  int a[100], b[100];

6  int main() {
7      scanf("%d%d", &n, &m);
8      for (int i = 1; i <= n; ++i)
9          a[i] = b[i] = 0;
10     for (int i = 1; i <= m; ++i) {
11         int x, y;
12         scanf("%d%d", &x, &y);
13         if (a[x] < y && b[y] < x) {
14             if (a[x] > 0)
15                 b[a[x]] = 0;
16             if (b[y] > 0)
17                 a[b[y]] = 0;
18             a[x] = y;
19             b[y] = x;
20         }
21     }
22     int ans = 0;
23     for (int i = 1; i <= n; ++i) {
24         if (a[i] == 0)
25             ++ans;
26         if (b[i] == 0)
27             ++ans;
28     }
29     printf("%d\n", ans);
30     return 0;
31 }
```

• 判断题

- 1) 当  $m > 0$  时, 输出的值一定小于  $2n$ 。 ( )
- 2) 执行完第 27 行的 “++ans” 时, ans 一定是偶数。 ( )
- 3)  $a[i]$  和  $b[i]$  不可能同时大于 0。 ( )
- 4) 若程序执行到第 13 行时,  $x$  总是小于  $y$ , 那么第 15 行不会被执行。

• 选择题

- 5) 若  $m$  个  $x$  两两不同, 且  $m$  个  $y$  两两不同, 则输出的值为 ( )  
  
A.  $2n-2m$     B.  $2n+2$     C.  $2n-2$     D.  $2n$
- 6) 若  $m$  个  $x$  两两不同, 且  $m$  个  $y$  都相等, 则输出的值为 ( )  
  
A.  $2n-2$     B.  $2n$     C.  $2m$     D.  $2n-2m$

程序分析: 这个程序考察的是对数组集合性质的认识, 程序的结构很简单, 首先是输入  $x, y$ ;

在一般情况下是  $a[x] = y; b[y] = x;$

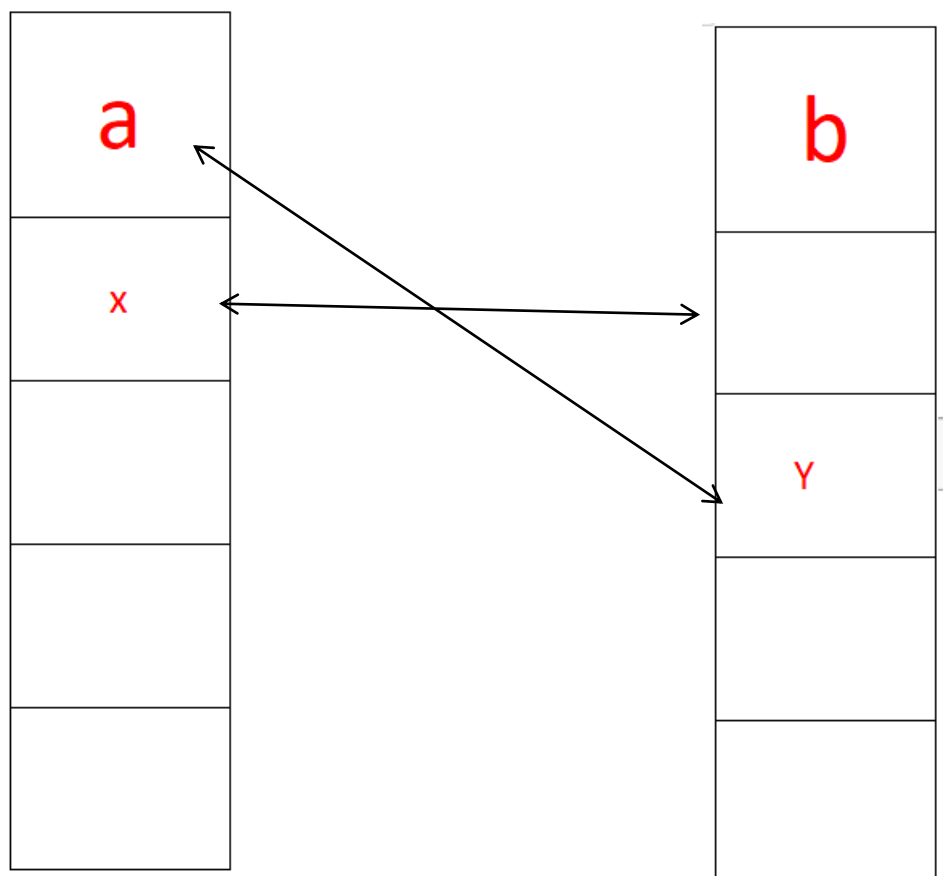
```

)   for (int i = 1; i <= m; ++i) {
)       int x, y;
)       scanf("%d%d", &x,&y);
)       if (a[x] < y && b[y] < x) {
)           if (a[x] > 0)
)               b[a[x]] = 0;
)           if (b[y] > 0)
)               a[b[y]] = 0;
)           a[x] = y;
)           b[y] = x;
)       }
)   }

```

我们发现必须触发了 `if (a[x] < y && b[y] < x)`

这个条件是在什么条件下成立呢？



是原来  $x$  连接的点  $a[x]$  小于  $y$ ，原来  $y$  连接的点  $b[y]$  小于  $x$

我们才能建立  $x$  跟  $y$  的连接

```
a[x] = y;↵  
b[y] = x;↵
```

然后，在这之前

我们需要让  $x$  连接的点  $a[x]$ ，它的连接  $b[a[x]]$  等于 0 (原来是等于  $x$  的); 让  $y$  连接的点  $b[y]$ ，它的连接  $a[b[y]]$  等于 0 (原来是等于  $y$  的)

```
if (a[x] > 0)↵  
    b[a[x]] = 0;↵  
if (b[y] > 0)↵  
    a[b[y]] = 0;↵
```

这个循环结束了，我们再统计一下  $a$ ， $b$  数组中无连接的个数和并输出

```
22  int ans = 0;↵  
23  for (int i = 1; i <= n; ++i) {↵  
24      if (a[i] == 0)↵  
25          ++ans;↵  
26      if (b[i] == 0)↵  
27          ++ans;↵  
28  }↵  
29  printf("%d\n",ans);↵
```

Ans 就是我们的要输出的答案

## 1. 判断题解析

1) 当  $m > 0$  时, 输出的值一定小于  $2n$ 。 ( )

正确答案: 错误

解析: 如果输入的是 1 0 0 ,ans 输出的答案是  $2 * n$

2) 执行完第 27 行的 “++ans” 时, ans 一定是偶数。 ( )

正确答案: 正确

解析: 连接是双向的连接, 所以答案必是偶数

3)  $a[i]$ 和  $b[i]$ 不可能同时大于 0。 ( )

正确答案: 错误

解析: 如果  $a[i] = i, b[i] = i, i > 0$ , 他们就都大于 0 了, 错误

4) 若程序执行到第 13 行时,  $x$  总是小于  $y$ , 那么第 15 行不会被执行。

正确答案: 错误

解析: 这个样例 3 15 34 35 ,  $x$  小于  $y$

```
14         if (a[x] > 0)↵  
15             b[a[x]] = 0;↵
```

就是  $x$  原来连接的点  $a[x], b[a[x]]$  的点清零了，我们发现第三次输入的  $x\ y(3,4)$  会覆盖前两次的连接情况，所以错误

## 2. 选择题解析

5) 若  $m$  个  $x$  两两不同, 且  $m$  个  $y$  两两不同, 则输出的值为 ( )

A.  $2n-2m$    B.  $2n+2$    C.  $2n-2$    D.  $2n$

正确答案: A

解析: 由于输入的  $x, y$  都不相同, 所以就不存在覆盖的情况, 在  $a[]$  数组中有连接的点是  $m$  个,  $b[]$  数组中有连接的点是  $m$  个, 所以两个数组中未连接的点总和是  $2*n - 2*m$

6) 若  $m$  个  $x$  两两不同, 且  $m$  个  $y$  都相等, 则输出的值为 ( )

A.  $2n-2$    B.  $2n$    C.  $2m$    D.  $2n-2m$

正确答案: A

解析:  $y$  都一样, 那么最后  $b[y]$  的数值必然是  $m$  个  $x$  中数值最大的, 剩下的都不会建立连接, 所以, 一共一对连接所以 Ans 等于  $2*n-2$ , 正确答案选 A

### 第三题





•判断题

- 1) 如果 a 数组有重复的数字, 则程序运行时会发生错误 ( )
- 2) 如果 b 数组全为 0,则输出为 0。 ( )

•选择题

- 3) 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

- 4) 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

- 5) 当  $n=10$  时,若 b 数组满足, 对任意  $0 \leq i < n$ 。都有  $b[i] = i + 1$ , 那么输出最大为 ( )。

A. 386      B. 383    C. 384    D. 385

- 6) (4 分) 当  $n=100$  时, 若 b 数组满足, 对任意  $0 \leq i < n$ , 都有  $b[i]=1$ ,那么输出最小为 ( )。

A. 582      B. 580    C. 579    D. 581

程序解析：这道题是个递归程序，首先我们的退出条件是第 8, 9 行的

```
8    if (l > r)↵  
9        return 0;↵
```

当  $l > r$  的时候,中止。然后我们从  $a[l] \sim a[r]$  中找到最小的值

```
11    for (int i = l; i <= r; ++i) {↵  
12        if (min > a[i]) {↵  
13            min = a[i];↵  
14            mink = i;↵  
15    }
```

我们让  $min$  作为这个最小的值， $mink$  作为这个值的下标

```
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];↵
```

然后就是分治， $lres$  接受  $a[l] \sim a[mink-1]$  的情况， $rres$  用来接受  $a[mink+1] \sim a[r]$  的情况

最后我们把所有的值加起来，输出！！

## 判断题解析：

1) 如果  $a$  数组有重复的数字, 则程序运行时会发生错误 ( )

正确答案：错误

解析:11 行~15 行返回的是最靠左的最小值，所以即使有重复的数字程序也能运行下去！！

2) 如果 b 数组全为 0,则输出为 0。 ( )

正确答案：正确

解析：返回的结果是  $\sum depth_i * b[i]$ ，由于  $b[i]$  恒等于 0,所以总的数也等于 0。

3. 选择题

3) 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

正确答案：A

解析：最坏情况下，程序所构造的二叉树的每个结点至多仅有一个子结点，此时，程序将递归 100 层，其中第  $i$  层进行  $100-i+1$  次第 12 行的比较运算，总执行次数是  $100+99+98+\dots+1 \approx 5000$ . 正确答案是 A。

4) 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

正确答案: D

解析: 最佳情况下, 程序构造二叉树时, 对于每个结点会尽可能均分其左右子树。定义根结点深度为 1, 则含  $n=100$  个结点的树的深度最小为  $\log n \approx 7$ , 此时每选定一层结点, 程序都需要执行约  $n$  次的第 12 行的比较运算, 因此总执行次数约为  $n \log n \approx 600$ 。

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

A. 386      B. 383      C. 384      D. 385

正确答案: D

解析: 要使输出的  $ans$  值尽可能大, 程序所构造的二叉树的深度应尽可能的大。定义根结点深度为 1, 则含 10 个结点的二叉树的最大深度为 10, 因此  $ans$  的最大值为  $1*1+2*2+3*3+\dots+10*10=385$ 。

6) (4分) 当  $n=100$  时, 若  $b$  数组满足, 对任意  $0 \leq i < n$ , 都有  $b[i]=1$ , 那么输出最小为 ( )。

A. 582      B. 580      C. 579      D. 581

正确答案: B

解析: 此时, 要使输出的  $ans$  值尽可能小, 程序构造的点的深度应该尽可能的小, 总深度最小是完全二叉树

深度	1	2	3	4	5	6	7
	1	2	4	8	16	32	37

所以总的深度是 580。

### 三、完善程序 (单选题, 每题 3 分, 共计 30 分)

1. (矩阵变幻) 有一个奇幻的矩阵, 在不停的变幻, 其变幻方式为: 数字 0 变成矩阵

$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ , 数字 1 变成矩阵  $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ , 最初该矩阵只有一个元素 0, 变幻  $n$  次后, 矩阵会变成什么样?

例如, 矩阵最初为:  $[0]$ ; 矩阵变幻 1 次后:  $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ ; 矩阵变幻 2 次后:

$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$

输入一行一个不超过 10 的正整数  $n$ 。输出变幻  $n$  次后的矩阵。

试补全程序。

提示:

“<<”表示二进制左移运算符, 例如  $(11)_2 \ll 2 = (1100)_2$ ;

而 “^”表示二进制异或运算符, 它将两个参与运算的数中的每个对应的二进制位一一进行比较, 若两个二进制位相同, 则运算结果的对应二进制位为 0, 反之为 1。 |

知乎 @ant

```

1  #include <stdio>
2  using namespace std;
3  int n;
4  const int max_size = 1 << 10;

6  int res[max_size][max_size];

8  void recursive(int x,int y,int n,int t) {
9      if (n == 0) {
10         res[x][y]= ① ;
11         return;
12     }
13     int step = 1 << (n - 1);
14     recursive(②,n - 1, t);
15     recursive(x, y + step, n - 1, t);
16     recursive(x + step, y, n - 1, t);
17     recursive(③, n - 1, !t);
18 }
19
20 int main() {
21     scanf("%d", &n);
22     recursive(0,0,④);
23     int size = ⑤ ;
24     for (int i = 0; i < size; ++i) {
25         for (int j = 0; j < size; ++j)
26             printf("%d", res[i][j]);
27         puts("");
28     }
29     return 0;
30 }

```

1) ①处应填 ( )

A.  $n \% 2$    B. 0   C. t   D. 1

2) ②处应填 ( )

A.  $x - \text{step}, y - \text{step}$    B.  $x, y - \text{step}$

C.  $x - \text{step}, y$    D.  $x, y$

3) ③处应填 ( )

A.  $x - \text{step}, y - \text{step}$    B.  $x + \text{step}, y + \text{step}$

C.  $x - \text{step}, y$    D.  $x, y - \text{step}$

4) ④处应填 ( )

A.  $n - 1, n \% 2$    B.  $n, 0$

C.  $n, n \% 2$    D.  $n - 1, 0$

5) ⑤处应填 ( )

A.  $1 \ll (n + 1)$    B.  $1 \ll n$

C.  $n + 1$    D.  $1 \ll (n - 1)$

程序分析：这题考的是对递归与矩阵的认识

每次递归进入更深一层的时候

1 -  $\rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$       2 -  $\rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

```
20 int main() {  
21     scanf("%d", &n);  
22     recursive(0,0,4);  
23     int size = 5 ;  
24     for (int i = 0; i < size; ++i) {  
25         for (int j = 0; j < size; ++j)  
26             printf("%d", res[i][j]);  
27         puts("");  
28     }  
29     return 0;  
30 }
```

知乎 @ant

我们分析一下这个程序的大致结构:经过 recursive()函数的处理,res[][]就是输出的结果那么这个 recursive()函数怎么作用到我们的 res 上呢?

```
8 void recursive(int x,int y,int n,int t) {  
9     if (n == 0) {  
10         res[x][y] = 1 ;  
11         return;  
12     }  
13     int step = 1 << (n - 1);  
14     recursive(2,n - 1, t);  
15     recursive(x, y + step, n - 1, t);  
16     recursive(x + step, y, n - 1, t);  
17     recursive(3, n - 1, !t);  
18 }
```



我们先看一下这个递归函数

首先是这个函数的形参  $x, y, n, t$  是什么意思呢？

$x$  表示的是内层坐标，

$y$  表示外层坐标，

$n$  表示递归的层数，

$t$  表示当前的输出数据是 1 还是 0。

```
9    if (n == 0) {  
10        res[x][y] = ① ;  
11        return;  
12    }
```

我们的递归在  $n = 0$  的时候结束。

```
14    recursive(②, n - 1, t);  
15    recursive(x, y + step, n - 1, t);  
16    recursive(x + step, y, n - 1, t);  
17    recursive(③, n - 1, !t);
```

递归是 1 函数声明生成 4 个

<code>recursive(②, n - 1, t);</code>	<code>recursive(x, y + step, n - 1, t);</code>
<code>recursive(x + step, y, n - 1, t)</code>	<code>recursive(③, n - 1, !t);</code>

由于最后一个数值与之相反，所以返回 ! $t$

当  $n = 0$  时， $res[x][y] = t$

## 选择题解析

1) ①处应填 ( )

A.  $n \% 2$    B. 0   C.  $t$    D. 1

正确答案:C

解析：①处

```
9    if (n == 0) {  
10        res[x][y] = ① ;  
11        return;  
12    }
```

我们一定是等于 t 的

∴ 2) ②处应填 ( )

A.  $x - \text{step}, y - \text{step}$     B.  $x, y - \text{step}$

C.  $x - \text{step}, y$                       D.  $x, y$

正确答案: D

解析:

```
14    recursive(②, n - 1, t);
```

第一个必然是在最左边，所以一切保持不变  $x, y$ ;

3) ③处应填 ( )

A.  $x - \text{step}, y - \text{step}$               B.  $x + \text{step}, y + \text{step}$

C.  $x - \text{step}, y$                       D.  $x, y - \text{step}$

正确答案：B

解析：③处的位置

```
17 recursive(③, n - 1, !t);
```

这是第四个位置，是横纵坐标+step，所以选 B。

4) ④处应填 ()

- A.  $n - 1, n \% 2$       B.  $n, 0$   
C.  $n, n \% 2$       D.  $n - 1, 0$

正确答案：B

解析：④处的位置是

```
22 recursive(0, 0, ④);
```

这个是后面两个传参变量的值  $n, t$ ，因为  $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$  是 1 层，所以  $n$  等于  $n$ ， $t$  是幻方的初始数值，根据题意一开始是  $[0]$ ，所以  $t = 0$ ;

5) ⑤处应填 ()

- A.  $1 << (n + 1)$       B.  $1 << n$   
C.  $n + 1$       D.  $1 << (n - 1)$

正确答案：B

解析：本题的递归算法从第 0 层开始，每进入一层矩阵的大小就扩大 1 倍，一共是  $n$  层，所以数值是  $1 * 2^n$ ，所以选 B。

2. (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  $1 \leq n \leq 10^7$ ,  $1 \leq a[i], b[i] \leq 10^4$ 。

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。

知乎 @ant

```
1  #include <cstdio>
2  #include <cstring>
3  using namespace std;
4  const int maxn = 10000000;
5  const int maxs = 10000;

7  int n;
8  unsigned a[maxn], b[maxn], res[maxn], ord[maxn];
9  unsigned cnt[maxs + 1];
10
11 int main() {
12     scanf("%d", &n);
13     for (int i = 0; i < n; ++i)
14         scanf("%d%d", &a[i], &b[i]);
15     memset(cnt, 0, sizeof(cnt));
16     for (int i = 0; i < n; ++i)
17         ① ; //利用 cnt 数组统计数量
18     for (int i = 0; i < maxs; ++i)
19         cnt[i + 1] += cnt[i];
20     for (int i = 0; i < n; ++i)
21         ② ; //记录初步排序结果
22     memset(cnt, 0, sizeof(cnt));
23     for (int i = 0; i < n; ++i)
24         ③ ; //利用 cnt 数组统计数量
25     for (int i = 0; i < maxs; ++i)
26         cnt[i + 1] += cnt[i];
27     for (int i = n - 1; i >= 0; --i)
28         ④ ; //记录最终排序结果
29     for (int i = 0; i < n; ++i)
30         printf("%d %d\n", ⑤);
31     return 0;
```

知乎 @ant

1) ①处应填()

A. ++cnt[i]

B. ++cnt[b[i]]

C. ++cnt[a[i] \* maxs + b[i]]

D. ++cnt[a[i]]

2) 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

3) ③处应填 ()

A. ++cnt[b[i]]

B. ++cnt[a[i] \* maxs + b[i]]

C. ++cnt[a[i]]

D. ++cnt [i]

4) 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]`

5) 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]]]`

程序解析：双关键字排序是非常需要对数组的理解！！

我们需要知道 `a[i], b[i]` 是每个数的输入序对应的第一关键字，跟第二关键字的数值。`Ord` 是第二关键字排序后的结果

```
16  for (int i = 0; i < n; ++i)↵
17      ① ; //利用 cnt 数组统计数量↵
18  for (int i = 0; i < maxs; ++i)↵
19      cnt[i + 1] += cnt[i];↵
20  for (int i = 0; i < n; ++i)↵
21      ② ; //记录初步排序结果↵
```

cnt 是数域带宽里的, 每一个<数值>, 对应的是<第二关键词最大的排名>,

这等于是将重复的原值离散化了

那么这个 ord[i]就是第二关键词排名数->原位置

然后下一步就是什么

```
23  for (int i = 0; i < n; ++i)↵
24      ③ ; //利用 cnt 数组统计数量↵
25  for (int i = 0; i < maxs; ++i)↵
26      cnt[i + 1] += cnt[i];↵
27  for (int i = n - 1; i >= 0; --i)↵
28      ④ ; //记录最终排序结果↵
29  for (int i = 0; i < n; ++i)↵
30      printf("%d %d\n", ⑤); ↵
```

cnt 是数域带宽里的, 每一个<数值>-><第一关键词最大的排名>

我们要根据第二关键词的反顺序来获得

res 就是我们双关键词排序的结果! (是双关键词排名-->位置)

### 选择题解析：

1) ①处应填()

A. ++cnt[i]

B. ++cnt[b[i]]

C. ++cnt[a[i] \* maxs + b[i]]

D. ++cnt[a[i]]

正确答案:B

解析：我们先搞出  $b[i]$  各个数值对应的排名，再按照这个排序，所以 B 选项正确

2) 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



正确答案：D

解析：

```
20   for (int i = 0; i < n; ++i)↵  
21       ② ; //记录初步排序结果↵
```

②处记录的是第二关键词排序的结果，应该是<排名>->位置，`--cnt[a[i]]`输出的是输入序第*i*位的排名，所以`ord[cnt[a[i]]]`应该等于这个排名的数的位置，所以等于*i*

正确答案是 D。

3) ③处应填 ()

A. `++cnt[b[i]]`

B. `++cnt[a[i] * maxs + b[i]]`

C. `++cnt[a[i]]`

D. `++cnt [i]`

正确答案：C

解析：这个就是搞出 `a[i]` 数组里各个数值对应的排名，再按照这个排序，所以 C 选项正确

4) ④处应填 ( )

A.  $\text{res}[\text{--cnt}[\text{a}[\text{ord}[\text{i}]]]] = \text{ord}[\text{i}]$

B.  $\text{res}[\text{--cnt}[\text{b}[\text{ord}[\text{i}]]]] = \text{ord}[\text{i}]$

C.  $\text{res}[\text{--cnt}[\text{b}[\text{i}]]] = \text{ord}[\text{i}]$

D.  $\text{res}[\text{--cnt}[\text{a}[\text{i}]]] = \text{ord}[\text{i}]$

正确答案：A

解析：我们要按照第二关键词的反顺序(从大到小)来确定，第一关键词的前后关系，由于 **res** 数组是双关键词序->原位置。**ord[i]**是第二关键词序对应的原位置。

**cnt[a[ord[i]]]**就是根据第二关键词反顺序得到的双关键词序。 **res** 数组是双关键词序->原位置！

所以  $\text{res}[\text{--cnt}[\text{a}[\text{ord}[\text{i}]]]] = \text{ord}[\text{i}]$ , 正确答案是 A。

5) ⑤处应填 ( )

A.  $\text{a}[\text{i}], \text{b}[\text{i}]$

B.  $\text{a}[\text{res}[\text{i}]], \text{b}[\text{res}[\text{i}]]$

C.  $\text{a}[\text{ord}[\text{res}[\text{i}]]], \text{b}[\text{ord}[\text{res}[\text{i}]]]$

D.  $\text{a}[\text{res}[\text{ord}[\text{i}]]], \text{b}[\text{res}[\text{ord}[\text{i}]]]$

正确答案：B

解析：我们要按照双关键词序来输出每个数的数值，我们需要确定双关键词序->原位置的关系，`res[i]`正是做这个的，确定了位置，就是原位置->数值的关系，所以选 B。