

# 考试科目名称：程序设计基础(期末 A 卷)

2020—2021 学年 第二学期 教师 刘奇志 考试方式：闭卷

系（专业）\_\_\_\_\_ 年级\_\_\_\_\_

学号\_\_\_\_\_ 姓名\_\_\_\_\_ 成绩\_\_\_\_\_

题号	一	二	三	四
分数				

得分

一、读程序。分析下列C/C++程序的处理结果，并从编程风格角度对程序的优劣进行点评。（本题满分20分，沾点儿边且言之成理且不重复的说法都给了分）

1. 3+4=7分

```
#include <stdio.h> // #include <iostream>
//using namespace std;
int Soo(int m);
int main()
{
    printf("%d ", Soo(2021));
    //cout << Soo(2021) << ' ';
    printf("%d ", Soo(703));
    //cout << Soo(703) << ' ';
    printf("%d ", Soo(80101));
    //cout << Soo(80101) << ' ';
    return 0;
}
int Soo(int m)
{
    while (m >= 10)
    {
        int sum = 0;
        int t = m;
        while (t)
        {
            sum += t % 10;
            t /= 10;
        }
        m = sum;
    }
    return m;
}
```

该程序执行后输出的结果是：

5 1 1

该程序代码有以下优点：

有缩进

有适当的空格

一行只写一句

有过程抽象（函数）

有声明

变量名小写

函数名大写开头

2. 6分

```
#include <stdio.h> // #include <iostream>
//using namespace std;
int n = 222;
bool Paa();
int main()
{
    if (Paa()) printf("%d ", n);
    //cout << n << ' ';
    return 0;
}
bool Paa()
{
    int m = 0;
    int temp = n;
    while (temp > 0)
    {
        m = m*10 + temp%10;
        temp /= 10;
    }
    return m == n;
}
```

该程序执行后输出的结果是：

222

该程序中函数 Paa 的功能是：

判断 n 是否回文数

该程序代码有以下不足：

没有恰当空行

有全局变量

### 3. 7分 (全排列。许多同学顺序写错了。)

```
#include <stdio.h> // #include <iostream>
//using namespace std;
void Swap(int &a, int &b);
//void Swap(int *a, int *b);
void Perm(int list[], int n, int k);
int main()
{
    int a[] = {1, 2, 3};
    Perm(a, 3, 0);
    return 0;
}
void Perm(int list[], int n, int k)
{
    if (k == n - 1)
    {
        for (int i = 0; i < n; ++i)
            printf("%d ", list[i]);
        //cout << list[i] << " ";
        printf("\n"); //cout << endl;
    }
    else
    {
        for (int i = k; i < n; ++i)
        {
            Swap(list[k], list[i]);
            //Swap(&list[k], &list[i]);
            Perm(list, n, k + 1);
            Swap(list[k], list[i]);
            //Swap(&list[k], &list[i]);
        }
    }
}
void Swap(int &a, int &b)
//void Swap(int *a, int *b)
{
    int temp = a; //int temp = *a;
    a = b; // *a = *b;
    b = temp; // *b = temp;
}
```

该程序执行后输出的结果是：

1 2 3

1 3 2

2 1 3

2 3 1

3 2 1

3 1 2

该程序代码有以下不足：

没有恰当空行

没有符号常量

得分

二、填代码。根据 要实现的功能 和 已经给出的C/C++代码，将缺失的代码补充完整。（本题满分30分）

#### 1. 求[2, 1000000]中的素数的个数。

```
#include <stdio.h> // #include <iostream>
//using namespace std;
bool Prime(int n);
int main()
{
    int cnt (①);
    for (int i = 2; i <= 1000000; ++i)
        if (Prime(i))
            (②);
    printf("%d\n", cnt);
    //cout << cnt << endl;
    return 0;
}
bool Prime(int i)
{
    for (int j=2; (③) <= i; ++j)
        if ((④))
            (⑤);
    return true;
}
```

①

= 0

②

cnt++

③

j\*j 或 j+1

④

i\*j == 0

⑤

return false

2. 计算 m、n 的最大公约数 Gcd 和最小公倍数 Lcm。

```
#include <stdio.h> // #include <iostream>
// using namespace std;
int Gcd(int m, int n);
int main()
{
    int m, n;
    scanf("%d%d", &m, &n);
    // cin >> m >> n;
    printf("Gcd: %d\n", Gcd(m, n));
    // cout << "Gcd: " << Gcd(m, n) << endl;
    printf("Lcm: %d\n", (①));
    // cout << "Lcm: " << (①) << endl;
}
int Gcd(int m, int n)
{
    while ((②))
    {
        int r = n;
        (③);
        m = r;
    }
    return m;
}
```

①  
 $m*n/Gcd(m,n)$   
②  
 $n \neq 0$  或  $n$   
或  $n \% m \neq 0$  或  $n \% m$   
③  
 $n = m \% n$  或  $n = m \% r$

3. 判断奇数阶方阵是否是“幻方”（“幻方”的每行、每列、主对角线及反主对角线上的元素之和都相等）。

```
const int N = 3;
bool Foo(int a[][N], int n)
{
    int i, j, row, column, m1, m2;
    for (i = 0; i < N; ++i)
    {
        (①);
        (②);
        for (j = 0; j < N; ++j)
        {
            row += a[i][j];
            column += a[j][i];
        }
        if ((③) || (④))
            return false;
    }
    m1 = 0;
    m2 = 0;
    for (i = 0; i < N; ++i)
    {
        j = N - i - 1;
        (⑤);
        (⑥);
    }
    if (m1 != m2)
        return false;
    (⑦);
}
```

①  
 $row = 0$   
②  
 $column = 0$   
③  
 $row \neq column$   
④  
 $row \neq m1$   
或  
 $row \neq m2$   
⑤  
 $m1 += a[i][i]$   
⑥  
 $m2 += a[i][j]$   
⑦  
 $return true$

①②顺序可颠倒；③④顺序可颠倒；  
⑤⑥顺序可颠倒。

此题按代码顺序调整前的解法1:

①  $m1=row$  ②  $row=0, column=0$  ③  $row!=column$  ④  $i>0 \& \& m1!=row$  或  $i>0 \& \& m1!=column$   
⑤  $m1+=a[i][i]$  ⑥  $m2+=a[j][i]$  ⑦  $return row==m1$

得分

三、改错误。基于 要实现的功能 和 所给出的C/C++程序，纠正注释符所在行左侧代码中的错误，将正确代码写在注释符的右侧。（本题满分30分）

1. 将主调函数中一个整数的各位偶数数字删去，剩下的数按照从高位到低位的顺序组成一个新数，并返回新数的数量级。例如，将 27638496 处理成 739，返回 100。

```

int Solve(unsigned long int* n)
{
    unsigned long int x = 0;
    int i, t = 1;
    while (n == 0) //①
    {
        i = n % 10; //②
        if (i % 2 != 0)
        {
            x = x + i * t;
            t = t * 10;
        }
        n /= 10; //③
    }
    n = i; //④
    return t/10;
}

```

**\*n 只占 2 分**

① \*n != 0 或 \*n

② i = \*n % 10;

③ \*n /= 10;

④ \*n = x;

2. 下面的程序是要搜索这样的 3 位数：百位、十位和个位数字的阶乘之和恰好等于它本身，即：abc = a! + b! + c!。

```

#include <stdio.h> //include <iostream>
//using namespace std;
bool Moo(int);
int main()
{
    for (int i = 100; i < 1000; ++i)
        if (!Moo(i)) //①
            printf("%d ", i);
    return 0;
}
bool Moo(int i)
{
    int f[] = {1,1,2,6,24,120,720,5040,40320,362880};
    int sum = 0;
    int x = i;
    if (x > 0) //②
    {
        sum += f[x/10]; //③
        x /= 10; //④
    }
    if (i == sum) //⑤
        return true;
    else
        return false;
}

```

① if(Moo(i))

② while(x > 0)

③ sum += f[x%10];

④ x /= 10;

⑤ if(i == sum)

3. 假定一个 OJ 系统要求所有用户的 ID 号码是一组连续的互不重复的正整数。因管理员的疏忽，录入一批新用户名单的时候输错了一个号码，造成这批新号码出现一对 ID 重号和另一个 ID 断号。下面的程序是要在录入新号码时帮助管理员找出这种错误。

```

#include <stdio.h> //include <iostream>
//using namespace std;
const int INF = 0x7fffffff;
int main()
{
    int n;
    scanf("%d", &n); //cin >> n; 充分大，不越界
    int *ha = (int *)malloc(n*sizeof(int));
    //new int [n];
    ha[n] = 0; //①
}

```

① for(int i=0; i < n; ++i)  
ha[i] = 0;

```

int minv = -INF;           //②
int maxv = INF;            //③
                             //④
for(int i=0; i < n; ++i)
{
    scanf("%d", &tp); //cin >> tp;
    if(tp < minv) minv = tp;
    if(tp > maxv) maxv = tp;
    ha[tp]++;
}
int ans1 = 0, ans2 = 0;
for(int i = minv; i <= maxv; ++i)
{
    if(ha[i] == ha[i+1])           //⑤
    {
        ans1 = i;
        cout << ans1 << endl;
    }
    else                           //⑥
    {
        ans2 = i;
        cout << ans2 << endl;
    }
}
return 0;
}

```

```

②
    int minv = INF;
③
    int maxv = -INF;
④
    int tp;
⑤
    if(ha[i] == 2)
或 if(ha[i] > 1)
⑥
    if(ha[i] == 0)

```

②③顺序可颠倒；⑤⑥顺序可颠倒。

得分

#### 四、编程题。根据功能和要求，实现C/C++函数。（本题满分20分）

假定南京大学乒乓球协会有一个保存所有男球员号码的数组 `int boys[]`，和一个保存所有女球员号码的数组 `int girls[]`，现在要用函数 `GenAllCouples` 创建一个单向链表，来保存所有可能的混双组合，每一种组合保存为链表中的一个节点：

```

typedef struct Node
{
    int no;           // 组合序号 (从 1 开始递增)
    int girl;         // 组合中女球员的号码 (不为 0)
    int boy;          // 组合中男球员的号码 (不为 0)
    struct Node *next;
}Node;

```

请你实现 `GenAllCouples` 函数，返回头节点的地址（头节点的序号存入组合总数，头节点的男女球员的号码分别存入 0）。函数原型为：

```
Node *GenAllCouples(int boys[], int n1, int girls[], int n2);
```

其中，`n1` 是男球员的人数，`n2` 是女球员的人数。

```
Node *GenAllCouples(int boys[], int n1, int girls[], int n2)
{ //尾部追加法
```

```
    Node *head = (Node *)malloc(sizeof(Node)); //创建特殊头节点
                                     // Node *head = new Node;

    head -> no = n1 * n2;
    head -> girl = 0;
    head -> boy = 0;
    Node *tail = head;
    int index = 1;
    for(int i = 0; i < n1; ++i)
        for(int j = 0; j < n2; ++j)
        {
            Node *p = (Node *)malloc(sizeof(Node)); //创建新节点
            p -> no = index;      //组合序号
            p -> boy = boys[i];
            p -> girl = girls[j];
            p -> next = NULL;
            tail -> next = p;
            tail = p;
            ++index;
        }
    return head;
}
```

或

```
Node *GenAllCouples(int boys[], int n1, int girls[], int n2)
{
```

```
    Node *head = (Node *)malloc(sizeof(Node)); //创建特殊头节点
    head -> no = n1 * n2;
    head -> girl = 0;
    head -> boy = 0;
    Node *q = head;
    int index = 1;
    Node *head2 = NULL, *p;
    for(int i = 0; i < n1; ++i)
        for(int j = 0; j < n2; ++j)
        {
            p = (Node *)malloc(sizeof(Node)); //创建新节点
            p -> no = index;      //组合序号
            p -> girl = girls[j];
            p -> boy = boys[i];
            q -> next = p;
            q = q -> next;
            ++index;
        }
    q -> next = NULL;
    return head;
}
```

或

```

Node *GenAllCouples(int boys[], int n1, int girls[], int n2)
{ //插入法
    .....
    head -> next = NULL;
    int index = head -> no;
    Node *head2 = NULL, *p;
    .....
        if(head2 == NULL)
            p -> next = NULL;    //将成为最后一个节点
        else
            p -> next = head2;
        head2 = p;
        --index;
    }
    head -> next = head2; //按上特殊头节点
    return head;
}

```