

# 2014年第二十二届NOIP信奥赛普及组初赛C++试题

题目总数：28 总分数：100

## 一、单项选择题

第1题 单选题

以下哪个是面向对象的高级语言（ ）。

- A. 汇编语言
- B. C++
- C. Fortran
- D. Basic

答案 B

第2题 单选题

1TB 代表的字节数量是（ ）。

- A.  $2^{10}$ 次方
- B.  $2^{20}$ 次方
- C.  $2^{30}$ 次方
- D.  $2^{40}$ 次方

答案 D

第3题 单选题

二进制数 00100100 和 00010101 的和是（ ）。

- A. 00101000
- B. 001010100
- C. 01000101
- D. 00111001

答案 D

第4题 单选题

以下哪一种设备属于输出设备（ ）。

- A. 扫描仪

- B. 键盘
- C. 鼠标
- D. 打印机

答案 D

第 5 题 单选题

下列对操作系统功能的描述最为完整的是（ ）。

- A. 负责外设与主机之间的信息交换
- B. 负责诊断机器的故障
- C. 控制和管理计算机系统的各种硬件和软件资源的使用
- D. 将源程序编译成目标程序

答案 C

第 6 题 单选题

CPU、存储器、I/O设备是通过（ ）连接起来的。

- A. 接口
- B. 总线
- C. 控制线
- D. 系统文件

答案 B

第 7 题 单选题

断电后会丢失数据的存储器是（ ）。

- A. RAM
- B. ROM
- C. 硬盘
- D. 光盘

答案 A

第 8 题 单选题

以下哪一种是属于电子邮件收发的协议（ ）。

- A. SMTP
- B. UDP
- C. P2P
- D. FTP

答案 A

第 9 题 单选题

下列选项中不属于图像格式的是（ ）。

- A. JPEG 格式
- B. TXT 格式
- C. GIF 格式
- D. PNG 格式

答案 B

第 10 题 单选题

链表不具有的特点是（ ）。

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

答案 B

第 11 题 单选题

列各无符号十进制整数中，能用八位二进制表示的数中最大的是（ ）。

- A. 296
- B. 133
- C. 256
- D. 199

答案 D

第 12 题 单选题

下列几个32位IP地址中，书写错误的是（ ）。

- A. 162.105.128.27
- B. 192.168.0.1
- C. 256.256.129.1
- D. 10.0.0.1

答案 C

第 13 题 单选题

要求以下程序的功能是计算： $s = 1 + 1/2 + 1/3 + \dots + 1/10$ 。

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

```
int main() {
    int n;
    float s;
    s = 1.0;
    for (n = 10; n > 1; n--)
        s = s + 1 / n;
    cout << s << endl;
    return 0;
}
```

程序运行后输出结果错误，导致错误结果的程序行是（ ）。

- A. s = 1.0;
- B. for (n = 10; n > 1; n--)
- C. s = s + 1 / n;
- D. cout << s << endl;

答案 C

#### 第 14 题 单选题

设变量x为float型且已赋值，则以下语句中能将x中的数值保留到小数点后两位，并将第三位四舍五入的是（ ）。

- A. x = (x \* 100) + 0.5 / 100.0;
- B. x = (x \* 100 + 0.5) / 100.0;
- C. x = (int) (x \* 100 + 0.5) / 100.0;
- D. x = (x / 100 + 0.5) \* 100.0;

答案 C

#### 第 15 题 单选题

有以下程序：

```
#include <iostream>
using namespace std;
int main() {
    int s, a, n;
    s = 0;
    a = 1;
    cin >> n;
    do {
        s += 1;
        a -= 2;
    } while (a != n);
    cout << s << endl;
    return 0;
}
```

```
}
```

若要使程序的输出值为2，则应该从键盘给n输入的值是（ ）。

- A. -1
- B. -3
- C. -5
- D. 0

答案 B

第 16 题 单选题

一棵具有 5 层的满二叉树中结点数为（ ）。

- A. 31
- B. 32
- C. 33
- D. 16

答案 A

第 17 题 单选题

有向图中每个顶点的度等于该顶点的（ ）。

- A. 入度
- B. 出度
- C. 入度与出度之和
- D. 入度与出度之差

答案 C

第 18 题 单选题

设有100个数据元素，采用折半搜索时，最大比较次数为（ ）。

- A. 6
- B. 7
- C. 8
- D. 10

答案 B

第 19 题 单选题

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>0。s=a;

```
for (b = 1; b <= c; b++)
```

```
s += 1;
```

则与上述程序段功能等价的赋值语句是（ ）。

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

答案 B

第 20 题 单选题

计算机界的最高奖是（ ）。

- A. 菲尔兹奖
- B. 诺贝尔奖
- C. 图灵奖
- D. 普利策奖

答案 C

## 二、问题求解

第 21 题 填空题

把M个同样的球放到N个同样的袋子里，允许有的袋子空着不放，问共有多少种不同的放置方法？（用K表示）。

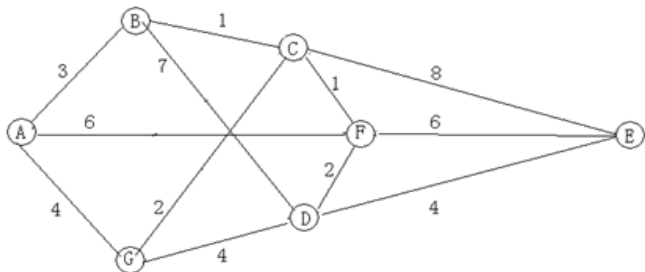
例如：M=7，N=3时，K=8；在这里认为（5,1,1）和（1,5,1）是同一种放置方法。

问：M=8，N=5时，K=\_\_\_\_\_。

答案 18

第 22 题 填空题

如图所示，图中每条边上的数字表示该边的长度，则从A到E的最短距离是\_\_\_\_\_。



答案 11

### 三、阅读程序写结果

#### 第 23 题 填空题

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a, b, c, d, ans;
6      cin >> a >> b >> c;
7      d = a - b;
8      a = d + c;
9      ans = a * b;
10     cout << "Ans = " << ans << endl;
11     return 0;
12 }
```

输入：2 3 4

输出：\_\_\_\_\_

答案 Ans = 9

#### 第 24 题 填空题

```
1  #include <iostream>
2  using namespace std;
3
4  int fun(int n) {
5      if (n == 1)
6          return 1;
7      if (n == 2)
8          return 2;
9      return fun(n - 2) - fun(n - 1);
10 }
11
12 int main() {
13     int n;
14     cin >> n;
15     cout << fun(n) << endl;
16     return 0;
17 }
```

输入：7

输出：\_\_\_\_\_

答案 -11

第 25 题 填空题

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      string st;
8      int i, len;
9      getline(cin, st);
10     len = st.size();
11     for (i = 0; i < len; i++){
12         if (st[i] >= 'a' && st[i] <= 'z')
13             st[i] = st[i] - 'a' + 'A';
14     }
15     cout << st << endl;
16     return 0;
17 }
```

输入: Hello, my name is Lostmonkey.

输出: \_\_\_\_\_

答案 HELLO, MY NAME IS LOSTMONKEY.

第 26 题 填空题

```
1  #include <iostream>
2  using namespace std;
3
4  const int SIZE = 100;
5
6  int main()
7  {
8      int p[SIZE];
9      int n, tot, i, cn;
10     tot = 0;
11     cin >> n;
12     for (i = 1; i <= n; i++)
13         p[i] = 1;
14     for (i = 2; i <= n; i++){
15         if (p[i] == 1)
16             tot++;
17         cn = i * 2;
18         while (cn <= n) {
19             p[cn] = 0;
20             cn += i;
21         }
22     }
```



```
23 |     cout << tot << endl;
24 |     return 0;
25 | }
```

输入： 30

输出： \_\_\_\_\_

答案 10

## 四、完善程序

第 27 题 问答题

完善程序：（数字删除）下面程序的功能是将字符串中的数字字符删除后输出。请填空。（每空3分，共12分）

```
#include <iostream>
```

```
using namespace std;
```

```
int delnum(char *s) {
    int i, j;
    j = 0;
    for (i = 0; s[i] != '\0'; i++)
        if (s[i] < '0'____(1)____s[i] > '9') {
            s[j] = s[i];
            ____ (2) ____;
        }
    return ____ (3) ____;
}
```

```
const int SIZE = 30;
```

```
int main() {
    char s[SIZE];
    int len, i;
    cin.getline(s, sizeof(s));
    len = delnum(s);
    for (i = 0; i < len; i++)
        cout << ____ (4) ____;
    cout << endl;
```

```

    return 0;
}

```

答案

```

(1)||
(2)j++ 或 j=j+1 或 ++j
(3)j
(4)s[i]

```

## 第 28 题 问答题

完善程序：（最大子矩阵和）给出m行n列的整数矩阵，求最大的子矩阵和（子矩阵不能为空）。

输入第一行包含两个整数m和n，即矩阵的行数和列数。之后m行，每行n个整数，描述整个矩阵。程序最终输出最大的子矩阵和。（最后一空 4 分， 其余 3 分， 共 16 分）

```
#include <iostream>
```

```
using namespace std;
```

```
const int SIZE = 100;
```

```
int matrix[SIZE + 1][SIZE + 1];
```

```
int rowsum[SIZE + 1][SIZE + 1]; //rowsum[i][j]记录第 i 行前 j 个数的和
int m, n, i, j, first, last, area, ans;
```

```
int main() {
```

```
    cin >> m >> n;
```

```
    for (i = 1; i <= m; i++)
```

```
        for (j = 1; j <= n; j++)
```

```
            cin >> matrix[i][j];
```

```
    ans = matrix____(1)____;
```

```
    for (i = 1; i <= m; i++)
```

```
        ____ (2) ____;
```

```
        for (i = 1; i <= m; i++)
```

```
            for (j = 1; j <= n; j++)
```

```
                rowsum[i][j] = ____ (3) ____;
```

```
        for (first = 1; first <= n; first++)
```

```
            for (last = first; last <= n; last++) {
```

```
                ____ (4) ____;
```

```
                for (i = 1; i <= m; i++) {
```

```
                    area += ____ (5) ____;
```

```
                    if (area > ans)
```

```
                        ans = area;
```

```
                    if (area < 0)
```

```

        area = 0;
    }
}
cout << ans << endl;
return 0;
}

```

答案

```

(1)[1][1]
(2)rowsum[i][0]=0
(3)rowsum[i][j-1]+matrix[i][j]
(4)area=0
(5)rowsum[i][last]-rowsum[i][first-1]

```