

# 初赛模赛

# 选择题

快快编程  
kkcoding.net

选择 1 数据结构中，“先进后出”是（ ）结构的特征。

A. 队列      B. 栈      C. 线性表      D. 树

答案：B

解析：栈的入栈/出栈顺序是先进后出（FILO），队列的入队/出队顺序是先进先出（FIFO），

选择

2 下面不同进制的数中最小的是（ ）。

A.  $(11101101.01)_2$

B.  $(323.87)_{10}$

C.  $(371.56)_8$

D.  $(2AF.4B)_{16}$

答案：A

解析：将各个数字转换成十进制进行比较，先对整数部分进行比较，A选项整数部分 $(237)_{10}$ ；C选项整数部分 $(249)_{10}$ ；D选项整数部分 $(687)_{10}$ ，故选A

快快编程  
kkcoding.net

选择

3 一个8位整型变量占用（ ）个字节

A. 32    B. 8    C. 4    D. 1

答案：D

解析：计算机中一个字节是8位二进制数，因此一个8位二进制数用1个字节存储。

快快编程  
kkcoding.net

选择

4 三位水仙花数是指每个位上的数字的3次幂之和等于它本身。（例如： $1^3+5^3+3^3=153$ ）找所有三位水仙花数程序如下，它的算法属于（ ）

```
for (i=1; i=9; i++);  
    for(j=0;j=9;j++);  
        for(k=0;k=9;k++);  
            if(i*i*i+j*j*j+k*k*k=i*100+j*10+k) cnt++;
```

A. 动态规划    B.递归    C. 枚举    D. 贪心

答案：C

解析：贪心算法指的是为了达到全局上的最好结果,贪婪地选择局部上的最好结果；递归函数的特点:函数定义中直接或间接地调用了本函数,必定存在可使递归调用终止的条件,否则导致出现无限递归；动态规划算法是通过拆分问题，定义问题状态和状态之间的关系，使得问题能够以递推（或者说分治）的方式去解决；由程序片段可知该方法枚举了所有可能，所以是枚举算法。

选择

5 下列四项中，合法的IP地址是（ ）

A. 190. 220. 5

B. 206. 53. 78

C. 206. 53. 312. 78

D. 123. 43. 82. 220

答案：D

解析：IP地址用32位的二进制数表示，把这32位的二进制数平均分成4组，每组用一个十进制数来表示，组和组之间用.作为分隔符。每个十进制数值范围0~255。

快快编程  
kkcoding.net

选择

6 网址http://www.tinghua.edu.cn中的“cn”代表的是（ ）

- A. 域名    B. 服务器名    C. IP地址    D. 国家

答案：D

解析：cn是指代国家的顶级域名，表示中国。一个完整的域名有主机名，组织机构名和顶级域名组成。

快快编程  
kkcoding.net



选择

7 把7个同样的球放入4个不同的袋子里，不允许有袋子空着不放，问一共有多少种不同的分法（ ）

A. 22    B. 24    C. 18    D. 20

答案：D

解析：插板法，在6个空隙中插入3块板将球分割为4组以对应4个不同的袋子，因此求 $C_6^3$ ，结果为20

选择

8 如果约定二叉树根结点是第一层，那么以下哪项不能成为完全二叉树（ ）

- A. 2 层共 2 个结点      B. 4 层共 9 个结点  
C. 5 层共 11 个结点      D. 6 层共 40 个结点

答案：C

解析：n 层完全二叉树的节点数从  $2^{(n-1)}$  ~  $2^n - 1$

快快编程  
kkcoding.net

选择

9. 12以内的正整数（包含12）互质的数共有（ ）对？(a,b)和(b,a)算同一对。

A. 43     B. 44     C. 45     D. 46

答案：C

解析：在数论，对正整数 $n$ ，欧拉函数是小于或等于 $n$ 的正整数中与 $n$ 互质的数的数目（因此 $\varphi(1)=1$ ）。欧拉函数的通项公式 $\varphi(n)=n*(1-1/p_1)*(1-1/p_2)*(1-1/p_3)*\dots*(1-1/p_k)$ ，其中 $p_i$ 为 $n$ 的质因数。12以内的互质的数有45对。比如12的因数有 $2*2*3$ 其中是质因数的有2, 3,  $\Phi(12)=12*(1-1/2)*(1-1/3)=4$ ，再算 $\varphi(11)=11*(1-1/11)=10, \dots$ ，最后把这些数字都加起来 $\varphi(2)+\dots+\varphi(12)=45$ ”}

选择

10 264和165的最大公约数是（ ）

A. 27    B. 31    C. 29    D. 33

答案：D

解析：对264进行短除法分解为 $2*2*2*3*11$ ；  
对165进行短除法分解为 $3*5*11$ ；  
因此最大公约数为 $3*11=33$

快快编程  
kkcoding.net

选择

11 某班有30个同学报名参加100、400、800m 3个运动项目比赛。已知有6人获 100m参赛资格，8人获400m参赛资格，18人获800m参赛资格。且其中有1人获全部3项参赛资格，则至少有（ ）人没有获任何项目参赛资格。

A. 0    B. 1    C. 2    D. 3

答案：A

解析：有1人获全部3项参赛资格,所以最大得参数资格人数为  $(6-1) + (8-1) + (18-1) + 1 = 30$ ，所以至少0人

选择

12 假设今天是星期六，再过 $10^{2020}$ 天后的那一天是星期（ ）

A. 一    B. 二    C. 三    D. 四

答案：C

解析：  $1 \div 7 = 0.142857142857..$

所以是6位一个循环

$2020 \div 6 = 336 \text{ 余 } 4$

所以10的2020次方除7的余数与 $10^4$ 的余数相同

$10000 \div 7 = 1428 \text{ 余 } 4$ ，4天后是星期三。

选择

13下面有关计算机的特点叙述，不正确的是( )。

- A. 运算速度快
- B. 有记忆和逻辑判断能力
- C. 具有自动执行程序的能力
- D. 至今没有任何人能给出如何求解方法的难题，计算机也都能求出解来

答案：D

解析：常识题

快快编程  
kkcoding.net

选择

14 假设一棵二叉树的中序遍历序列为 **DGBAIECHF**，后序遍历序列为 **GDBEIHFC**A，则其前序遍历序列为( )。

A. ABCDFGHE

B. ABDGCEFH

C. ACBGDHEF

D. ACEFHBGD

答案： B

解析：后序：左-右-根。中序：左-根-右。先后序中末位结点即为根（在本题中为**A**），然后通过中序里根**A**的所在位置，划分左右子树（根分左右）；再根据左右子树的结点个数，去划分得到后序的左右子树。递归进行上述步骤。可画出树的形态，最后再求先序遍历。



选择

15 Facebook 是美国的一个社交网络服务网站，它的创始人是( )。

- A. 比尔盖茨
- B. 史蒂夫乔布斯
- C. 马克扎克伯格
- D. 马特达蒙

答案：C

解析：常识题。

快快编程  
kkcoding.net

# 阅读程序

快快编程  
kkcoding.net

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

1

识别变量

常见变量名  
翻译循环变量  
根据变量名的英文推断

2

找出关键语句

控制结构(for, if)  
常见算法的基本操作  
函数参数、返回值

3

理解代码段作用

翻译解释代码段

# 阅读程序

## 解释变量的作用

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

x	输入变量1
y	输入变量2
i	循环变量（公因子）
p	最大公因数

本程序输入二个整数的整数x和整数y，求出两者最大公因数p，之和将p和x/p和y/p三数之积输出（本质是两数所有不同质因子之积）

# 阅读程序

## 关键字句

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

循环查找a和b的公因子

- 1.将公因子累乘到p，求出最大公因数
- 2.将x除以公因子，最终得到x/p（x剩余质因子之积）
- 3.将y除以公因子，最终得到y/p（y剩余质因子之积）

求出三数乘积

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

判断

1. 输入两个数字x必须大于等于y，否则则程序无输出。( )

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

判断

2. 若将第6行的“i=2”改为“i=1”，程序输出变为原来的1/2。( )

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

判断

3. 若将第8行的“x%i==0&&y%i==0”改为“!(x%i&&y%i)”，程序运行结果不会改变。( )



# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

判断

4. 若输入的数据互质，那么输出的数字为两者之积。( )

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

选择

5. 若输入的数据为88 32，那么输出为（ ）。

A. 8    B. 352    C. 4    D. 32

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int x,y;
5      cin>>x>>y;
6      int i=2,p=1;
7      do{
8          while(x%i==0&&y%i==0){
9              p*=i;
10             x/=i;
11             y/=i;
12         }
13         i++;
14     }while(x>=i&&y>=i);
15     cout<<p*x*y<<endl;
16     return 0;
17 }
```

选择 6. 若输出结果是280，则（ ）不可能是输入数据

A. 140 2

B. 56 35

C. 40 7

D. 140 56

# 阅读程序

快快编程  
kkcoding.net

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

1

识别变量

常见变量名  
翻译循环变量  
根据变量名的英文推断

2

找出关键字句

控制结构(for, if)  
常见算法的基本操作  
函数参数、返回值

3

理解代码段作用

翻译解释代码段

# 阅读程序

## 解释变量的作用

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

n

修改范围最大值

m

数字串最大长度

p

游标，指向待修改值

i

循环变量

a[i]

数字串每一位

# 阅读程序

## 关键词句

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

p号元素自增

超过n则回到0，p后退一位

p前进一位，拷贝前一项数值

if分支判段：超过m则输出前m项数字串，p后退一位

注意关键条件

遇到0结束while

# 简化：数组模拟

i	p	2	3	4
a[i]	1	0	0	0

```
8  
9  
10  
11  
12  
13  
14  
15  
16
```

```
a[p]++;  
if(a[p]>n){  
    a[p]=0;  
    p--;  
}  
else{  
    p++;  
    a[p]=a[p-1];  
}
```

```
17  
18  
19  
20  
21  
22
```

```
if(p>m){  
    for(int i=1;i<=m;i++) cout<<a[i];  
    cout<<" ";  
    p--;  
}  
}while(p>=1);
```



# 简化：数组模拟

i	1	p	3	4
a[i]	1	2	0	0

```
8  
9  
10  
11  
12  
13  
14  
15  
16
```

```
a[p]++;  
if(a[p]>n){  
    a[p]=0;  
    p--;  
}  
else{  
    p++;  
    a[p]=a[p-1];  
}
```

```
17  
18  
19  
20  
21  
22
```

```
if(p>m){  
    for(int i=1;i<=m;i++) cout<<a[i];  
    cout<<" ";  
    p--;  
}  
}while(p>=1);
```

# 简化：数组模拟

i	1	2	p	4
a[i]	1	2	3	0

```
8  
9  
10  
11  
12  
13  
14  
15  
16
```

```
a[p]++;  
if(a[p]>n){  
    a[p]=0;  
    p--;  
}  
else{  
    p++;  
    a[p]=a[p-1];  
}
```

```
17  
18  
19  
20  
21  
22
```

```
if(p>m){  
    for(int i=1;i<=m;i++) cout<<a[i];  
    cout<<" ";  
    p--;  
}  
}while(p>=1);
```

# 简化：数组模拟

i	1	2	3	p
a[i]	1	2	3	3

输出1 2 3

```
8
9
10
11
12
13
14
15
16
```

```
a[p]++;
if(a[p]>n){
    a[p]=0;
    p--;
}
else{
    p++;
    a[p]=a[p-1];
}
```

```
17
18
19
20
21
22
```

```
if(p>m){
    for(int i=1;i<=m;i++) cout<<a[i];
    cout<<" ";
    p--;
}
}while(p>=1);
```

# 简化：数组模拟

i	1	2	p	4
a[i]	1	2	4	3

```
8  
9  
10  
11  
12  
13  
14  
15  
16
```

```
a[p]++;  
if(a[p]>n){  
    a[p]=0;  
    p--;  
}  
else{  
    p++;  
    a[p]=a[p-1];  
}
```

```
17  
18  
19  
20  
21  
22
```

```
if(p>m){  
    for(int i=1;i<=m;i++) cout<<a[i];  
    cout<<" ";  
    p--;  
}  
}while(p>=1);
```

# 简化：数组模拟

i	1	2	3	p
a[i]	1	2	4	4

输出1 2 4

```
8  
9  
10  
11  
12  
13  
14  
15  
16
```

```
a[p]++;  
if(a[p]>n){  
    a[p]=0;  
    p--;  
}  
else{  
    p++;  
    a[p]=a[p-1];  
}
```

```
17  
18  
19  
20  
21  
22
```

```
if(p>m){  
    for(int i=1;i<=m;i++) cout<<a[i];  
    cout<<" ";  
    p--;  
}  
}while(p>=1);
```

# 简化：数组模拟

i	1	2	p	4
a[i]	1	2	5	4

```
8  
9  
10  
11  
12  
13  
14  
15  
16
```

```
a[p]++;  
if(a[p]>n){  
    a[p]=0;  
    p--;  
}  
else{  
    p++;  
    a[p]=a[p-1];  
}
```

```
17  
18  
19  
20  
21  
22
```

```
if(p>m){  
    for(int i=1;i<=m;i++) cout<<a[i];  
    cout<<" ";  
    p--;  
}  
}while(p>=1);
```

# 简化：数组模拟

i	1	p	3	4
a[i]	1	3	0	4

```
8  
9  
10  
11  
12  
13  
14  
15  
16
```

```
a[p]++;  
if(a[p]>n){  
    a[p]=0;  
    p--;  
}  
else{  
    p++;  
    a[p]=a[p-1];  
}
```

```
17  
18  
19  
20  
21  
22
```

```
if(p>m){  
    for(int i=1;i<=m;i++) cout<<a[i];  
    cout<<" ";  
    p--;  
}  
}while(p>=1);
```

# 简化：数组模拟

i	1	2	p	4
a[i]	1	3	3	4

```
8  
9  
10  
11  
12  
13  
14  
15  
16
```

```
a[p]++;  
if(a[p]>n){  
    a[p]=0;  
    p--;  
}  
else{  
    p++;  
    a[p]=a[p-1];  
}
```

```
17  
18  
19  
20  
21  
22
```

```
if(p>m){  
    for(int i=1;i<=m;i++) cout<<a[i];  
    cout<<" ";  
    p--;  
}  
}while(p>=1);
```



# 简化：数组模拟

i	1	2	3	p
a[i]	1	3	4	4

输出???

```
8
9
10
11
12
13
14
15
16
```

```
a[p]++;
if(a[p]>n){
    a[p]=0;
    p--;
}
else{
    p++;
    a[p]=a[p-1];
}
```

```
17
18
19
20
21
22
```

```
if(p>m){
    for(int i=1;i<=m;i++) cout<<a[i];
    cout<<" ";
    p--;
}
}while(p>=1);
```

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

判断

1. 输入两个数字n必须大于等于m，否则则程序无输出。( )

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

判断

2. 若将第3行和第4行置换，程序输出结果保持不变。( )

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

判断

3. 若输入的数据为9 5，则变量p在程序运行中不超过5。( )

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

判断 4. 若输入的数据为9 5，则输出所有数字串长度均不超过5。( )

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

选择 若输入的数据为5 3，则输出数字串的数量为( )

A. 5

B. 8

C. 10

D. 12

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int a[101]={};
4  int main(){
5      int n,m; cin>>n>>m;
6      int p=1;
7      do{
8          a[p]++;
9          if(a[p]>n){
10             a[p]=0;
11             p--;
12         }
13         else{
14             p++;
15             a[p]=a[p-1];
16         }
17         if(p>m){
18             for(int i=1;i<=m;i++) cout<<a[i];
19             cout<<" ";
20             p--;
21         }
22     }while(p>=1);
23     return 0;
24 }
```

选择

6.若输入的数据为4 2，那么输出为（ ）。

A. 12 13 14 23 24 34

B. 12 13 14

C. 23 24 34

D. 123 124 134 234

# 阅读程序

快快编程  
kkcoding.net



3分钟

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int b[6];
4  int main(){
5      for(int i=1;i<=5;i++) b[i]=i;
6      int s=1;
7      int j=1;
8      while(j>0){
9          j=5;
10         while(j>0&& b[j]==10+j-5) j--;
11         if(j>0){
12             s++;b[j]++;
13             for(int i=j+1;i<=5;i++) b[i]=b[j]+i-j;
14         }
15     }
16     cout<<s<<endl;
17     return 0;
18 }
```

1

识别变量

常见变量名  
翻译循环变量  
根据变量名的英文推断

2

找出关键语句

控制结构(for, if)  
常见算法的基本操作  
函数参数、返回值

3

理解代码段作用

翻译解释代码段

# 简化：数组模拟

j	1	2	3	4	5
b[j]	1	2	3	4	5

```
8 while(j>0){
9     j=5;
10    while(j>0&& b[j]==10+j-5) j--;
11    if(j>0){
12        s++; b[j]++;
13        for(int i=j+1; i<=5; i++) b[i]=b[j]+i-j;
14    }
15 }
```

# 简化：数组模拟

j	1	2	3	4	5
b[j]	1	2	3	4	6

```
8 while(j>0){
9     j=5;
10    while(j>0&& b[j]==10+j-5) j--;
11    if(j>0){
12        s++; b[j]++;
13        for(int i=j+1; i<=5; i++) b[i]=b[j]+i-j;
14    }
15 }
```

# 简化：数组模拟

j	1	2	3	4	5
b[j]	1	2	3	4	10

```
8 while(j>0){
9     j=5;
10    while(j>0&& b[j]==10+j-5) j--;
11    if(j>0){
12        s++; b[j]++;
13        for(int i=j+1; i<=5; i++) b[i]=b[j]+i-j;
14    }
15 }
```

# 简化：数组模拟

j	1	2	3	4	5
b[j]	1	2	3	5	6

```
8 while(j>0){
9     j=5;
10    while(j>0&& b[j]==10+j-5) j--;
11    if(j>0){
12        s++; b[j]++;
13        for(int i=j+1; i<=5; i++) b[i]=b[j]+i-j;
14    }
15 }
```

# 简化：数组模拟

j	1	2	3	4	5
b[j]	1	2	3	5	10

```
8 while(j>0){
9     j=5;
10    while(j>0&& b[j]==10+j-5) j--;
11    if(j>0){
12        s++; b[j]++;
13        for(int i=j+1; i<=5; i++) b[i]=b[j]+i-j;
14    }
15 }
```

# 简化：数组模拟

j	1	2	3	4	5
b[j]	6	7	8	9	10

```
8 while(j>0){
9     j=5;
10    while(j>0&& b[j]==10+j-5) j--;
11    if(j>0){
12        s++; b[j]++;
13        for(int i=j+1; i<=5; i++) b[i]=b[j]+i-j;
14    }
15 }
```

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int b[6];
4  int main(){
5      for(int i=1;i<=5;i++) b[i]=i;
6      int s=1;
7      int j=1;
8      while(j>0){
9          j=5;
10         while(j>0&& b[j]==10+j-5) j--;
11         if(j>0){
12             s++;b[j]++;
13             for(int i=j+1;i<=5;i++) b[i]=b[j]+i-j;
14         }
15     }
16     cout<<s<<endl;
17     return 0;
18 }
```

## 判断

1.若将第13行的” int i=j+1;i<=5;i++”改为” int i=j;i<=5;i++”, 程序运行结果不会改变。( )



# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int b[6];
4  int main(){
5      for(int i=1;i<=5;i++) b[i]=i;
6      int s=1;
7      int j=1;
8      while(j>0){
9          j=5;
10         while(j>0&& b[j]==10+j-5) j--;
11         if(j>0){
12             s++;b[j]++;
13             for(int i=j+1;i<=5;i++) b[i]=b[j]+i-j;
14         }
15     }
16     cout<<s<<endl;
17     return 0;
18 }
```

判断

2.程序执行过程中，数组b一直保持从小到大排列（相邻可以相等）。（ ）

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int b[6];
4  int main(){
5      for(int i=1;i<=5;i++) b[i]=i;
6      int s=1;
7      int j=1;
8      while(j>0){
9          j=5;
10         while(j>0&& b[j]==10+j-5) j--;
11         if(j>0){
12             s++;b[j]++;
13             for(int i=j+1;i<=5;i++) b[i]=b[j]+i-j;
14         }
15     }
16     cout<<s<<endl;
17     return 0;
18 }
```

选择

3.程序执行完毕时，下面哪个说法是正确的  
( )。

A. b[1]=2    B. b[2]=10    C. b[3]=8    D. b[5]=5

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int b[6];
4  int main(){
5      for(int i=1;i<=5;i++) b[i]=i;
6      int s=1;
7      int j=1;
8      while(j>0){
9          j=5;
10         while(j>0&& b[j]==10+j-5) j--;
11         if(j>0){
12             s++;b[j]++;
13             for(int i=j+1;i<=5;i++) b[i]=b[j]+i-j;
14         }
15     }
16     cout<<s<<endl;
17     return 0;
18 }
```

选择

4.若将第12行的“s++”改为“s+=2”，程序运行结果变为原来的（ ）倍

A. 1

B. 2

C. 10

D. 4

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int b[6];
4  int main(){
5      for(int i=1;i<=5;i++) b[i]=i;
6      int s=1;
7      int j=1;
8      while(j>0){
9          j=5;
10         while(j>0&& b[j]==10+j-5) j--;
11         if(j>0){
12             s++;b[j]++;
13             for(int i=j+1;i<=5;i++) b[i]=b[j]+i-j;
14         }
15     }
16     cout<<s<<endl;
17     return 0;
18 }
```

选择

5.若将第5行的” b[i]=i;” 改为 “ (    ) ” 程序运行结果最小

A. b[i]=1;            B. b[i]=i-1;

C. b[i]=i+1;        D. b[i]=i\*2;

# 阅读程序

```
1  #include<iostream>
2  using namespace std;
3  int b[6];
4  int main(){
5      for(int i=1;i<=5;i++) b[i]=i;
6      int s=1;
7      int j=1;
8      while(j>0){
9          j=5;
10         while(j>0&& b[j]==10+j-5) j--;
11         if(j>0){
12             s++;b[j]++;
13             for(int i=j+1;i<=5;i++) b[i]=b[j]+i-j;
14         }
15     }
16     cout<<s<<endl;
17     return 0;
18 }
```

选择

6. (4分) 输出的值为 ( )

A. 210

B. 252

C. 126

D. 462

# 完善程序

快快编程  
kkcoding.net

# 完善程序

1.（金蝉素数）某古寺的一块石碑上依稀刻有一些三位或四位的神秘自然数。专家研究发现：这些数字是素数，且从低位去掉一位，或两位，.....后都还是素数，从高位去掉一位，或两位，.....后也都仍为素数，更奇妙的是，同时去掉它的最高位与最低位数字后还是素数。因此，人们把这些神秘的素数成为金蝉素数，喻意金蝉脱壳之后仍然为美丽的金蝉。试求出石碑上的金蝉素数。

```

1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4 int main(){
5     a[1]=2,a[2]=3,a[3]=5,a[4]=7;
6     int u=4;
7     for(int i=11;i<10000;i+=2){
8         int x=3;
9         while(x*x<=i&&____(1)____) x++;
10        if(x*x<=i)continue;
11        if(a[u]<1000)a[++u]=i//i是素数
12        if(i<=100)continue;//三位及以上才行
13        int L=log(i)/log(10)+1;//i的位数
14        int t=1,s=0;
15        for(int j=1;j<L;j++){
16            t*=10;
17            int hi=____(2)____;//高j位
18            int lo=____(3)____;//低L-j位
19            int v=1,mid=10000;
20            if(j==L-1)mid=lo/10;//去首去尾的中间部分
21            while(a[v]<=hi||a[v]<=lo){
22                if(a[v]==hi) s++;
23                if(a[v]==lo) s++;
24                if(a[v]==mid) s++;
25                ____ (4) ____;
26            }
27        }
28        if(____(5)____) cout<<i<<endl;
29    }
30    return 0;
31 }

```

1

识别变量

常见变量名  
翻译循环变量  
根据变量名的英文推断

2

找出关键语句

控制结构(for, if)  
常见算法的基本操作  
函数参数、返回值

3

理解代码段作用

翻译解释代码段



# 完善程序

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4  int main(){
5      a[1]=2,a[2]=3,a[3]=5,a[4]=7;
6      int u=4;
7      for(int i=11;i<10000;i+=2){
8          int x=3;
9          while(x*x<=i&&____(1)____) x++;
10         if(x*x<=i)continue;
11         if(a[u]<1000)a[++u]=i//i是素数
12         if(i<=100)continue;//三位及以上才行
13         int L=log(i)/log(10)+1;//i的位数
14         int t=1,s=0;
15         for(int j=1;j<L;j++){
16             t*=10;
17             int hi=____(2)____;//高j位
18             int lo=____(3)____;//低L-j位
19             int v=1,mid=10000;
20             if(j==L-1)mid=lo/10;//去首去尾的中间部分
21             while(a[v]<=hi||a[v]<=lo){
22                 if(a[v]==hi) s++;
23                 if(a[v]==lo) s++;
24                 if(a[v]==mid) s++;
25                 ____ (4) ____;
26             }
27         }
28         if(____(5)____) cout<<i<<endl;
29     }
30     return 0;
31 }

```

1: 寻找素数

2: 判断是否超过三位, 计算该素数的位数

3: 循环提取前j位和后j位数并进行删除

4: 枚举判断前j位/后j位/删除后的中间位是否为素数, 统计个数

4: 根据该数的变形后素数个数判断是否为金禅素数

## 完善程序

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4  int main(){
5      a[1]=2,a[2]=3,a[3]=5,a[4]=7;
6      int u=4;
7      for(int i=11;i<10000;i+=2){
8          int x=3;
9          while(x*x<=i&&____(1)____) x++;
10         if(x*x<=i)continue;
11         if(a[u]<1000)a[++u]=i//i是素数
12         if(i<=100)continue;//三位及以上才行
13         int L=log(i)/log(10)+1;//i的位数
14         int t=1,s=0;
15         for(int j=1;j<L;j++){
16             t*=10;
17             int hi=____(2)____;//高j位
18             int lo=____(3)____;//低L-j位
19             int v=1,mid=10000;
20             if(j==L-1)mid=lo/10;//去首去尾的中间部分
21             while(a[v]<=hi||a[v]<=lo){
22                 if(a[v]==hi) s++;
23                 if(a[v]==lo) s++;
24                 if(a[v]==mid) s++;
25                 ____ (4) ____;
26             }
27         }
28         if(____(5)____) cout<<i<<endl;
29     }
30     return 0;
31 }

```

判断i是否为素数，除不尽则继续加x

i是素数，则加入a数组

2: 判断是否超过三位，计算该素数的位数

3: 循环提取前j位和后j位数并进  
位数操作，取出（用除 $10^k$ 删除k位；  
对 $10^k$ 取余提取k位）

循环v，依次比对，判断是否为已记录的素数

金禅素数保证每次变形都是素数，  
共 $2*L$ 次

# 完善程序

```
1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4  int main(){
5      a[1]=2,a[2]=3,a[3]=5,a[4]=7;
6      int u=4;
7      for(int i=11;i<10000;i+=2){
8          int x=3;      i%x>0
9          while(x*x<=i&&____(1)____) x++;
10         if(x*x==i)continue;
11         if(a[u]<1000)a[++u]=i//i是素数
12         if(i<=100)continue;//三位及以上才行
13         int L=log(i)/log(10)+1;//i的位数
14         int t=1,s=0;
15         for(int j=1;j<L;j++){
16             t*=10;
17             int hi=____(2)____; //高位      i/t
18             int lo=____(3)____; //低位L-j位  i%t
19             int v=1,mid=10000;
20             if(j==L-1)mid=lo/10;//去首去尾的中间部分
21             while(a[v]<=hi||a[v]<=lo){
22                 if(a[v]==hi) s++;
23                 if(a[v]==lo) s++;
24                 if(a[v]==mid) s++;
25                 ____ (4) ____;      v++
26             }
27         }
28         if(____(5)____) cout<<i<<endl;
29     }
30     return 0;      S==2*L-1
31 }
```

# 完善程序

快快编程  
kkcoding.net

# 完善程序

（修理栅栏）农夫约翰为了修理栅栏，要将一块很长的木板切成N块。准备切成的木板长度为L1、L2、.....LN，未切割前木板的长度恰好为切割后木板长度的总和。每次切断木板时，需要的开销为这块木板的长度。例如长度为21的木板要切成长度为5、8、8的三块木板。长21的木板切成长度为13和8的板时，开销为21.再将长度为13的板切成长度为5和8的板时，开销是13。于是合计开销是34。请求出按照目标要求将木板切割完最小的开销是多少？

# 手算样例

输入样例：

3

8 5 8

输出多少？

输出样例：

34

“分割”问题可以等效转换  
为“合并”问题

```

1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  const int N=10009;
5  int n,L[N];
6  void solve(){
7      ll ans=0;
8      while(n>1){
9          int mii1=0,mii2=1;
10         if(L[mii1]>L[mii2]) swap(mii1,mii2);
11         for(int i=2;i<n;i++){
12             if(L[i]<L[mii1]){
13                 mii2=mii1;
14                 ____ (1) ____;
15             }
16             else if(____ (2) ____ ) mii2=i;
17         }
18
19         int t=____ (3) ____;
20         ans+=t;
21         if(mii1==n-1) swap(mii1,mii2);
22         L[mii1]=____ (4) ____;
23         L[mii2]=L[n-1];
24         ____ (5) ____;
25     }
26     cout<<ans<<endl;
27 }
28 int main(){
29     cin>>n;
30     for(int i=0;i<n;i++)
31         cin>>L[i];
32     solve();
33     return 0;
34 }

```

1

识别变量

常见变量名  
翻译循环变量  
根据变量名的英文推断

2

找出关键语句

控制结构(for, if)  
常见算法的基本操作  
函数参数、返回值

3

理解代码段作用

翻译解释代码段

## 解释变量作用

```
1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  const int N=10009;
5  int n,L[N];
6  void solve(){
7      ll ans=0;
8      while(n>1){
9          int mii1=0,mii2=1;
10         if(L[mii1]>L[mii2]) swap(mii1,mii2);
11         for(int i=2;i<n;i++){
12             if(L[i]<L[mii1]){
13                 mii2=mii1;
14                 ____ (1) ____;
15             }
16             else if(____ (2) ____ ) mii2=i;
17         }
18
19         int t=____ (3) ____;
20         ans+=t;
21         if(mii1==n-1) swap(mii1,mii2);
22         L[mii1]=____ (4) ____;
23         L[mii2]=L[n-1];
24         ____ (5) ____;
25     }
26     cout<<ans<<endl;
27 }
28 int main(){
29     cin>>n;
30     for(int i=0;i<n;i++)
31         cin>>L[i];
32     solve();
33     return 0;
34 }
```

ans

当前最小花销

n

木板总数

L[i]

i号木板的长度

t

合并完新增木板长度

mii1

最短木板编号

mii2

次短木板编号



## 完善程序

```

1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  const int N=10009;
5  int n,L[N];
6  void solve(){
7      ll ans=0;
8      while(n>1){
9          int mii1=0,mii2=1;
10         if(L[mii1]>L[mii2]) swap(mii1,mii2);
11         for(int i=2;i<n;i++){
12             if(L[i]<L[mii1]){
13                 mii2=mii1;
14                 ____ (1) ____;
15             }
16             else if(____ (2) ____ ) mii2=i;
17         }
18
19         int t=____ (3) ____;
20         ans+=t;
21         if(mii1==n-1) swap(mii1,mii2);
22         L[mii1]=____ (4) ____;
23         L[mii2]=L[n-1];
24         ____ (5) ____;
25     }
26     cout<<ans<<endl;
27 }
28 int main(){
29     cin>>n;
30     for(int i=0;i<n;i++)
31         cin>>L[i];
32     solve();
33     return 0;
34 }

```

1: 打擂台法每次寻找最短的两根木板

2: 贪心法每次寻找最短的两根木板进行合并

3: 输入各木板长度,调用solve使用进行合并

# 完善程序

```

1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  const int N=10009;
5  int n,L[N];
6  void solve(){
7      ll ans=0;
8      while(n>1){
9          int mii1=0,mii2=1;
10         if(L[mii1]>L[mii2]) swap(mii1,mii2);
11         for(int i=2;i<n;i++){
12             if(L[i]<L[mii1]){
13                 mii2=mii1;
14                 ____ (1) ____;
15             }
16             else if(____ (2) ____ ) mii2=i;
17         }
18
19         int t=____ (3) ____;
20         ans+=t;
21         if(mii1==n-1) swap(mii1,mii2);
22         L[mii1]=____ (4) ____;
23         L[mii2]=L[n-1];
24         ____ (5) ____;
25     }
26     cout<<ans<<endl;
27 }
28 int main(){
29     cin>>n;
30     for(int i=0;i<n;i++)
31         cin>>L[i];
32     solve();
33     return 0;
34 }

```

1: 打擂台法每次寻找最短的两根木板

依次查看每块木板打擂台比较:  
 1.和最短板进行比较,更新最短板和次短板  
 2.和次短板比较,更新次短板

计算最短板和次短板合并后产生的新长度

将新长度保存,总数减少

3: 输入各木板长度,调用solve使用进行合并

```

1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  const int N=10009;
5  int n,L[N];
6  void solve(){
7      ll ans=0;
8      while(n>1){
9          int mii1=0,mii2=1;
10         if(L[mii1]>L[mii2]) swap(mii1,mii2);
11         for(int i=2;i<n;i++){
12             if(L[i]<L[mii1]){
13                 mii2=mii1;
14                 ____ (1) ____; mii1=i
15             }
16             else if(____ (2) ____ ) mii2=i; L[i]<L[mii2]
17         }
18
19         int t=____ (3) ____; L[mii1]+L[mii2]
20         ans+=t;
21         if(mii1==n-1) swap(mii1,mii2);
22         L[mii1]=____ (4) ____; t
23         L[mii2]=L[n-1];
24         ____ (5) ____; n--
25     }
26     cout<<ans<<endl;
27 }
28 int main(){
29     cin>>n;
30     for(int i=0;i<n;i++)
31         cin>>L[i];
32     solve();
33     return 0;
34 }

```