

第十一届全国青少年信息学奥林匹克联赛初赛试题

(提高组 C 语言 二小时完成)

●● 全部试题答案均要求写在答卷纸上, 写在试卷纸上一律无效 ●●

一、单项选择题 (共 10 题, 每题 1.5 分, 共计 15 分。每题有且仅有一个正确答案.)。

1. 字符串“ababacbab”和字符串“abcba”的最长公共子串是 ()。
A. abcbab B. cba C. abc D. ab E. bcba
2. 设全集 $I = \{a, b, c, d, e, f, g, h\}$, 集合 $A \cup B = \{a, b, c, d, e, f\}$, $A \cap C = \{c, d, e\}$, $A \cap \sim B = \{a, d\}$, 那么集合 $A \cap B \cap C$ 为 ()。
A. $\{c, e\}$ B. $\{d, e\}$ C. $\{e\}$ D. $\{c, d, e\}$ E. $\{d, f\}$
3. 以下二进制数的值与十进制数 23.456 的值最接近的是 ()。
A. 10111.0101 B. 11011.1111 C. 11011.0111 D. 10111.0111 E. 10111.1111
4. 完全二叉树的结点个数为 $4 * N + 3$, 则它的叶结点个数为 ()。
A. $2 * N$ B. $2 * N - 1$ C. $2 * N + 1$ D. $2 * N - 2$ E. $2 * N + 2$
5. 平面上有五个点 A(5, 3), B(3, 5), C(2, 1), D(3, 3), E(5, 1)。以这五点作为完全图 G 的顶点, 每两点之间的直线距离是图 G 中对应边的权值。图 G 的最小生成树中的所有边的权值综合为 ()。
A. 8 B. $7 + \sqrt{5}$ C. 9 D. $6 + \sqrt{5}$ E. $4 + 2\sqrt{2} + \sqrt{5}$
6. 下列设备中没有计算功能的是 ()。
A. 笔记本电脑 B. 掌上电脑 C. 智能手机
D. 电子计算器 E. 液晶显示器
7. Intel 的首颗 64 位处理器是 ()。
A. 8088 B. 8086 C. 80386 D. 80486 E. Pentium
8. 常见的邮件传输服务器使用 () 协议发送邮件。
A. HTTP B. SMTP C. TCP D. FTP E. POP3
9. 不能在 Linux 上使用的网页浏览器是 ()。
A. Internet Explore B. Netscape C. Opera D. Firefox E. Mozilla
10. 一位艺术史学家有 20000 幅 $1024 * 768$ 的真彩色图像, 如果将这些图像以位图形式保存在 CD 光盘上 (一张 CD 光盘的容量按 600M 计算), 大约需要 () 张 CD 光盘。
A. 1 B. 10 C. 100 D. 1000 E. 10000

二、不定项选择题（共 10 题，每题 1.5 分，共计 15 分。多选或少选均不得分）。

11. 设 $A = \text{true}$, $B = \text{false}$, $C = \text{false}$, $D = \text{true}$, 以下逻辑运算表达式值为真的有 ()。
- A. $(A \wedge B) \vee (C \wedge D)$ B. $((A \wedge B) \vee C) \wedge D$ C. $A \wedge ((B \vee C) \vee D)$
D. $(A \wedge (B \vee C)) \vee D$ E. $(A \vee B) \wedge (C \vee D)$
12. $(3725)_8 + (B)_{16}$ 的运算结果是 ()。
- A. $(3736)_8$ B. $(2016)_{10}$ C. $(11111100000)_2$ D. $(3006)_{10}$ E. $(7E0)_{16}$
13. 二叉树 T 的宽度优先遍历序列为 A B C D E F G H I, 已知 A 是 C 的父结点, D 是 G 的父结点, F 是 I 的父结点, 树中所有结点的最大深度为 3 (根结点深度设为 0), 可知 E 的父结点可能是 ()。
- A. A B. B C. C D. D E. F
14. 设栈 S 的初始状态为空, 元素 a, b, c, d, e, f, g 依次入栈, 以下出栈序列不可能出现的有 ()。
- A. a, b, c, e, d, f, g B. b, c, a, f, e, g, d C. a, e, c, b, d, f, g
D. d, c, f, e, b, a, g E. g, e, f, d, c, b, a
15. 下列外设接口中可以通过无线连接的方式连接设备的是 ()。
- A. USB 2.0 高速版 B. 红外 C. 蓝牙 D. 串口 E. IEEE 802.11g 无线网卡
16. 处理器 A 每秒处理的指令数是处理器 B 的 2 倍。某一特定程序 P 分别编译为处理器 A 和处理器 B 的指令, 编译结果处理器 A 的指令数是处理器 B 的 4 倍。已知程序 P 的算法时间复杂度为 $O(n^2)$, 如果处理器 A 执行程序 P 时能在一小时内完成的输入规模为 n, 则处理器 B 执行程序 P 时能在一小时内完成的输入规模为 ()。
- A. $4 * n$ B. $2 * n$ C. n D. $n / 2$ E. $n / 4$
17. 以下哪个 (些) 不是计算机的输出设备 ()。
- A. 鼠标 B. 显示器 C. 键盘 D. 扫描仪 E. 绘图仪
18. 以下断电之后将不能保存数据的有 ()。
- A. 硬盘 B. 寄存器 C. 显存 D. 内存 E. 高速缓存
19. 下列活动中属于信息学奥赛系列活动的是 ()。
- A. NOIP B. NOI C. IOI D. 冬令营 E. 国家队选拔赛
20. 下列关于高级语言的说法正确的有 ()。
- A. Ada 是历史上的第一个高级语言
B. Pascal 和 C 都是编译执行的高级语言
C. C++ 是历史上的第一个支持面向对象的语言
D. 编译器将高级语言程序转变为目标代码
E. 高级语言程序比汇编语言程序更容易从一种计算机移植到另一种计算机上

三. 问题求解（请在空格处填上答案，每空 5 分，共计 10 分）

1. 将数组{32, 74, 25, 53, 28, 43, 86, 47}中的元素按从小到大的顺序排列，每次可以交换任意两个元素，最少需要交换_____次。
2. 取火柴游戏的规则如下：一堆火柴有 N 根，A、B 两人轮流取出。每人每次可以取 1 根或 2 根，最先没有火柴可取的人为败方，另一方为胜方。如果先取者有必胜策略则记为 1，先取者没有必胜策略记为 0。当 N 分别为 100, 200, 300, 400, 500 时，先取者有无必胜策略的标记顺序为_____（回答应为一个由 0 和/或 1 组成的字符串）。

四. 阅读程序（共 4 题，每题 8 分，共计 32 分）

```
1. #include <stdio.h>
int main() {
    int a, b, c, p, q, r[3];
    scanf("%d%d%d", &a, &b, &c);
    p = a / b / c;
    q = b - c + a + p;
    r[0] = a * p / q * q;
    r[1] = r[0] * (r[0] - 300);
    if (3 * q - p % 3 <= r[0] && r[2] == r[2])
        r[1] = r[r[0] / p % 2];
    else
        r[1] = q % p;
    printf("%d\n", r[0] - r[1]);
    return 0;
}
```

输入: 100 7 3

输出: _____

```
2. #include <stdio.h>
#include <math.h>
int a[50];
void work(int p, int r) {
    if (p < r) {
        int i = p - 1, j, temp;
        for (j = p; j < r; j++) {
            if (a[j] >= a[r]) {
                i++;
                temp = a[i]; a[i] = a[j]; a[j] = temp;
            }
        }
    }
}
```

```

        temp = a[i + 1]; a[i + 1] = a[r]; a[r] = temp;
        work(p, i);
        work(i + 2, r);
    }
}
int main() {
    int n, i, sum = 0;
    scanf("%d", &n);
    for (i = 0; i < n; i++) scanf("%d", &(a[i]));
    work(0, n - 1);
    for (i = 0; i < n - 1; i++) sum += abs(a[i + 1] - a[i]);
    printf("%d\n", sum);
    return 0;
}

```

输入: 10 23 435 12 345 3123 43 456 12 32 -100

输出: _____

```

3. #include<stdio.h>
#include<string.h>
int main(){
    char str[60];
    int len, i, j, chr[26];
    char mmin = 'z';
    scanf("%s", str);
    len = strlen(str);
    for (i = len - 1; i >= 1; i--)
        if (str[i - 1] < str[i]) break;
    if (i == 0) {
        printf("No result!\n");
        return 0;
    }
    for (j = 0; j < i - 1; j++) putchar(str[j]);
    memset(chr, 0, sizeof(chr));
    for (j = i; j < len; j++) {
        if (str[j] > str[i - 1] && str[j] < mmin)
            mmin = str[j];
        chr[str[j] - 'a']++;
    }
    chr[mmin - 'a']--;
    chr[str[i - 1] - 'a']++;
    putchar(mmin);
    for(i = 0; i < 26; i++)
        for(j = 0; j < chr[i]; j++)
            putchar(i + 'a');
}

```

```

        putchar( '\n' );
        return 0;
    }

```

输入: zzyzcccbbbaaa

输出: _____

```

4. #include <stdio.h>
    long g(long k) {
        if (k <= 1) return k;
        return (2002 * g(k - 1) + 2003 * g(k - 2)) % 2005;
    }
    int main() {
        long n;
        scanf("%ld", &n);
        printf("%ld\n", g(n));
        return 0;
    }

```

输入: 2005

输出: _____

五. 完善程序 (前 5 空, 每空 2 分, 后 6 空, 每空 3 分, 共 28 分)

1. 木材加工

题目描述:

木材厂有一些原木,现在想把这些木头切割成一些长度相同的小段木头(木头有可能有剩余),需要得到的小段的数目是给定了的。当然,我们希望得到的小段越长越好,你的任务是计算能够得到的小段木头的最大长度。

木头长度的单位是 **cm**。原木的长度都是正整数,我们要求切割得到的小段木头的长度也是正整数。

输入:

第一行是两个正整数 N 和 K ($1 \leq N \leq 10000$, $1 \leq K \leq 10000$), N 是原木的数目, K 是需要得到的小段的数目。

接下来的 N 行,每行有一个 1 到 10000 之间的正整数,表示一根原木的长度。

输出:

输出能够切割得到的小段的最大长度。如果连 1cm 长的小段都切不出来,输出 "0"。

输入样例:

```

3 7
232
124
456

```

输出样例:

```

114

```

程序:

```

#include <stdio.h>

```

```

int n, k, len[10000];
int isok(int t) {
    int num = 0, i;
    for (i = 0; i < n; i++) {
        if (num >= k) break;
        num = ①;
    }
    if (②) return 1;
    else return 0;
}
int main() {
    int i, left, right, mid;
    scanf("%d%d", &n, &k);
    right = 0;
    for (i = 0; i < n; i++) {
        scanf("%d", &(len[i]));
        if (right < len[i]) right = len[i];
    }
    right++;
    ③;
    while (④ < right) {
        mid = (left + right) / 2;
        if (⑤) right = mid;
        else left = mid;
    }
    printf ("%d\n", left);
    return 0;
}

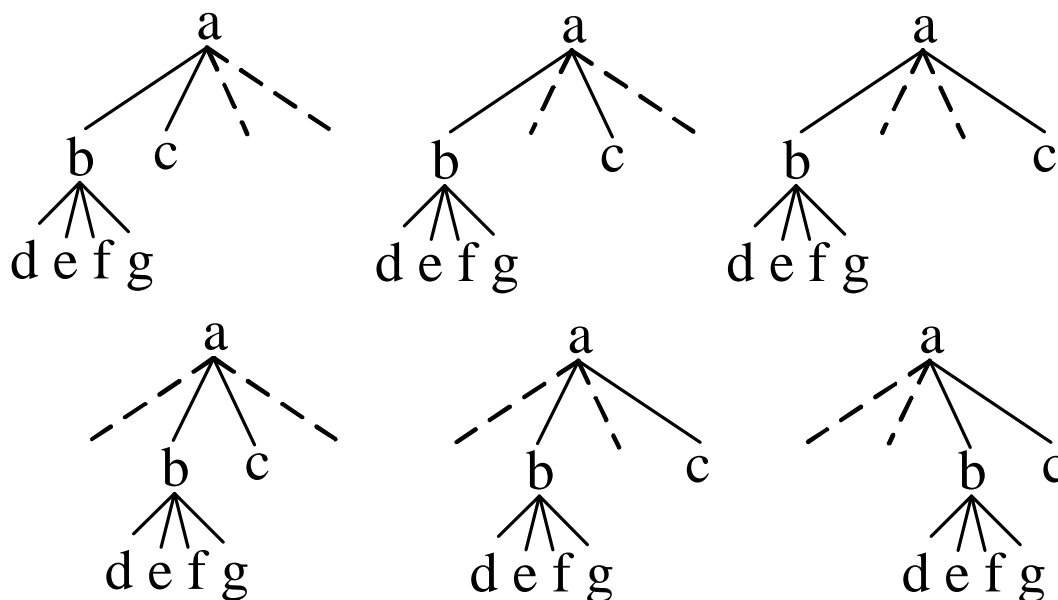
```

2. N 叉树

题目描述：

我们都了解二叉树的先根遍历，中根遍历和后根遍历。当知道先根遍历的结果和中根遍历结果的时候，我们可以唯一的确定二叉树；同样的，如果知道了后根遍历的结果和中根遍历结果，二叉树也是唯一确定的。但是如果只知道先根遍历和后根遍历的结果，二叉树就不是唯一的了。但是我们可以计算满足条件的不同二叉树的一共有多少个。这不是一个很困难的问题，稍微复杂一点，我们把这个问题推广到 N 叉树。

我们用小写英文字母来表示 N 叉树的结点，不同的结点用不同的字母表示。比如，对于 4 叉树，如果先根遍历的结果是 abdefgc，后根遍历的结果是 defgbca，那么我们可以得到 6 个不同的 4 叉树（如下图）。



输入:

输入数据包括3行。

第一行是一个正整数 N ($1 \leq N \leq 20$), 表示我们要考虑 N 叉树。

第二行和第三行分别是两个字符串序列, 分别表示先根遍历和后根遍历的结果。

输出:

输出不同的 N 叉树的数目。题目中给的数据保证得到的结果小于 2^{31} 。

输入样例:

```
4
abdefgc
defgbca
```

输出样例:

```
6
```

程序:

```
#include <stdio.h>
#include <string.h>
char str1[100], str2[100];
int N;
long com[100][100];
long getcom(int x, int y) {
    if (y == 0 || x == y) ①;
    else if (com[x][y] != 0) return com[x][y];
    else {
        com[x][y] = getcom(x - 1, y) + ②;
        return com[x][y];
    }
}
long count(int a, int b, int c){
```

```
long sum = 1;
int k = 0;
int s = a + 1, t = c, p;
if (a == b) return 1;
while(s <= b){
    p = t;
    while(str1[s] != str2[t]) t++;
    sum = sum * count(s, s + t - p, p);
    s = ③;
    ④;
    k++;
}
return ⑤ * getcom(N, k);
}
int main(){
    int len;
    scanf("%d", &N);
    scanf("%s%s", str1, str2);
    len = strlen(str1);
    printf("%ld\n", count(⑥));
    return 0;
}
```


赛区_____市_____学校 姓名_____

===== 密 封 线 =====

第十一届全国青少年信息学奥林匹克联赛初赛试题

提高组答卷纸

阅													
---	--	--	--	--	--	--	--	--	--	--	--	--	--

===== 以下由考生填写 =====

答卷部分

一. 单项选择题 (共 10 题, 每题 1.5 分, 共计 15 分。每题有且仅有一个正确答案.)。

题号	1	2	3	4	5	6	7	8	9	10
选择										

二. 不定项选择题 (共 10 题, 每题 1.5 分, 共计 15 分。多选或少选均不得分)。

题号	11	12	13	14	15	16	17	18	19	20
选择										

三. 问题解答 (每题 5 分, 共 10 分)

1. 答: _____

2. 答: _____

四. 阅读程序, 并写出程序的正确运行结果: (每题 8 分, 共 32 分)

(1) 程序的运行结果是:

(2) 程序的运行结果是:

_____赛区 _____市 _____学校 姓名_____

===== 密 封 线 =====

(3) 程序的运行结果是：

(4) 程序的运行结果是：

五. 完善程序 (前 5 空, 每空 2 分, 后 6 空, 每空 3 分, 共 28 分)

C 语言

=====

1.

(1) _____

(2) _____

(3) _____

(4) _____

(5) _____

2.

(1) _____

(2) _____

(3) _____

(4) _____

(5) _____

(6) _____