

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

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

#### 解释变量的作用

输入字符串

计数器

循环变量

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

```
#include <iostream>
                                                     关键语句
 2 #include <string>
   using namespace std;
   int main() {
 5
        string str;
        int i;
 6
 7
        int count;
 8
        count = 0;
                                                     遍历字符串str
 9
        getline(cin, str);
        for (i = 0; i < str.length(); i++)</pre>
10 申
                                                   判断str[i]是小写字符,
            if(str[i] >= 'a' && str[i] <= 'z')
11
                                                         count++
12
                count++;
            if(str[i] >= 'A' && str[i] <= 'Z')
13
                                                   判断str[i]是大写字符,
14
                count++;
                                                         coutn++
15
16
        cout <<count << endl;
17
        return 0;
18 L
```

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

判断 程序删除第8行,运行结果不变。( )

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

判断 将11、12行代码与13、14行代码交换,运行结果不变。(

```
1 #include <iostream>
 2 #include <string>
    using namespace std;
 4 pint main() {
        string str;
        int i;
 6
        int count;
 8
        count = 0;
        getline(cin, str);
        for (i = 0; i < str.length(); i++) {</pre>
10日
            if(str[i] >= 'a' && str[i] <= 'z')
11
12
                count++;
            if(str[i] >= 'A' && str[i] <= 'Z')</pre>
13
                count++;
14
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()( )

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

判断 将11行、14行删除,程序运行结果不变( )

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

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

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

```
#include <iostream>
    using namespace std;
 3 poid fun(char *a, char *b) {
 4
        a = b;
 5
        (*a)++;
 7 pint main() {
 8
        char c1, c2, *p1, *p2;
 9
        c1 = 'A';
        c2 = 'a';
10
11
        p1 = &c1;
        p2 = &c2;
12
        fun(p1, p2);
13
        cout << c1 << c2 << endl;
14
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

#### 解释变量的作用 #include <iostream> using namespace std; 3 poid fun(char \*a, char \*b) { 字符指针 a 4 a = b; b 字符指针 5 (\*a)++; 7 **int** main() { char c1, c2, \*p1, \*p2; 8 c1 字符变量 9 c1 = 'A';c2 字符变量 10 c2 = 'a';\*p1 字符指针,指向c1 11 p1 = &c1;12 p2 = &c2;\*p2 字符指针,指向c2 fun(p1, p2); 13 14 cout << c1 << c2 << endl; 15 return 0; 16

```
#include <iostream>
                                           关键语句
   using namespace std;
 3 poid fun(char *a, char *b)
 4
                               a指向b所指向的变量
 5
       (*a)++;
                                a指向的变量数值+1
7 pint main() {
       char c1, c2, *p1, *p2;
 8
 9
       c1 = 'A';
10
       c2 = 'a';
11
       p1 = &c1;
                               p1指向c1, p2指向c2
12
       p2 = &c2;
       fun(p1, p2);
13
                                调用函数,将p1和p2作为参数,
       cout << c1 << c2 << endl:
14
                                以通过指针修改变量c1和c2的数值
       return 0;
15
16
```

```
#include <iostream>
   using namespace std;
 3 poid fun(char *a, char *b) {
 4
        a = b;
 5
        (*a)++;
 7 pint 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 <sup>L</sup> }
```

判断 将第9行c1='A'改为c1="A",程序运行结果不变。(

```
#include <iostream>
   using namespace std;
 3 poid fun(char *a, char *b) {
 4
       a = b;
 5
       (*a)++;
 7 pint 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 L }
```

判断 11行也可以写成\*p1=c1。( )

```
#include <iostream>
   using namespace std;
 3 poid fun(char *a, char *b) {
 4
       a = b;
 5
       (*a)++;
 7 pint 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 <sup>L</sup> }
```

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()
```

```
#include <iostream>
    using namespace std;
 3 poid fun(char *a, char *b) {
 4
        a = b;
 5
        (*a)++;
 7 pint 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 <sup>L</sup> }
```

判断 在3-5行函数中,变量c1和c2的数值都会改变(

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

选择 程序输出结果()。

A. Aa B. Ab C. Ba D. ab

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

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

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

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

### 解释变量的作用

n	输入,边界值
m	输入,边界值
X	数值从1到n再到1,循环
у	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

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

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

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

#### 带入自拟数据

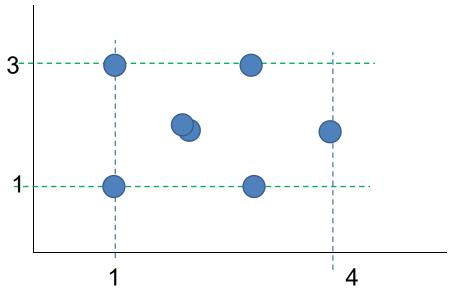
输入:

43

X	dx	У	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

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

х	dx	у	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



```
1 #include <iostream>
    using namespace std;
                                   阅读程序
 3 pint main() {
        int n, m;
 4
 5
        cin >> n >> m;
 6
        int x = 1;
 7
        int y = 1;
 8
        int dx = 1;
        int dy = 1;
 9
        int cnt = 0;
10
11 申
        while (cnt != 2) {
12
            cnt = 0;
13
            x = x + dx;
            y = y + dy;
14
            if (x == 1 || x == n) {
15 🗦
16
                ++cnt;
                dx = -dx;
17
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。( )

```
1 #include <iostream>
    using namespace std;
                                   阅读程序
 3 pint main() {
 4
        int n, m;
 5
        cin >> n >> m;
 6
        int x = 1;
 7
        int y = 1;
 8
        int dx = 1;
        int dy = 1;
 9
        int cnt = 0;
10
        while (cnt != 2) {
11 申
12
            cnt = 0;
13
            x = x + dx;
            y = y + dy;
14
            if (x == 1 || x == n) {
15 
16
                ++cnt;
                dx = -dx;
17
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。( )

```
1 #include <iostream>
    using namespace std;
                                   阅读程序
 3 pint main() {
        int n, m;
 4
 5
        cin >> n >> m;
 6
        int x = 1;
 7
        int y = 1;
 8
        int dx = 1;
        int dy = 1;
 9
        int cnt = 0;
10
11 申
        while (cnt != 2) {
12
            cnt = 0;
13
            x = x + dx;
            y = y + dy;
14
            if (x == 1 || x == n) {
15 
16
                ++cnt;
                dx = -dx;
17
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。( )

```
1 #include <iostream>
    using namespace std;
                                   阅读程序
 3 pint 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 申
        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11, 输出11。( )

```
1 #include <iostream>
    using namespace std;
                                   阅读程序
 3 □ int main() {
        int n, m;
 4
 5
        cin >> n >> m;
 6
        int x = 1;
 7
        int y = 1;
 8
        int dx = 1;
        int dy = 1;
 9
        int cnt = 0;
10
        while (cnt != 2) {
11 申
12
            cnt = 0;
13
            x = x + dx;
14
            y = y + dy;
            if (x == 1 || x == n) {
15 
16
                ++cnt;
                dx = -dx;
17
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。
```

```
1 #include <iostream>
    using namespace std;
 3 pint main() {
        int n, m;
 4
 5
        cin >> n >> m;
        int x = 1;
 6
 7
        int y = 1;
 8
        int dx = 1;
 9
        int dy = 1;
10
        int cnt = 0;
        while (cnt != 2) {
11 申
12
            cnt = 0;
13
            x = x + dx;
14
            y = y + dy;
15 申
            if (x == 1 || x == n) {
16
                ++cnt;
17
                dx = -dx;
18
            if (y == 1 || y == m) {
19申
20
                ++cnt;
21
                dv = -dv;
22
23
        cout << x << " " << v << endl;
24
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. 35 B. 46 C. 57 D. 25

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

(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],...a[n]。 现在需要一个函数,以整数p(1≤p≤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(n2)、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

```
const int SIZE=100;
                                                  识别变量
   int a[SIZE],n;
5 p void swap1(int p){
                                              常见变量名
       int i,j,b[SIZE];
6
                                             翻译循环变量
       for(i=1;i<=p;i++)</pre>
                                          根据变量名的英文推断
8
           b[____(1)___]=a[i];
       for(i=p+1;i<=n;i++)</pre>
           b[i-p]=___(2)___
10
                                           2
                                                找出关键语句
       for(i=1;i<=___(3)___;i++)
11
12
           a[i]=b[i];
                                            控制结构(for, if)
13
                                           常见算法的基本操作
14 poid swap2(int p){
                                            函数参数、返回值
15
       int i,j,temp;
       for(i=p+1;i<=n;i++){</pre>
16 🗦
17
           temp=a[i];
           for(j=i;j>=____(4)___;j--)
18
                                           3
                                               理解代码段作用
               a[j]=a[j-1];
19
            ____(5)___=temp;
20
                                            翻译解释代码段
21
22
```

const int SIZE=100;

```
int a[SIZE],n;
5 p void swap1(int p){
       int i,j,b[SIZE];
6
                                        n表示数组元素个数
       for(i=1;i<=p;i++)</pre>
           b[____(1)___]=a[i];
8
                                        a[i]表示i号元素的值
       for(i=p+1;i<=n;i++)</pre>
           b[i-p]=___(2)__
10
                             ;i++)
       for(i=1;i<=___(3)___
                                      参数p表示 将数组a前p
11
           a[i]=b[i];
12
                                       个数与后n-p个数对调
13
14 poid swap2(int p){
                                     数组b存储swap后的结果
       int i,j,temp;
15
       for(i=p+1;i<=n;i++){</pre>
16 🗦
                                       变量temp临时存储a[i]
           temp=a[i];
17
           for(j=i;j>=___(4)___;j--)
18
              a[j]=a[j-1];
19
              ___(5)____=temp;
20
21
22
```

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

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

```
const int SIZE=100;
   int a[SIZE],n;
 5 p void swap1(int p){
       int i,j,b[SIZE];
 6
                                           再让数组a第i个数
       for(i=1;i<=p;i++)</pre>
                                           (i=p+1,p+2,...,n)
 8
           b[____(1)__
                           _]=a[i];
                                         分别存到数组b最前面
       for(i=p+1;i<=n;i++)</pre>
                                         也就对应b中i-p号元素
           b[i-p] = (2)
10
       for(i=1;i<= (3)
11
           a[i]=b[i];
12
13
14 poid swap2(int p){
        int i,j,temp;
15
       for(i=p+1;i<=n;i++){</pre>
16 🗦
           temp=a[i];
17
            for(j=i;j>=___(4)___;j--)
18
               a[j]=a[j-1];
19
                 _(5)____=temp;
20
21
22
```

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

```
const int SIZE=100;
   int a[SIZE],n;
5 p void swap1(int p){
       int i,j,b[SIZE];
6
       for(i=1;i<=p;i++)
8
           b[____(1)___]=a[i];
       for(i=p+1;i<=n;i++)</pre>
           b[i-p]=____(2)____;
10
       for(i=1;i<=___(3)___;i++)
                                        对从p+1开始的后n-p个数
11
                                       先将每个a[i]取出暂存temp
           a[i]=b[i];
12
                                       再将从a[i-1]开始的数都往后
13
                                         挪动一位 即a[i]=a[i-1]
14 poid swap2(int p){
                                       最后将temp放在i-p的位置上
15
       int i,j,temp;
       for(i=p+1;i<=n;i++){</pre>
16 
           temp=a[i];
17
           for(j=i;j>=___(4)___;j--)
18
               a[j]=a[j-1];
19
            (5) =temp;
20
21
22
```

```
const int SIZE=100;
   int a[SIZE],n;
 5 p void swap1(int p){
       int i,j,b[SIZE];
 6
                                     数组a第i个数(i=1,2,...,p)分别顺序
       for(i=1;i<=p;i++)</pre>
                                     存储到数组b中n-p个数之后的序号上
           b[ (1)_
                            ]=a[i];
 8
       for(i=p+1;i<=n;i++)</pre>
                                       让数组a第i个数(i=p+1,p+2,...,n)
           b[i-p]=_
                                          分别存到数组b最前面位置
10
                                 ;i++)
                                        也就赋值到数组b的i-p号元素中
       for(i=1;i<=___(3)__
11
12
           a[i]=b[i];
                                         将数组b复制给a完成序列重排
13
14 poid swap2(int p){
15
       int i,j,temp;
       for(i=p+1;i<=n;i++){</pre>
16 申
                                            对从p+1开始的后n-p个数
17
           temp=a[i];
                                            先将每个a[i]取出暂存temp
            for(j=i;j>=____(4)__
18
                                            再将从a[i-1]开始的数都往
               a[j]=a[j-1];
19
                                             后挪动一位 即a[j]=a[j-1]
                __(5)___=temp;
20
                                            最后将temp放到i-p的位置
21
```

22

```
const int SIZE=100;
   int a[SIZE],n;
 5 p void swap1(int p){
       int i,j,b[SIZE];
 6
                                        数组a第i个数(i=1,2,...,p)分别
       for(i=1;i<=p;i++)</pre>
                                        存到数组b中n-p个数之后序号上
 8
           b[_ n-p+i
                           _]=a[i];
       for(i=p+1;i<=n;i++)</pre>
                                       让数组a第i个数(i=p+1,p+2,...,n)
           b[i-p]=\underline{a[i]}
                                          分别存到数组b最前面位置
10
                                         也就赋值到数组b的i-p号元素中
       for(i=1;i<= n
11
           a[i]=b[i];
12
                                         将数组b复制给a完成序列重排
13
14 poid swap2(int p){
                                           需要将第i(i=p+1,p+2,...n)
        int i,j,temp;
15
                                           号元素平移到第i-p的位置上
       for(i=p+1;i<=n;i++){</pre>
                                           所以从i到i-p+1都需要被前面
16 🗦
                                                 一个数替代
           temp=a[i];
17
                                             也就是a[j]=a[j-1]的过程
           for(j=i;j>= i-p+1
18
               a[j]=a[j-1];
19
               a[i-p]
                                            最后将temp放到i-p的位置
                          =temp;
20
21
```

22

### 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

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

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

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

#### 解释变量的作用

num	出题人已给出			
negative	出题人已给出			
С	出题人已给出			
readint	读取字符里的整数,返回 其数值			

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

#### 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

```
3 ☐ int readint() {
       int num = 0;  // 存储读取到的整数
4
       int negative = 0; // 负数标志
5
       char c; // 存储当前读取到的字符
6
       c = cin.get();
8
       while ((c < '0' || c > '9') && c != '-')
9
          c = cin.get()
10
       if (c == '-')
11
           negative = 1;
12
       else
             num=c-'0'
13
14
       c = cin.get();
       while (c)=0.88c<=9.9
15 E
          num=num*10+c-'0'
16
17
           c = cin.get();
18
       if (negative == 1)
19
20
                num=-num
21
       return num;
22
```

水kcoding.net

### 完善程序

237 读入整数

1979 序列重排