

# 信奥 算法

3分钟

# 阅读程序

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

# 阅读程序

## 解释变量的作用

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

str	输入字符串
count	计数器
i	循环变量



# 阅读程序

关键词句

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

遍历字符串str

判断str[i]是小写字符，  
count++

判断str[i]是大写字符，  
countn++

# 阅读程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

判断 程序删除第8行，运行结果不变。( )

# 阅读程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

判断 将11、12行代码与13、14行代码交换，运行结果不变。( )

# 阅读程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

判断 无论输出的字符串长度是多少，结果都小于str.size() ( )

# 阅读程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

判断 将11行、14行删除，程序运行结果不变( )



# 阅读程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

选择

输入的字符串长度为n，由大写和小写字母构成，程序输出结果( )。

A. 2n    B. n+1    C. n-1    D. n

# 阅读程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

选择

输入的字符串长度为n，只包含大写字母，程序输出结果（ ）

A.  $n/2$

B.  $2n$

C.  $n$

D. 0

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  void fun(char *a, char *b) {
4      a = b;
5      (*a)++;
6  }
7  int main() {
8      char c1, c2, *p1, *p2;
9      c1 = 'A';
10     c2 = 'a';
11     p1 = &c1;
12     p2 = &c2;
13     fun(p1, p2);
14     cout << c1 << c2 << endl;
15     return 0;
16 }
```

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  void fun(char *a, char *b) {
4      a = b;
5      (*a)++;
6  }
7  int main() {
8      char c1, c2, *p1, *p2;
9      c1 = 'A';
10     c2 = 'a';
11     p1 = &c1;
12     p2 = &c2;
13     fun(p1, p2);
14     cout << c1 << c2 << endl;
15     return 0;
16 }
```

a

字符指针

b

字符指针

c1

字符变量

c2

字符变量

\*p1

字符指针,指向c1

\*p2

字符指针,指向c2



# 阅读程序

## 关键语句

```
1  #include <iostream>
2  using namespace std;
3  void fun(char *a, char *b) {
4      a = b;
5      (*a)++;
6  }
7  int main() {
8      char c1, c2, *p1, *p2;
9      c1 = 'A';
10     c2 = 'a';
11     p1 = &c1;
12     p2 = &c2;
13     fun(p1, p2);
14     cout << c1 << c2 << endl;
15     return 0;
16 }
```

a指向b所指向的变量

a指向的变量数值+1

p1指向c1, p2指向c2

调用函数, 将p1和p2作为参数, 可以通过指针修改变量c1和c2的数值

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  void fun(char *a, char *b) {
4      a = b;
5      (*a)++;
6  }
7  int main() {
8      char c1, c2, *p1, *p2;
9      c1 = 'A';
10     c2 = 'a';
11     p1 = &c1;
12     p2 = &c2;
13     fun(p1, p2);
14     cout << c1 << c2 << endl;
15     return 0;
16 }
```

判断 将第9行c1='A'改为c1="A", 程序运行结果不变。( )

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  void fun(char *a, char *b) {
4      a = b;
5      (*a)++;
6  }
7  int main() {
8      char c1, c2, *p1, *p2;
9      c1 = 'A';
10     c2 = 'a';
11     p1 = &c1;
12     p2 = &c2;
13     fun(p1, p2);
14     cout << c1 << c2 << endl;
15     return 0;
16 }
```

判断 11行也可以写成\*p1=c1。( )

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  void fun(char *a, char *b) {
4      a = b;
5      (*a)++;
6  }
7  int main() {
8      char c1, c2, *p1, *p2;
9      c1 = 'A';
10     c2 = 'a';
11     p1 = &c1;
12     p2 = &c1;
13     fun(p1, p2);
14     cout << c1 << c2 << endl;
15     return 0;
16 }
```

判断

第12行p2=&c1是允许的，因为p1和p2可以指向同一个变量( )



# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  void fun(char *a, char *b) {
4      a = b;
5      (*a)++;
6  }
7  int main() {
8      char c1, c2, *p1, *p2;
9      c1 = 'A';
10     c2 = 'a';
11     p1 = &c1;
12     p2 = &c2;
13     fun(p1, p2);
14     cout << c1 << c2 << endl;
15     return 0;
16 }
```

判断 在3-5行函数中，变量c1和c2的数值都会改变( )

# 阅读程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

选择 程序输出结果( )。

A. Aa    B. Ab    C. Ba    D. ab

# 阅读程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      string str;
6      int i;
7      int count;
8      count = 0;
9      getline(cin, str);
10     for (i = 0; i < str.length(); i++) {
11         if(str[i] >= 'a' && str[i] <= 'z')
12             count++;
13         if(str[i] >= 'A' && str[i] <= 'Z')
14             count++;
15     }
16     cout << count << endl;
17     return 0;
18 }
```

选择 想要让程序输出AC，需要将c1或c2的值修改为（ ）

A. c2='A'

B. c2='B'

C. c2='C'

D. c1='C'

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

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 n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

n	输入，边界值
m	输入，边界值
x	数值从1到n再到1，循环
y	数值从1到m再到1，循环
dx	取值+1或-1，控制x增减
dy	取值+1或-1，控制y增减
cnt	计数器，x和y满足条件的数量，取值0，1，2

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

关键语句

x,y同时满足条件，循环结束

x值为1或n时，cnt+1，同时  
dx改变符号，x的变化改变  
方向

y值为1或m时，cnt+1，同时  
dy改变符号，y的变化改变  
方向

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

带入自拟数据

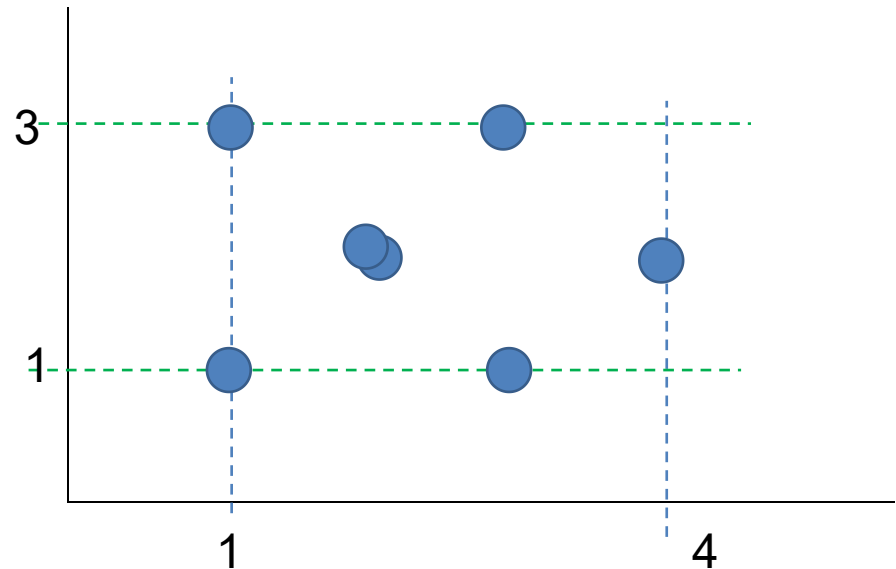
输入：  
4 3

x	dx	y	dy	cnt
1	1	1	1	0
2	1	2	1	0
3	1	3	-1	1
4	-1	2	-1	1
3	-1	1	1	1
2	-1	2	1	0
1	1	3	-1	2

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

x	dx	y	dy	cnt
1	1	1	1	0
2	1	2	1	0
3	1	3	-1	1
4	-1	2	-1	1
3	-1	1	1	1
2	-1	2	1	0
1	1	3	-1	2





# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

判断 程序结束时，cnt的值是3。( )

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

判断 输入的数值n和m相等，那么程序输出n n。( )

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

判断 程序运行过程中x的值可以减小到0。( )

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

判断 输入1 1，输出1 1。( )

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

判断

如果 $n$ 和 $m$ 互质，那么程序需要循环 $(n-1)*(m-1)$ 次。  
( )



# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

选择

输入n m，下面哪一组输入，输出的结果跟输入数字相同仍是n m( )。

A. 3 5    B. 4 6    C. 5 7    D. 2 5

# 阅读程序

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int n, m;
5      cin >> n >> m;
6      int x = 1;
7      int y = 1;
8      int dx = 1;
9      int dy = 1;
10     int cnt = 0;
11     while (cnt != 2) {
12         cnt = 0;
13         x = x + dx;
14         y = y + dy;
15         if (x == 1 || x == n) {
16             ++cnt;
17             dx = -dx;
18         }
19         if (y == 1 || y == m) {
20             ++cnt;
21             dy = -dy;
22         }
23     }
24     cout << x << " " << y << endl;
25     return 0;
26 }
```

选择 输入n m，下面哪一组输入，输出的结果是1 m ( )

A. 2332 3886

B. 2019 2020

C. 2020 2021

D. 100 1000

# 完善程序

（序列重排）全局数组变量 $a$ 定义如下：

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

```
int a[SIZE],n;
```

它记录着一个长度为 $n$ 的序列 $a[1],a[2],\dots,a[n]$ 。

现在需要一个函数，以整数 $p$ （ $1 \leq p \leq n$ ）为参数，实现如下功能：将序列 $a$ 的前 $p$ 个数与后 $n-p$ 个数对调，且不改变这 $p$ 个数

（或 $n-p$ 个数）之间的相对位置。例如，长度为5的序列1,2,3,4,5，当 $p=2$ 时重排结果为3,4,5,1,2。

有一种朴素的算法可以实现这一需求，其时间复杂度为 $O(n)$ 、空间复杂度为 $O(n)$ 。对应`swap1()`函数

我们也可以用时间换空间，使用时间复杂度为 $O(n^2)$ 、空间复杂度为 $O(1)$ 的算法。对应`swap2()`函数

# 手算样例

输入样例：

7

1 2 3 4 5 6 7

（7个整数输入到数组a）

5

（传入函数swap的参数p）

输出多少？

输出样例：

6 7 1 2 3 4 5

# 完善程序

```

3  const int SIZE=100;
4  int a[SIZE],n;
5  void swap1(int p){
6      int i,j,b[SIZE];
7      for(i=1;i<=p;i++)
8          b[____(1)____]=a[i];
9      for(i=p+1;i<=n;i++)
10         b[i-p]=____(2)____;
11     for(i=1;i<=____(3)____;i++)
12         a[i]=b[i];
13 }
14 void swap2(int p){
15     int i,j,temp;
16     for(i=p+1;i<=n;i++){
17         temp=a[i];
18         for(j=i;j>=____(4)____;j--)
19             a[j]=a[j-1];
20         ____ (5) ____=temp;
21     }
22 }

```

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

```
3  const int SIZE=100;
4  int a[SIZE],n;
5  void swap1(int p){
6      int i,j,b[SIZE];
7      for(i=1;i<=p;i++)
8          b[____(1)____]=a[i];
9      for(i=p+1;i<=n;i++)
10         b[i-p]=____(2)____;
11     for(i=1;i<=____(3)____;i++)
12         a[i]=b[i];
13 }
14 void swap2(int p){
15     int i,j,temp;
16     for(i=p+1;i<=n;i++){
17         temp=a[i];
18         for(j=i;j>=____(4)____;j--)
19             a[j]=a[j-1];
20         ____ (5) ____=temp;
21     }
22 }
```

n表示数组元素个数

a[i]表示i号元素的值

参数p表示 将数组a前p  
个数与后n-p个数对调

数组b存储swap后的结果

变量temp临时存储a[i]

# 完善程序

```

3  const int SIZE=100;
4  int a[SIZE],n;
5  void swap1(int p){
6      int i,j,b[SIZE];
7      for(i=1;i<=p;i++)
8          b[____(1)____]=a[i];
9      for(i=p+1;i<=n;i++)
10         b[i-p]=____(2)____;
11     for(i=1;i<=____(3)____;i++)
12         a[i]=b[i];
13 }
14 void swap2(int p){
15     int i,j,temp;
16     for(i=p+1;i<=n;i++){
17         temp=a[i];
18         for(j=i;j>=____(4)____;j--)
19             a[j]=a[j-1];
20         ____ (5) ____=temp;
21     }
22 }

```

要让数组a中第i个数  
( $i=1,2,\dots,p$ ) 分别顺序  
存储到数组b中n-p个数  
之后的对应序号上

# 完善程序

```

3  const int SIZE=100;
4  int a[SIZE],n;
5  void swap1(int p){
6      int i,j,b[SIZE];
7      for(i=1;i<=p;i++)
8          b[____(1)____]=a[i];
9      for(i=p+1;i<=n;i++)
10         b[i-p]=____(2)____;
11     for(i=1;i<=____(3)____;i++)
12         a[i]=b[i];
13 }
14 void swap2(int p){
15     int i,j,temp;
16     for(i=p+1;i<=n;i++){
17         temp=a[i];
18         for(j=i;j>=____(4)____;j--)
19             a[j]=a[j-1];
20         ____ (5) ____=temp;
21     }
22 }

```

再让数组a第i个数  
( $i=p+1, p+2, \dots, n$ )  
分别存到数组b最前面  
也就对应b中 $i-p$ 号元素

# 完善程序

```

3  const int SIZE=100;
4  int a[SIZE],n;
5  void swap1(int p){
6      int i,j,b[SIZE];
7      for(i=1;i<=p;i++)
8          b[____(1)____]=a[i];
9      for(i=p+1;i<=n;i++)
10         b[i-p]=____(2)____;
11         for(i=1;i<=____(3)____;i++)
12             a[i]=b[i];
13     }
14 void swap2(int p){
15     int i,j,temp;
16     for(i=p+1;i<=n;i++){
17         temp=a[i];
18         for(j=i;j>=____(4)____;j--){
19             a[j]=a[j-1];
20             ____ (5) ____=temp;
21         }
22     }

```

将数组b复制给a



# 完善程序

```

3  const int SIZE=100;
4  int a[SIZE],n;
5  void swap1(int p){
6      int i,j,b[SIZE];
7      for(i=1;i<=p;i++)
8          b[____(1)____]=a[i];
9      for(i=p+1;i<=n;i++)
10         b[i-p]=____(2)____;
11     for(i=1;i<=____(3)____;i++)
12         a[i]=b[i];
13 }
14 void swap2(int p){
15     int i,j,temp;
16     for(i=p+1;i<=n;i++){
17         temp=a[i];
18         for(j=i;j>=____(4)____;j--)
19             a[j]=a[j-1];
20         ____ (5) ____=temp;
21     }
22 }

```

对从p+1开始的后n-p个数  
先将每个a[i]取出暂存temp  
再将从a[i-1]开始的数都往后  
挪动一位 即a[j]=a[j-1]  
最后将temp放在i-p的位置上



# 完善程序

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

```
4  int a[SIZE],n;
```

```
5  void swap1(int p){
```

```
6      int i,j,b[SIZE];
```

```
7      for(i=1;i<=p;i++)
```

```
8          b[____(1)____]=a[i];
```

```
9      for(i=p+1;i<=n;i++)
```

```
10         b[i-p]=____(2)____;
```

```
11         for(i=1;i<=____(3)____;i++)
```

```
12             a[i]=b[i];
```

```
13     }
```

```
14 void swap2(int p){
```

```
15     int i,j,temp;
```

```
16     for(i=p+1;i<=n;i++){
```

```
17         temp=a[i];
```

```
18         for(j=i;j>=____(4)____;j--)
```

```
19             a[j]=a[j-1];
```

```
20             ____ (5) ____=temp;
```

```
21         }
```

```
22     }
```

数组a第i个数 ( $i=1,2,\dots,p$ ) 分别顺序  
存储到数组b中n-p个数之后的序号上

让数组a第i个数 ( $i=p+1,p+2,\dots,n$ )  
分别存到数组b最前面位置  
也就赋值到数组b的i-p号元素中

将数组b复制给a完成序列重排

对从p+1开始的后n-p个数  
先将每个a[i]取出暂存temp  
再将从a[i-1]开始的数都往  
后挪动一位 即a[j]=a[j-1]  
最后将temp放到i-p的位置

# 完善程序

```
3  const int SIZE=100;
4  int a[SIZE],n;
5  void swap1(int p){
6      int i,j,b[SIZE];
7      for(i=1;i<=p;i++)
8          b[n-p+i]=a[i];
9      for(i=p+1;i<=n;i++)
10         b[i-p]=a[i];
11     for(i=1;i<=n;i++)
12         a[i]=b[i];
13 }
14 void swap2(int p){
15     int i,j,temp;
16     for(i=p+1;i<=n;i++){
17         temp=a[i];
18         for(j=i;j>=i-p+1;j--){
19             a[j]=a[j-1];
20             a[i-p]=temp;
21         }
22     }
```

数组a第i个数 ( $i=1,2,\dots,p$ ) 分别存到数组b中n-p个数之后序号上

让数组a第i个数 ( $i=p+1,p+2,\dots,n$ ) 分别存到数组b最前面位置也就赋值到数组b的i-p号元素中

将数组b复制给a完成序列重排

需要将第i ( $i=p+1,p+2,\dots,n$ ) 号元素平移到第i-p的位置上所以从i到i-p+1都需要被前面一个数替代也就是 $a[j]=a[j-1]$ 的过程

最后将temp放到i-p的位置

# 读入整数

（读入整数）请完善下面的程序，使得程序能够读入两个int范围内的整数，并将这两个整数分别输出，每行一个。

输入的整数之间和前后只会出现空格或者回车。输入数据保证合法。

例如：

输入：

123 -789

输出：

123

-789

# 完善程序

```

1  #include<iostream>
2  using namespace std;
3  int readint() {
4      int num = 0;           // 存储读取到的整数
5      int negative = 0;      // 负数标志
6      char c;               // 存储当前读取到的字符
7      c = cin.get();
8      while ((c < '0' || c > '9') && c != '-')
9          c = ____ (1) ____;
10     if (c == '-')
11         negative = 1;
12     else
13         ____ (2) ____;
14     c = cin.get();
15     while (____ (3) ____) {
16         ____ (4) ____;
17         c = cin.get();
18     }
19     if (negative == 1)
20         ____ (5) ____;
21     return num;
22 }
23 int main() {
24     int a, b;
25     a = readint();
26     b = readint();
27     cout << a << endl << b << endl;
28     return 0;
29 }

```

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

## 解释变量的作用

```
1  #include<iostream>
2  using namespace std;
3  int readint() {
4      int num = 0;           // 存储读取到的整数
5      int negative = 0;      // 负数标志
6      char c;               // 存储当前读取到的字符
7      c = cin.get();
8      while ((c < '0' || c > '9') && c != '-')
9          c = ____ (1) ____;
10     if (c == '-')
11         negative = 1;
12     else
13         ____ (2) ____;
14     c = cin.get();
15     while (____ (3) ____) {
16         ____ (4) ____;
17         c = cin.get();
18     }
19     if (negative == 1)
20         ____ (5) ____;
21     return num;
22 }
23 int main() {
24     int a, b;
25     a = readint();
26     b = readint();
27     cout << a << endl << b << endl;
28     return 0;
29 }
```

num

出题人已给出

negative

出题人已给出

c

出题人已给出

readint

读取字符里的整数，返回  
其数值



# 完善程序

```
1  #include<iostream>
2  using namespace std;
3  int readint() {
4      int num = 0;           // 存储读取到的整数
5      int negative = 0;      // 负数标志
6      char c;               // 存储当前读取到的字符
7      c = cin.get();
8      while ((c < '0' || c > '9') && c != '-')
9          c = ____ (1) ____;
10     if (c == '-')
11         negative = 1;
12     else
13         ____ (2) ____;
14     c = cin.get();
15     while (____ (3) ____) {
16         ____ (4) ____;
17         c = cin.get();
18     }
19     if (negative == 1)
20         ____ (5) ____;
21     return num;
22 }
23 int main() {
24     int a, b;
25     a = readint();
26     b = readint();
27     cout << a << endl << b << endl;
28     return 0;
29 }
```

关键语句

跳过无关字符

c是负号，另一种情况c是  
数字字符

继续读取数字剩余部分

num是数字绝对值  
负数返回-num

# 完善程序

```
3 int readint() {  
4     int num = 0;           // 存储读取到的整数  
5     int negative = 0;      // 负数标志  
6     char c;                // 存储当前读取到的字符  
7     c = cin.get();  
8     while ((c < '0' || c > '9') && c != '-')  
9         c = cin.get();  
10    if (c == '-')  
11        negative = 1;  
12    else  
13        num = c - '0';  
14    c = cin.get();  
15    while (c >= '0' && c <= '9')  
16        num = num * 10 + c - '0';  
17        c = cin.get();  
18    }  
19    if (negative == 1)  
20        num = -num;  
21    return num;  
22 }
```

# 完善程序

237 读入整数

1979 序列重排