

全国青少年信息学奥林匹克联赛初赛模拟考试（一）  
（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 15 题，每题 2 分，共计 30 分。每题有且仅有一个正确选项。）

1、世界上公认的第一台电子计算机诞生在（ ）

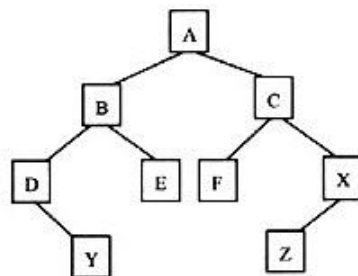
A、中国 B、美国 C、英国 D、日本

2、将 (2, 6, 10, 17) 分别存储到某个地址区间为 0~10 的哈希表中，如果哈希函数  $h(x) = ( )$ ，将不会产生冲突，其中  $a \bmod b$  表示  $a$  除以  $b$  的余数。

A、 $x \bmod 11$  B、 $x^2 \bmod 11$

C、 $2x \bmod 11$  D、 $\lfloor \sqrt{x} \rfloor \bmod 11$ ，其中  $\lfloor \sqrt{x} \rfloor$  表示  $\sqrt{x}$  下取整

3、下列二叉树，先序遍历



A、DYBEAFCZX

B、YDEBFZXCA

C、ABDYECFXZ

D、ABCDEFXYZ

4、一棵二叉树中共有 80 个叶子结点与 70 个度为 1 的结点，则该二叉树中的总结点数为（ ）

A、219

B、229

C、230

D、231

5、假设线性表的长度为  $n$ ，则在最坏情况下，冒泡排序需要的比较次数为

A、 $\log_2 n$

B、 $n^2$

C、 $O(n^{1.5})$

D、 $n(n-1)/2$

6、计算机技术应用广泛，以下属于科学计算方面的是（ ）

A、图像信息处理      B、视频信息处理

C、火箭轨道计算      D、信息检索

7、冯诺依曼 (Von Neumann) 在总结研制 ENIAC 计算机时, 提出两个重要的改进是 (      )

A、引入 CPU 和内存储器的概念      C、采用机器语言和十六进制

B、采用二进制和存储程序控制的概念      D、采用 ASCII 编码系统

8、如果进栈序列为  $e_1, e_2, e_3, e_4$ , 则可能的出栈序列是 ( )

A、 $e_3, e_1, e_4, e_2$       B、 $e_2, e_4, e_3, e_1$       C、 $e_3, e_4, e_1, e_2$       D、任意顺序

9、在标准 ASCII 码表中, 英文字母 a 和 A 的码值之差的十进制值是 (      )

A、20      B、32      C、-20      D、-32

10、标准 ASCII 码字符集有 128 个不同的字符代码, 它所使用的二进制位数是 (      )

A、6      B、7      C、8      D、16

11、十进制数 39 转换成无符号二进制整数是 (      )

A、100011      B、100101      C、100111      D、100011

12、微机中, 西文字符所采用的编码是 (      )

A、EBCDIC 码      B、ASCII 码      C、国标码      D、BCD 码

13、下面叙述正确的是

A、算法的执行效率与数据的存储结构无关

B、算法的空间复杂度是指算法程序中指令(或语句)的条数

C、算法的有穷性是指算法必须能在执行有限个步骤之后终止

D、以上三种描述都不对

14、某二叉树有 5 个度为 2 的结点, 则该二叉树中的叶子结点数是 (      )

A、10      B、8      C、6      D、4

15、计算机的存储器中，组成一个字节 (Byte) 的二进制位 (bit) 个数是 (      )

A、4      B、8      C、16      D、32

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 using namespace std;
3 int gcd(int a,int b){
4     int r = a%b;
5     while( r ){
6         a = b;
7         b = r;
8         r = a%b;
9     }
10    return b;
11 }
12 int main(){
13     int a,b;
14     cin >> a >> b;
15     cout << gcd(a,b) << endl;
16     return 0;
17 }
```

## 判断题

- 1、输入浮点数，程序能够得出正确结果 ( )
- 2、若将第 5 行的 while( r ) 改为 while ( r != 0 ) 结果不会改变 ( )
- 3、若将第 15 行的函数调用 gcd (a,b) 改为 gcd (b,a) 效果相同 ( )

4、若 b 的输入为 0，则程序将会异常结束（）

## 选择题

5、入数据 10 5，那么输出的结果为（）

A. 50      B. 1      C. 15      D. 5

6、若输入数据 10 -3，那么输出结果为（）

A. -1      B. 1      C. -2      D. 2

## 2、

```

1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n;
5     cin >> n;
6     for( int i = 1; i <= n; i++) {
7         for( int j = 1; j <= n-i; j++)
8             cout << " ";
9         for( int j = 1; j <= 2*i-1; j++)
10             cout << "*";
11         cout << endl;
12     }
13     return 0;
14 }
```

## 判断题

1、以上程序输出的是一个\*号等腰三角形（）

2、若将第 9 行的  $j \leq 2*i-1$  改为  $j < 2*i$  效果相同（）

3、若将第 7，8 行代码注释掉，那么会得到一个直接三角形（）

4、若 n 的输入过大，图形可能无法得到三角形效果（）

## 选择题

- 5、若  $n=20$ ，那么最长的那行将会有 ( ) 个星号。  
A. 40      B. 41      C. 39      D. 42
- 6、若第 9 行的  $j \leq 2*i-1$  改为  $j \leq i$ ，输入  $n=10$ ，那么图形总共有 ( ) 个星号。  
A. 10      B. 11      C. 100      D. 55

## 3、

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n;
5     cin >> n;
6     int sum = 0;
7     for(int i = 1; i <=n; i++) {
8         if( n%i == 0 )
9             sum += i;
10    }
11    cout << sum << endl;
12    return 0;
13 }
```

## 判断题

- 1、如果 sum 没有初始化，那么结果不确定 ( )
- 2、若将第 7 行  $i \leq n$  改成  $i < n$ ，结果没有区别 ( )

## 选择题

- 3、若  $n=10$ ，那么结果为 ( )。  
A. 18      B. 10      C. 9      D. 55

4、若  $n$  为质数，则结果为 ( )。

- A.  $n/2$       B.  $n/2+1$       C.  $n+1$       D.  $(n-1)/2$

5、如果  $i$  从 2 开始， $i$  小于  $n$  那么， $n = 21$  时结果为多少 ( )。

- A. 32      B. 10      C. 31      D. 21

6、(4 分)如果有  $m$  个数要求这个  $sum$ ，那么请问时间复杂度是多少 ( )

- A.  $O(n)$       B.  $O(n^2)$       C.  $O(m*n)$       D.  $O(m^2)$

### 三、完善程序 (每小题 3 分，总共 30 分)

1、给你一个整数  $n$ ，按要求输出  $n*n$  的回型矩阵

例如：

输入：4

输出：

1   2   3   4

12 13 14 5

11 16 15 6

10   9   8   7

```
#include <iostream>
```

```
using namespace std;
```

```
int a[16][16];
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cin >> n;
```

```
    int p = 0, q = n-1;
```

```
    int num = 1;
```

```
    while( p < q )
```

```
    {
```

```
        for(int i = p; i < q; i++)
```

```

{
    __①__ ;
}
for(int i = p; i < q; i++)
{
    a[i][q] = num++;
}
for(int i = q; i > p; i--)
{
    __②__ ;
}
for(int i = q; i > p; i--)
{
    a[i][p] = num++;
}
__③__
__④__ ;
}
if( __⑤__ )
{
    a[n/2][n/2] = num++;
}
for( int i = 0; i < n; i++)
{
    for( int j = 0; j < n; j++)
    {
        cout << a[i][j] << " ";
    }
    cout << endl;
}
return 0;
}

```

1) ①处应填 ( )

- A.  $a[p][i] = num++$                       B.  $a[q][i] = num++$   
 C.  $a[i][p] = num++$                       D.  $a[i][q] = num++$

2) ②处应填 ( )

- A.  $a[p][i] = num++$                       B.  $a[q][i] = num++$   
 C.  $a[i][p] = num++$                       D.  $a[i][q] = num++$

3) ③处应填 ( )

- A.  $p++$                       B.  $q++$                       C.  $p--$                       D.  $q--$

4) ④处应填 ( )

- A.  $p++$                       B.  $q++$                       C.  $p--$                       D.  $q--$

5) ⑤处应填 ( )

- A.  $n/2$                       B.  $n>num$                       C.  $n\%2$                       D.  $(n-1)\%2$

2、在大规模数据处理中，经常会遇到的一类问题：在海量数据中找出出现频率最好的前 k 个数，或者从海量数据中找出最大的前 k 个数，这类问题通常被称为 top K 问题。例如，在搜索引擎中，统计搜索最热门的 10 个查询词；在歌曲库中统计下载最高的前 10 首歌等。下面的代码是经典的堆排序过程。请补充完整整个程序。

```
#include <iostream>
#include <algorithm>
using namespace std;
void heap_adjust(int *arr,int i,int size)
{
    int lchild = 2*i;
    int rchild = 2*i+1;
    int max = i;
    if( i <= size/2 )
    {
        if( ① )
        {
            max = lchild;
```



```

    }
    if( ② )
    {
        max = rchild;
    }
    if( ③ )
    {
        swap(arr[i], arr[max]);
        ④ ;
    }
}
}

```

```

void build_heap(int *arr, int size)
{
    for( int i = size/2; i >=1; i--)
    {
        heap_adjust(arr, i, size);
    }
}

```

```

void heap_sort(int *arr, int size)
{
    int i;
    build_heap(arr, size);
    for( i = size; i >=1; i--)
    {
        swap(arr[1], arr[i]);
        ⑤ ;
    }
}

```

```

void print_array(int *a, int size)

```

```

{
    for(int i = 1; i <= size; i++)
    {
        cout << a[i] <<" ";
    }
    cout << endl;
}

```

```

int main()
{
    int n;
    cin >> n;
    int a[n+1];
    for( int i = 1; i <= n; i++)
    {
        cin >> a[i];
    }
    print_array(a,n);
    heap_sort(a,n);
    print_array(a,n);
    return 0;
}

```

1) ①处应填 ( )

- A. lchild > size && arr[lchild] > arr[max]
- B. lchild <= size && arr[lchild] > arr[max]
- C. rchild > size && arr[rchild] > arr[max]
- D. rchild <= size && arr[rchild] > arr[max]

2) ②处应填 ( )

- A. lchild > size && arr[lchild] > arr[max]
- B. lchild <= size && arr[lchild] > arr[max]
- C. rchild > size && arr[rchild] > arr[max]
- D. rchild <= size && arr[rchild] > arr[max]

3) ③处应填 ( )

- A. max                      B. max != i                      C. max%2                      D. max <= size

4) ④处应填 ( )

- A. heap\_adjust(arr, i, size)                      B. heap\_adjust(arr, max, size)  
C. heap\_adjust(arr, max-1, size)                      D. heap\_adjust(arr, max, size/2)

5) ⑤处应填 ( )

- A. heap\_adjust(arr, 1, i)                      B. heap\_adjust(arr, 1, i+1)  
C. heap\_adjust(arr, 1, i-1)                      D. heap\_adjust(arr, 1, size)