第二十四届全国青少年信息学奥林匹克联赛初赛

普及组 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 7 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。

1. 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）
2. D
3. D
4. D
5. B
6. B
7. A
8. A
9. A
10. A
11. B
12. A
13. B
14. B
15. B
16. B
17. 以下哪一种设备属于输出设备：（ ）
    1. 扫描仪 B. 键盘 C. 鼠标 D. 打印机

解析：这是一道送分题，会就会，不会就不会。所以选D。

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（ ）。
   1. (269)16
   2. (617)10
   3. (1151)8

D．(1001101011)2

解析：这题考核的是进制转换。（269）16= 2 \* 16^2 + 6 \* 16^1 + 9 \* 16 ^0 = 617，(1151)8 = 1 \* 8^3 + 1 \* 8^2 + 5 \* 8^1 + 1 \* 8^0 = 617，而(1001101011)2=619。所以选D。

1. 1MB 等于（ ）。
   1. 1000 字节 B. 1024 字节

C. 1000 X 1000 字节 D. 1024 X 1024 字节

解析：这题考核的是字节之间的转换。1MB=1024KB=1024\*1024B。所以选D。

1. 广域网的英文缩写是（ ）。
   1. LAN

B．WAN

C．MAN

D．LNA

解析：这题是送命题，，会就会，不会就不会。WAN全称为Wide Area Network。所以选B。

1. 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。
   1. 1983
   2. 1984
   3. 1985
   4. 1986

解析：这题是送命题，会就会，不会就不会。NOI 是全国青少年信息学奥林匹克竞赛(National Olympiad in Informatics)的简称。NOI 是国内包括港澳在内的省级代表队最高水平的大赛，自 1984 年至今，在国内包括香港、澳门组织竞赛活动。所以选B。

1. 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

（ ）。

* 1. A B. S C. D D. a

解析：考查数学基础之周期问题。CapsLock为大小写切换（没有输出），所以默认按键5个（输出4个）为一轮。81 ＝ 4\*20 ＋ 1，即从0轮开始，求第20轮的第一个输出。开始状态为小写，第0轮为大写，第1轮为小写，依次类推，第20轮为大写。所以答案为大写状态的第一个字符‘A’。所以选A。

1. 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。
   1. (k h+1 - 1) / (k - 1)
   2. k h-1
   3. k h
   4. (k h-1) / (k - 1)

解析：考查数学基础和树的知识。满k叉树中，深度为0的层有1个节点，深度为1的层有k个节点，深度为2的层有k\*k个节点，依次类推，…，深度为h的层有k^h个节点。因此，总共有1+k+k^2+…+k^h=(1-k^(h+1))/(1-k)=(k^(h+1)-1)/(k-1).所以选A。

1. 以下排序算法中，不需要进行关键字比较操作的算法是（ ）。
   1. 基数排序
   2. 冒泡排序
   3. 堆排序
   4. 直接插入排序

解析：考查NOIP之排序问题。基数排序是利用分配和收集两种基本操作来达到排序的目的，并不需要进行关键字比较操作。所以选A。

1. 给定一个含 N 个不相同数字的数组，在最坏情况下，找出其中最大或最小的数，至少需要 N - 1 次比较操作。则最坏情况下，在该数组中同时找最大与最小的数至少需要（ ）次比较操作。（⌈ ⌉表示向上取整，⌊ ⌋表示向下取整）
   1. ⌈3N / 2⌉ - 2
   2. ⌊3N / 2⌋ - 2
   3. 2N - 2
   4. 2N - 4

解析：如果分别找最大值、最小值，则至少都需要N-1次操作。

同时找最大最小值，有更优化的方法，如果没有学过这个算法，本题只能根据题面猜测肯定小于2N-2，需在A和B里面蒙一个，50%几率。

学过的话，按照下面的优化算法：

N为奇数时，比较次数为3\*(N-1)/2 =(3N+1)/2 - 2

N为偶数时，比较次数为1 +3\*(N-2)/2 = 3N/2 – 2

综合奇偶，显然答案为A

找最大最小值的优化算法：

初始值：

N为奇数，最大值、最小值的初始值都设为第一个元素。

N为偶数，将前两个元素比较，最大值初始值为大的元素，最小值初始值为小的元素。

枚举，每次两个元素（循环步长为2）

比较两个元素，分出大小。

大的元素与最大值比较，比最大值大则设为该元素。

小的元素与最小值比较，比最小值小则设为该元素。

循环结束，得到最大、最小值。

所以选A。

1. 下面的故事与（ ）算法有着异曲同工之妙。

从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事：“从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事：‘从前有座山，山里有座庙，庙里有个老和尚给小和尚讲故事……’”

* 1. 枚举 B. 递归 C. 贪心 D. 分治

解析：这题是送命题，会就会，不会就不会。递归是不断调用自身，和题目中描述类似，所以选B。

1. 由四个没有区别的点构成的简单无向连通图的个数是（ ）。
   1. 6
   2. 7
   3. 8
   4. 9



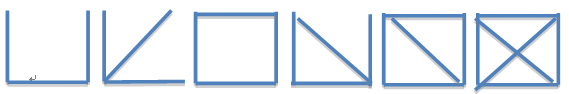
解析：考查NOIP之图论知识。

1. 四个区别的点，意味着以下的图形为相同形状。

用d=[1, 2, 2,1]表示对应点的度，则以上的图的d数组都是一样的。

1. 而以下的图形则是不同的另外一个形状，其d=[1, 3, 1, 1]。题目要求，实际就是找出不同的d数组的个数。
2. 根据边数来分类判断：
3. 小于3条边，不构成连通，排除掉。
4. 3条边：d=[1, 2, 2, 1]和[1, 3,1,1]两种
5. 4条边：d=[2,2,2,2]和[1,3,2,2]两种
6. 5条边：d=[2,2,3,3]一种
7. 6条边：d=[3,3,3,3]一种

一共6种不同的图形，答案是A。



4） 既不含平行边也不包含自环的图称为简单图。所以选A。

1. 设含有 10 个元素的集合的全部子集数为 S，其中由 7 个元素组成的子集数为

T，则 T / S 的值为（ ）。

* 1. 5 / 32
  2. 15 / 128
  3. 1 / 8
  4. 21 / 128

解析： 子集总数S为 2的10次方= 1024

7个元素集合数T为C（10,7）=10!/（3!7!）= 120

T/S = 120/1024 = 15/128。所以选B。

1. 10000 以内，与 10000 互质的正整数有（ ）个。
   1. 2000
   2. 4000
   3. 6000
   4. 8000

解析： 互质的意思，与10000没有公约数，也即不能被10000的质因子整除。

10000分解质因子：10000 =2\*2\*2\*2\*5\*5\*5\*5

10000以内被2整除的数有5000个

10000以内被5整除的数有2000个

2）和3）重复计算的数，即被10整除的数，有1000个。

被2或5整除的数有：5000 +2000 – 1000 =6000

互质的数有：10000 - 6000 ＝ 4000个。所以选B。

1. 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（ ）。

* 1. x >>= 1
  2. x &= x - 1
  3. x |= x >> 1
  4. x <<= 1

解析： 如果知道x = x&(x-1)是二进制从后往前去掉1个1的话，答案B自然知道。

如果不知道的话，就自己模拟一下吧，比如用一个数5=(101)2

A选项模拟，结果为3

B选项模拟，结果为2

C选项模拟，死循环

D选项模拟，死循环。所以选B。

1. 下图中所使用的数据结构是（ ）。

|  |
| --- |
|  |
| B |
| A |

|  |
| --- |
|  |
|  |
| A |

|  |
| --- |
|  |
|  |
| A |

|  |
| --- |
|  |
| C |
| A |

|  |
| --- |
|  |
|  |
|  |

压入 A 压入 B 弹出 B 压入 C



* 1. 哈希表 B. 栈 C. 队列 D. 二叉树

**解析： 2013年相同的题，所以选B。**

1. 问题求解（共 **2** 题，每题 **5** 分，共计 **10** 分）
2. **去了 没去 没去 没下雨**
3. **544**

**容斥原理：**

在计数时，必须注意没有重复，没有遗漏。为了使重叠部分不被重复计算，人们研究出一种新的计数方法，这种方法的基本思想是：先不考虑重叠的情况，把包含于某内容中的所有对象的数目先计算出来，然后再把计数时重复计算的数目排斥出去，使得计算的结果既无遗漏又无重复，这种计数的方法称为容斥原理。

要计算几个集合并集的大小，我们要先将所有单个集合的大小计算出来，然后减去所有两个集合相交的部分，再加回所有三个集合相交的部分，再减去所有四个集合相交的部分，依此类推，一直计算到所有集合相交的部分。

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\_\_\_\_（去了/没去）(1分)，乙\_\_\_没去\_\_\_\_\_（去了/没去）(1分)，丁\_\_没去\_\_\_\_\_\_（去了/没去）(1分)，周末\_\_\_没下雨\_\_\_\_\_（下雨/ 没下雨）(2分)。

解析：简单的逻辑推理， ③是突破口。如果丙去了，那么丁没去。根据②，乙也没去。根据④，丁没去，丙去了，那么甲一定去了。根据①，周末肯定没下雨。

1. 从1到2018这2018个数中，共有\_\_ 544\_\_\_\_\_\_\_\_个包含数字8的数。包含数字8的数是指有某一位是“8”的数， 例如“2018”与“188”。

解析：小学奥数。数码问题，我用了一种类似于容斥原理的方法

计算含1个、2个、3个8的数的个数，然后容斥。

考虑三位数，含1个：C(3,1)\*100=300;

含2个：C(3,2)\*10=30;

含3个：C(3,3)\*1=1.

考虑2000以内的数，千位是0和1，总共2种，共2\*（300-30+1）=542中。

再考虑2000-2018,共有两个，总共544个。

2.

1--99有19个，100-199有19个，......

800---899有100个

1800--1899有100个

一共有:

（20-2）×19+2×100+2

=18×19+202

=342+202

=544个

三、阅读程序写结果（共 **4** 题，每题 **8** 分，共计 **32** 分）

1. **RuanHuoMianTai**
2. **4**
3. **8**
4. **6**
5. #include <cstdio>

char st[100];

int main() {

scanf("%s", st);

for (int i = 0; st[i];++i) {

if ('A' <= st[i] && st[i] <= 'Z')

st[i] += 1;

}

printf("%s\n", st);

return 0;

}

输入：QuanGuoLianSai

输出：RuanHuoMianTai

解析：模拟跑一边即可。我们可以发现，程序要做的是对大写字母增大1.

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：\_\_\_ 4\_\_\_\_\_\_

解析：模拟跑一边即可。我们可以发现，程序要做的是求出[0,x)内比完全平方数对x取余数，余数为1的有几个。

输入15时，有i=1，4, 11, 14总共4个。

1. #include <iostream>

using namespace std;

int n, m;

int findans(int n, int m) {

if (n == 0) return m;

if (m == 0) return n % 3;

return findans(n - 1, m) - findans(n, m - 1) + findans(n - 1, m - 1);

}

int main(){

cin >> n >> m;

cout << findans(n, m) << endl;

return 0;

}

输入：5 6 输出：\_\_\_8\_\_\_\_\_\_

解析：模拟跑一边即可。我们画个表格，从左上角往右下角计算即可，得到findans(5,6)=8.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | M=0 | 1 | 2 | 3 | 4 | 5 | 6 |
| N=0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1 | 0 | 3 | 2 | 5 | 4 | 7 |
| 2 | 2 | -1 | 4 | 1 | 6 | 3 | 8 |
| 3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 4 | 1 | 0 | 3 | 2 | 5 | 4 | 7 |
| 5 | 2 | -1 | 4 | 1 | 6 | 3 | 8 |

**4.** #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[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：\_\_\_\_ 6\_\_\_\_\_

解析：模拟跑一边即可。实际上是求a->b->c->…->a的个数，

总共有10个数，

0->7->8->0,

1->1

2->4>2

3->3

5->5

6->9->6

四、完善程序（ 共 **2** 题，每题 **14** 分，共计 **28** 分）

**(1) i\*i**

**(2)n/i**

**(3)return a**

**(4) a%b**

**(5)ans+gcd(a[i],a[j])**

**(1)a[x]=i**

**(2)i+1**

**(3)R[a[i]]**

**(4)a[i]**

**(5)r[i]**

**1.** （最大公约数之和）下列程序想要求解整数𝑛的所有约数两两之间最大公约

数的和对10007求余后的值，试补全程序。（第一空 2 分，其余 3 分）

举例来说，4的所有约数是1,2,4。1和2的最大公约数为1；2和4的最大公约 数为2；1和4的最大公约数为1。于是答案为1 + 2 + 1 = 4。

 要求 getDivisor 函数的复杂度为𝑂函数的复杂度为𝑂(log max(𝑎, 𝑏))。

#include <iostream> using namespace std; const int N = 110000, P = 10007;

int n; int a[N], len;

int ans;

//求所有的约数

void getDivisor() { len = 0;

for (int i = 1; (1)  i \* i  <= n; ++i) //根号求余数

if (n % i == 0) {

a[++len] = i;

if ( (2) n / i != i) a[++len] = n / i; //不是完全平方数的平方根才加

}

}

//求最大公约数的递归形式

int gcd(int a, int b) {

if (b == 0) {

(3) return a ;

}

return gcd(b, (4) a % b );

}

int main() {

cin >> n;

getDivisor();

ans = 0;

for (int i = 1; i <= len; ++i) {

for (int j = i + 1; j <= len; ++j) {

ans = ( (5) ans + gcd(a[i], a[j]) ) % P; //求和

}

}

cout << ans << endl;

return 0;

}

**2.** 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𝑞𝑖为 第𝑖个位置之后第一个比𝑃𝑖值更大的位置，如果不存在这样的位置，则𝑞𝑖 = 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 ≤ 𝑛 ≤ 105。

#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) a[x] = i; ; //根据值统计位数

}

for (int i = 1; i <= n; ++i) {

R[i] = (2) i + 1 ; //链表的右指针

L[i] = i - 1;

}

for (int i = 1; i <= n; ++i) {

L[ (3) R[a[i]] ] = L[a[i]]; //删除i

R[L[a[i]]] = R[ (4) a[i] ];

}

for (int i = 1; i <= n; ++i) {

cout << (5) R[i] << " "; //输出第𝑖个位置之后第一个比𝑃𝑖值更大的位置

}

cout << endl;

return 0;

}

第二十四届全国青少年信息学奥林匹克联赛初赛

普及组参考答案

一、单项选择题（共 15 题，每题 1.5 分，共计 22.5 分）

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

二、问题求解（共 2 题，每题 5 分，共计 10 分）

|  |  |  |
| --- | --- | --- |
| 1. | 去了、没去、没去、没下雨(第四空2分，其余1分) |  |
| 2 | 544 |  |

三、阅读程序写结果（共 4 题，每题 8 分，共计 32 分）

1. RuanHuoMianTai
2. 4
3. 8

4. 6

五、完善程序（共计 28 分，以下各程序填空可能还有一些等价的写法，由各省赛区组织本省专家审定及上机验证，可以不上报 CCF NOI 科学委员会复核）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Pascal 语言 |  | | C++语言 |  | C 语言 | 分值 |
|  |  |  |  | |  |  |  |  |
| 1 | (1) |  | i\*i | | |  |  | 2 |
| . |  |  |  | |  |  |  |  |
| (2) | n div i | |  | | n/i |  | 3 |
|  |  |  |
|  |  |  |  | |  |  | |  |
|  | (3) | exit(a) |  | |  | return a | | 3 |
|  |  |  |  | |  | | |  |
|  | (4) | a mod b |  | | a % b | | | 3 |
|  |  |  |  | |  |  | |  |
|  | (5) | ans+gcd(a[i],a[j]) | | | | | | 3 |
|  |  |  |
|  | (1) | a[x]:=i |  | | a[x]=i | | | 3 |
| 2 |  | |
|  |  | |  |
| . |  |  |  | |  |  |  |  |
| (2) | i+1 | | | | | | 2 |
|  |
|  |  |  |
|  | (3) | R[a[i]] | | | | | | 3 |
|  |
|  |  |  |
|  |  |  |
|  | (4) | a[i] | | | | | | 3 |
|  |  |  |
|  | (5) | R[i] | | | |  |  | 3 |
|  |  |  |  | |  |  |  |  |