

2011年第十七届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

在二进制下， $1011001 + (\quad) = 1100110$ 。

- A. 1011
- B. 1101
- C. 1010
- D. 1111

答案 B

第2题 单选题

字符“0”的ASCII码为48，则字符“9”的ASCII码为（ ）。

- A. 39
- B. 57
- C. 120
- D. 视具体的计算机而定

答案 B

第3题 单选题

一片容量为8G的SD卡能储存大约（ ）张大小为2MB的数码照片。

- A. 1600
- B. 2000
- C. 4000
- D. 16000

答案 C

第4题 单选题

摩尔定律（ Moore's law）是由英特尔创始人之一戈登·摩尔（Gordon Moor）提出来的。根据摩尔定律，在过去几十年一级在可预测的未来纪念，单块集成电路的集成度大约每（ ）个月翻一番。

- A. 1
- B. 6
- C. 18
- D. 36

答案 C

第 5 题 单选题

无向完全图是图中每对顶点之间都恰好有一条边的简单图。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

答案 B

第 6 题 单选题

寄存器是（ ）的重要组成部分。

- A. 硬盘
- B. 高速缓存
- C. 内存
- D. 中央处理器 (CPU)

答案 D

第 7 题 单选题

如果根结点的深度记为1，则一棵恰有2011个叶结点的二叉树的深度最少是（ ）。

- A. 10
- B. 11
- C. 12
- D. 13

答案 B

第 8 题 单选题

体育课的铃声响了，同学们都陆续地奔向操场，按老师的要求从高到矮站成一排。每个同学按顺序来到操场时，都从排尾走到排头，找到第一个比自己高的同学，并站在他的后面。这种站队的方法类似于（ ）算法。

- A. 快速排序
- B. 插入排序

- C. 冒泡排序
- D. 归并排序

答案 B

第 9 题 单选题

一个正整数在二进制下有100 位，则它在十六进制下有（ ）位。

- A. 7
- B. 13
- C. 25
- D. 不能确定

答案 C

第 10 题 单选题

有人认为，在个人电脑送修前，将文件放入回收站中就是已经将其删除了。这种想法是（ ）。

- A. 正确的，将文件放入回收站以为着彻底删除、无法恢复
- B. 不正确的，只有将回收站清空后，才意味着彻底删除、无法恢复
- C. 不正确的，即使回收站清空，文件只是被标记为删除，仍可能通过恢复软件找回
- D. 不正确的，只要在硬盘上出现过的文件，永远不可能被彻底删除

答案 C

第 11 题 单选题

广度优先搜索时，需要用到的数据结构是（ ）。

- A. 链表
- B. 队列
- C. 栈
- D. 散列表

答案 B

第 12 题 单选题

在使用高级语言编写程序时，一般提到的“空间复杂度”中的“空间”是指（ ）。

- A. 程序运行时理论上所占的内存空间
- B. 程序运行时理论上所占的数组空间
- C. 程序运行时理论上所占的硬盘空间
- D. 程序源文件理论上所占的硬盘空间

答案 A

第 13 题 单选题

在含有 n 个元素的双向链表中查询是否存在关键字为 k 的元素，最快情况下运行的时间复杂度是（ ）。

- A. $O(1)$
- B. $O(\log n)$
- C. $O(n)$
- D. $O(n \log n)$

答案 C

第 14 题 单选题

生物特征识别，是利用人体本身的生物特征进行身份认证的一种技术。目前，指纹识别、虹膜识别、人脸识别等技术已广泛应用于政府、银行、安全防卫等领域。 一下不属于生物特征识别技术及其应用的是（ ）。



指静脉验证



步态验证



ATM机密码验证



声音验证

答案 C

第 15 题 单选题

现有一段文言文，要通过二进制哈夫曼编码进行压缩。简单起见，假设这段文言文只由4个汉字“之”、“呼”、“者”、“也”组成，它们出现的次数分别为700、600、300、200。那么，“也”字的编码长度是（ ）。

- A. 1
- B. 2
- C. 3
- D. 4

答案 C

第 16 题 单选题

关于汇编语言，下列说法错误的是()

- A. 是一种与具体硬件相关的程序设计语言
- B. 在编写复杂程序时，相对于高级语言而言代码量较大，且不易调试
- C. 可以直接访问寄存器、内存单元、以及I/O端口
- D. 随着高级语言的诞生，如今已完全被淘汰，不再使用

答案 D

第 17 题 单选题

() 是一种选优搜索法，按选优条件向前搜索，以达到目标。当搜索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择。

- A. 回溯法
- B. 枚举法
- C. 动态规划
- D. 贪心

答案 A

第 18 题 单选题

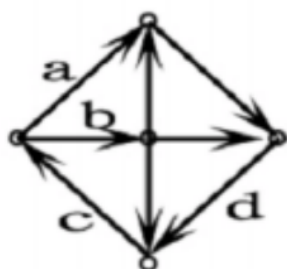
1956 年 () 手语肖克利、巴丁和布拉顿，以表彰他们对半导体的研究和晶体管效应的发现。

- A. 诺贝尔物理学奖
- B. 约翰·冯·诺依曼奖
- C. 图灵奖
- D. 高德纳奖

答案 A

第 19 题 单选题

对一个有向图而言，如果每个节点都存在到达其他任何节点的路径，那么就称它是强连通的。例如，有图就是一个强连通图。事实上，在删掉边()后，它依然是强连通的。



- A. a

- B. b
- C. c
- D. d

答案 A

第 20 题 单选题

从ENIAC到当前最先进的计算机，冯·诺依曼体系结构始终占有重要地位。冯诺依曼提醒结构的核心内容是（ ）。

- A. 采用开关电路
- B. 采用半导体器件
- C. 采用存储程序和程序控制原理
- D. 采用键盘输入

答案 C

二、问题求解

第 21 题 填空题

每份考卷都有一个8位二进制序列号。当且仅当一个序列号含有偶数个1 时，它是有效的。例如，0000000、01010011 都是有效的序列号，而 11111110不是。那么，有效的序列号共有_____个。

答案 128

第 22 题 填空题

定义字符串的基本操作为：删除一个字符\插入一个字符和将一个字符修改成另外一个字符这三种操作。将字符串A变成字符串B的最少操作步数，称为字符串A到字符串B的编辑距离。字符串“ ABCDEFG ”到字符串“ BADECG ”的编辑距离为_____。

答案 3

三、阅读程序写结果

第 23 题 填空题

```
1 | #include<iostream>
2 | using namespace std;
3 | int main()
4 | {
```

```

5 | inti,n,m,ans;
6 | cin>>n>>m;
7 | i=n;
8 | ans=0;
9 | while(i<=m){
10 | ans+=i;
11 | i++;
12 | }
13 | cout<<ans<<endl;
14 | return 0;
15 | }

```

输入： 10 20

输出： _____

艮 答案 165

第 24 题 填空题

```

1 | #include<iostream>
2 | #include<string>
3 | using namespace std;
4 | int main()
5 | {
6 | string map= "2223334445556667778889999";
7 | string tel;
8 | inti;
9 | cin>>tel;
10 | for(i=0;i<tel.length();i++)
11 | if((tel[i]>='0') && (tel[i]<='9') )
12 | cout<<tel[i];
13 | elseif( (tel[i]>='A') && (tel[i]<='Z'))
14 | cout<<map[tel[i]-'A'];
15 | cout<<endl;
16 | return 0;
17 | }

```

输入： CCF-NOIP-2011

输出： _____

艮 答案 22366472011

第 25 题 填空题

```

1  #include<iostream>
2  #include<cstring>
3  using namespace std;
4  const int SIZE = 100;
5  int main()
6  {
7  int n,i,sum,x,a[SIZE];
8  cin>>n;
9  memset(a,0,sizeof(a));
10 for(i=1;i<=n;i++){
11 cin>>x;
12 a[x]++;
13 }
14 i=0;
15 sum=0;
16 while(sum<(n/2+1)){
17 i++;
18 sum+=a[i];
19 }
20 cout<<i<<endl;
21 return 0;
22 }

```

输入:

11

4 5 6 6 4 3 3 2 3 2 1

输出: _____

答案 3

第 26 题 填空题

```

1  #include<iostream>
2  using namespace std;
3  int solve(intn,intm)
4  {
5  inti,sum;
6  if(m==1) return 1;
7  sum=0;
8  for(i=1;i<n;i++)
9  sum+= solve(i,m-1);
10 return sum;
11 }
12 int main()
13 {
14 int n,m;
15 cin>>n>>m;
16 cout<<solve(n,m)<<endl;

```



```
17 | return 0;
18 | }
```

输入： 7 4

输出： _____

答案 20

四、完善程序

第 27 题 问答题

(子矩阵) 给输入一个 $n_1 \times m_1$ 的矩阵a, 和 $n_2 \times m_2$ 的矩阵b, 问a中是否存在子矩阵和b相等。若存在, 输出所有子矩阵左上角的坐标: 若不存在输出"There isno answer"。

```
#include<iostream>
```

```
using namespace std;
```

```
const int SIZE = 50;
```

```
int n1,m1,n2,m2,a[SIZE][SIZE],b[SIZE][SIZE];
```

```
int main()
```

```
{
```

```
    int i,j,k1,k2;
```

```
    bool good ,haveAns;
```

```
    cin>>n1>>m1;
```

```
    for(i=1;i<=n1;i++)
```

```
        for(j=1;j<=m1;j++)
```

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

```
    cin>>n2>>m2;
```

```
    for(i=1;i<=n2;i++)
```

```
        for(j=1;j<=m2;j++)
```

```
            _____ ①
```

```
    haveAns=false;
```

```
    for(i=1;i<=n1-n2+1;i++)
```

```
        for(j=1;j<=_____ ② ;j++){
```

```
            _____ ③ ;
```

```

for(k1=1;k1<=n2;k1++)
    for(k2=1;k2<= _____④;k2++){
        if(a[i+k1-1][j+k2-1]!=b[k1][k2])
            good=false;
    }
if(good){
    cout<<i<<"<j<<endl;
    _____⑤;
}
}
if(!haveAns)
    cout<<"There is no answer"<<endl;
return 0;
}

```

答案

- ① cin>>b[i][j]
- ② m1-m2+1
- ③ good=true
- ④ m2
- ⑤ haveAns=true

第 28 题 问答题

(大整数开方) 输入一个正整数 n ($1 \leq n \leq 10^{100}$)，试用二分法计算它的平方根的整数部分。

```

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

const int SIZE=200;
struct hugeint{
    int len,num[SIZE];
};
// 其中len表示大整数的位数；num[1]表示个位，num[2]表示十位，以此类推

hugeint times(hugeinta,hugeintb)
// 计算大整数a和b的乘积
{
    inti,j;
    hugeint ans;
    memset(ans.num,0,sizeof(ans.num));

```

```

for(i=1;i<=a.len;i++)
    for(j=1;j<=b.len;j++)
        _____①_____+=a.num[i]*b.num[j];
for(i=1;i<=a.len+b.len;i++){
    ans.num[i+1]+=ans.num[i]/10;
    _____②_____;
}
if(ans.num[a.len+b.len]>0)
    ans.len=a.len+b.len;
else
    ans.len=a.len+b.len-1;
return ans;
}

```

```

hugeint add(hugeinta,hugeintb)
// 计算大整数a和b的和
{
    inti;
    hugeint ans;
    memset(ans.num,0,sizeof(ans.num));
    if(a.len>b.len)
        ans.len=a.len;
    else
        ans.len=b.len;
    for(i=1;i<=ans.len;i++){
        ans.num[i]+=_____③_____;
        ans.num[i+1]+= ans.num[i]/10;
        ans.num[i]%=10;
    }
    if(ans.num[ans.len+1]>0)
        ans.len++;
    return ans;
}

```

```

hugeint average(hugeinta,hugeintb)
// 计算大整数a和b的平均数的整数部分
{
    inti;
    hugeint ans;
    ans=add(a,b);

```

```

for(i=ans.len;i>=2;i--){
    ans.num[i-1]+=(_____④_____) *10;

    ans.num[i]/=2;
}
ans.num[1]/=2;
if(ans.num[ans.len]==0)
    ans.len--;
return ans;
}

```

```

hugeint plustwo(hugeinta)
// 计算大整数加2之后的结果
{
    inti;
    hugeint ans;
    ans=a;
    ans.num[1]+=2;
    i=1;
    while( (i<=ans.len)&&(ans.num[i]>=10) ){
        ans.num[i+1]+=ans.num[i]/10;
        ans.num[i]%=10;
        i++;
    }
    if(ans.num[ans.len+1]>0)
        _____⑤_____;
    return ans;
}

```

```

boolover(hugeinta,hugeintb)
// 若大整数a>b则返回true, 否则返回false
{
    inti;
    if(_____⑥_____)

        return false;
    if( a.len>b.len )
        return true;
    for(i=a.len;i>=1;i--){

```

```

        if(a.num[i]<b.num[i])
            return false;
        if(a.num[i]>b.num[i])
            return true;
    }
    return false;
}

int main()
{
    string s;
    inti;
    hugeint target,left,middle,right;
    cin>>s;
    memset(target.num,0,sizeof(target.num));
    target.len=s.length();
    for(i=1;i<=target.len;i++)
        target.num[i]=s[target.len-i]- _____ ⑦
    memset(left.num,0,sizeof(left.num));
    left.len=1;
    left.num[1]=1;
    right=target;
    do{
        middle=average(left,right);
        if(over(_____ ⑧))
            right=middle;
        else
            left=middle;
    }while(!over(plustwo(left),right) );
    for(i=left.len;i>=1;i--)
        cout<<left.num[i];
    return 0;
}

```

答案

- ① ans.num[i+j-1]
- ② ans.num[i]%10
- ③ a.num[i]+b.num[i]
- ④ ans.num[i] % 2
- ⑤ ans.len++
- ⑥ a.len<b.len
- ⑦ '0'或 48

⑧ times(middle,middle),target