

全国青少年信息学奥林匹克联赛初赛模拟考试（二）
（ 普及组 C++ 语言 两小时完成 ）

●● 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 20 题，每题 1.5 分，共计 30 分。每题有且仅有一个正确选项。）

1、一棵二叉树共有 25 个结点，其中 5 个是叶子结点，则度为 1 的结点数为（ ）

A、16 B、10 C、6 D、4

2、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

3、已知 $a=001111000B$ 和 $b=2FH$ ，则两者比较的正确不等式是（ ）

A、 $a>b$ B、 $a=b$ C、 $a<b$ D、不能比较

4、设任意一个十进制整数为 D，转换成二进制数为 B。根据数制的概念，下列叙述中正确的是（ ）

A、数字 B 的位数 < 数字 D 的位数 B、数字 B 的位数 \leq 数字 D 的位数

C、数字 B 的位数 \geq 数字 D 的位数 D、数字 B 的位数 > 数字 D 的位数

5、已知二叉树后序遍历序列是 dabec，中序遍历序列是 debac，它的前序遍历序列是

A、acbed B、decab C、deabc D、cedba

6、有一个由 4000 个整数构成的顺序表，假定表中的元素已经按升序排列，采用二分查找定位一个元素。则最多需要几次比较就能确定是否存在所查找的元素：

A、11 次 B、12 次 C、13 次 D、14 次

7、假设某台式计算机内存储器的容量为 1KB，其最后一个字节的地址是（ ）

A、1023H B、1024H C、0400H D、03FFH

8、假设某台式计算机的内存储器容量为 256MB，硬盘容量为 20GB。硬盘的容量是内存容量的（ ）

A、40 倍 B、60 倍 C、80 倍 D、100 倍

9、若有以下定义：

```
struct link { int data; struct link *next; } a,b,c,*p,*q; 且变量
```

a 和 b 之间已有如下图所示的链表结构： 指针 p 指向变量 a, q 指向变量 c。

则能够把 c 插入到 a 和 b 之间并形成新的链表的语句组是：

A、a.next=c; c.next=b; B、p.next=q; q.next=p.next;

C、p->next=&c; q->next=p->next;

D、(*p).next=q; (*q).next=&b;

10、随着 Internet 的发展，越来越多的计算机感染病毒的可能途径之一是（
）。

A、从键盘上输入数据 B、通过电源线

C、所使用的光盘表面不清洁

D、通过 Internet 的 E-mail，附着在电子邮件的信息中

11、对长度为 n 的线性表排序，在最坏情况下，比较次数不是 $n(n-1)/2$ 的排序方法是()

A、快速排序 B、冒泡排序 C、直接插入排序 D、堆排序

12、无符号二进制整数 111110 转换成十进制数是（ ）

A、62 B、60 C、58 D、56

13、无向完全图是图中每对顶点之间都恰有一条边的简单图。已知无向完全图 G 有 7 个顶点，则它共有（ ）条边。

A、7 B、21 C、42 D、49

14、下列关于栈的叙述正确的是（ ）

A、栈按“先进先出”组织数据 B、栈按“先进后出”组织数据

C、只能在栈底插入数据 D、不能删除数据

15、某二叉树共有 7 个结点，其中叶子结点只有 1 个，则该二叉树的深度为（假设根结点在第 1 层）（ ）

A、3 B、4 C、6 D、7

二、阅读程序（判断题正确的填√，错误的填×；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

1、

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 int main() {
5     char str[105]={0};
6     fgets(str,105,stdin);
7     for( int i = 0; i < strlen(str); i++) {
8         if( str[i] >= 'a' && str[i] <='z')
9             if( str[i] == 'z' )
10                 str[i] = 'a';
11         else
12             str[i] += 1;
13     }
14     cout << str << endl;
15     return 0;
16 }
```

判断题

- 1、本程序是将小写字母替换成大写字母（ ）
- 2、若输入的字符串长度超过 105 不会出现问题（ ）
- 3、若将第 8 行的&&改成||有可能出现不可见字符（ ）
- 4、把 12 行改成 str[i] +=32,那么会将除 z 以外的小写字母转成大写（ ）

选择题

5、输入数据 abcdz，那么输出的结果为（）

- A. ABCDZ B. BCDEa C. bcdea D. BCDEA

6、输入数据 gdkkn bzs，那么输出结果为（）

- A. hello B. GDKKN C. hello cat D. GDKKN BZS

2、

```

1 #include <iostream>
2 using namespace std;
3 bool judge(int n){
4     int raw = n;
5     int r = 0;
6     while(n > 0){
7         r = r*10+n%10;
8         n /= 10;
9     }
10    if( r == raw)
11        return true;
12    else
13        return false;
14 }
15 int main(){
16     int n ;
17     cin >> n;
18     int count = 0;
19     for( int i = 1; i <= n; i++)
20         if( judge(i) )
21             ++count;
22     cout << count << endl;

```

```
23 return 0;
}
```

判断题

- 1、若将第 6 行 `n>0` 改成 `n != 0` 效果相同 ()
- 2、若将函数 `judge(int n)` 改成 `judge(int& n)` 可能造成死循环 ()
- 3、如果 `n` 小于 10, `count` 有可能等于 0 ()
- 4、若第 5 行 `int r = 0` 改成 `int r`, 结果不会改变 ()

选择题

- 5、输入数据 10, 那么输出的结果为 ()
A. 10 B. 55 C. 0 D. 9
- 6、输入数据 30, 那么输出结果为 ()
A. 30 B. 11 C. 2 D. 9

3、

```
1 #include<cstdio>
2 using namespace std;
3 int f(int n,int m){
4     if(n==0)
5         return m;
6     if(m==0)
7         return n;
8     return f(n%m,n)+f(m,m%n);
9 }
10 int main() {
11     int n,m;
12     scanf("%d%d",&n,&m);
13     printf("%d\n",f(n,m));
14     return 0;
```

15 }

判断题

- 1、如果输入 0 0 那么结果为 0 ()
- 2、如果 12 行被注释，那么结果为 0 ()

选择题

- 3、输入 9 3 结果为 ()。
A. 12 B. 11 C. 3 D. 15
- 4、输入 9 5 结果为 ()。
A. 14 B. 29 C. 17 D. 1
- 5、如果 n 为质数，那么递归次数 ()。
A. 大于 n B. 等于 n C. 小于 n D. 不确定
- 6、(4 分)输入 11 7 结果为 ()
A. 11 B. 7 C. 18 D. 45

三、完善程序（每小题 3 分，总共 30 分）

- 1、给定一个有序（非降序）数组 A，可含有重复元素，求最小的 i 使得 A[i] 等于 target，不存在则返回-1。

```
#include <iostream>
using namespace std;
int BinarySearch(int A[],int n,int target){
    if(n <= 0){
        return -1;
    }
    int start = 0,end = n-1;
    while(start < end){
        int mid = __①__;
        if(A[mid] < target){
            start = __②__;
        }
    }
}
```

```

        }
        else{
            __③__
        }
    }
    if(__④__) {
        return -1;
    }
    else{
        return start;
    }
}

int main() {
    int A[] = {2, 3, 4, 4, 4, 4, 4, 5, 6, 7, 8};
    cout<<BinarySearch(__⑤__)<<endl;
    return 0;
}

```

1) ①处应填 ()

- | | |
|--------------------|-----------------------|
| A. start + end | B. (start + end) >> 1 |
| C. (start-end) / 2 | D. end - 1 |

2) ②处应填 ()

- | | | | |
|--------|----------|----------|------------|
| A. mid | B. mid+1 | C. mid-1 | D. start+1 |
|--------|----------|----------|------------|

3) ③处应填 ()

- | | | | |
|--------|----------|----------|------------|
| A. mid | B. mid+1 | C. mid-1 | D. start+1 |
|--------|----------|----------|------------|

4) ④处应填 ()

- | | |
|-------------------|-------------------|
| A. A[mid]==target | B. A[mid]>=target |
| C. A[mid]!=target | D. A[mid]<target |

5) 如果要搜索 4 这个值⑤处应填 ()

- | | | | |
|-------------|-------------|-------------|-------------|
| A. A, 11, 4 | B. A, 10, 4 | C. A, 4, 11 | D. A, 4, 10 |
|-------------|-------------|-------------|-------------|

2、编写程序实现两种不同进制之间的数据转换。输入数据共有三行，第一行是一个正整数，表示需要转换的数的进制 n ($2 \leq n \leq 16$)，第二行是一个 n 进制数，

若 $n > 10$ 则用大写字母 A~F 表示数码 10~15，并且该 n 进制数对应的十进制的值不超过 1000000000，第三行也是一个正整数，表示转换之后的数的进制 m ($2 \leq m \leq 16$)。

```
#include <iostream>
#include <cmath>
#include <cstring>
using namespace std;

char c[32];
char t[32];

void change(int n,int base)
{
    int index = 0;
    while(n>0)
    {
        if( n%base >= 10)
            t[index++] = __①__;
        else
            t[index++] = __②__;
        __③__;
    }
}

int main()
{
    int base,taget;
    cin >> base >> c >> taget;
    int len = strlen(c);
    int value = 0;
    for( int i = 0; i < len; i++)
    {
        if( c[i] < 'A' )
            value += (__④__)*pow(base, len-1-i);
```



```

        else
            value += (__⑤__)*pow(base, len-1-i);
    }
    change(value, taget);
    int t_len = strlen(t);
    for(int i = t_len-1; i>= 0;i--)
    {
        cout << t[i];
    }
    cout << endl;
    return 0;
}

```

1) ①处应填 ()

- | | |
|-------------------|-------------------|
| A. n%base-10 | B. n%base-10+ 'a' |
| C. n%base-10+ 'A' | D. n%base+ 'A' |

2) ②处应填 ()

- | | |
|----------------|-------------------|
| A. n%base+ '0' | B. n%base-10+ '0' |
| C. n%base-10+0 | D. n%base+ '\0' |

3) ③处应填 ()

- | | | | |
|--------------|--------------|--------------|--------------|
| A. n %= base | B. n /= base | C. n -= base | D. n = base |
|--------------|--------------|--------------|--------------|

4) ④处应填 ()

- | | |
|--------------|--------------|
| A. c[i]+ '0' | B. c[i]- 0 |
| C. c[i]+0 | D. c[i]- '0' |

5) ⑤处应填 ()

- | | |
|------------------|------------------|
| A. c[i]+10 | B. c[i]- 'a' +10 |
| C. c[i]- '0' +10 | D. c[i]- 'A' +10 |