

【华为OD机考 统一考试机试C卷】素数之积/RSA加密算法（C++ Java JavaScript Python C语言）

华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

目前在考C卷，经过两个月的收集整理，**C卷真题已基本整理完毕**
抽到原题的概率为2/3到3/3，**也就是最少抽到两道原题。请注意：大家刷完C卷真题，最好要把B卷的真题刷一下，因为C卷的部分真题来自B卷。**
另外订阅专栏还可以联系笔者开通在线 **OJ** 进行刷题，提高刷题效率。
真题目录：华为OD机考机试 真题目录（C卷 + D卷 + B卷 + A卷） + 考点说明
专栏：2023华为OD机试(B卷+C卷+D卷) (C++JavaJSPy)
华为OD面试真题精选：华为OD面试真题精选
在线OJ：点击立即刷题，模拟真实机考环境

题目描述

RSA加密算法在网络安全世界中无处不在，它利用了极大整数因数分解的困难度，数据越大，安全系数越高，给定一个32位正整数，请对其进行因数分解，找出是哪两个素数的乘积。

输入描述

一个正整数num， $0 < num \leq 2147483647$

输出描述

如果成功找到，以单个空格分割，从小到大输出两个素数，分解失败，请输出-1, -1

用例

输入	15
输出	3 5

输入	27
输出	-1 -1

解题思路

C++

```
1  #include <iostream>
2  #include <cmath>
3
4  // 函数: 检查一个数是否为素数
5  bool isPrime(int num) {
6      if (num <= 3) {
7          return num > 1;
8      }
9      if (num % 6 != 1 && num % 6 != 5) {
10         return false;
11     }
12     for (int i = 5; i <= sqrt(num); i += 6) {
13         if (num % i == 0 || num % (i + 2) == 0) {
14             return false;
15         }
16     }
17     return true;
18 }
19
20 int main() {
21     int num;
22     std::cin >> num; // 读取输入
23
24     // 如果数字本身就是素数, 那么它不能被分解
25     if (isPrime(num)) {
26         std::cout << "-1 -1" << std::endl;
27         return 0;
28     }
29
30     // 分解数字
31     for (int i = 2; i <= sqrt(num); i++) {
32
```

```

33     if (num % i == 0) {
34         int j = num / i;
35         // 检查 i 和 j 是否都是素数
36         if (isPrime(i) && isPrime(j)) {
37             std::cout << (i < j ? std::to_string(i) + " " + std::to_string(j) : std::to_string(j) + " " + std::to_string(i)) << std::endl;
38             return 0;
39         }
40     }
41 }
42 std::cout << "-1 -1" << std::endl;
43 return 0;
44 }
45

```

Java

```

1  import java.util.Scanner;
2
3  public class Main {
4      public static boolean isPrime(int num) {
5          if (num <= 3) {
6              return num > 1;
7          }
8          if (num % 6 != 1 && num % 6 != 5) {
9              return false;
10         }
11         for (int i = 5; i <= Math.sqrt(num); i += 6) {
12             if (num % i == 0 || num % (i + 2) == 0) {
13                 return false;
14             }
15         }
16         return true;
17     }
18
19     public static void main(String[] args) {
20         Scanner scanner = new Scanner(System.in);
21         int num = scanner.nextInt();
22         // 如果输入的数本身就是素数, 无法进行因数分解
23         if (isPrime(num)) {
24

```

```

24         System.out.println("-1 -1");
25         return;
26     }
27     // 因数分解
28     for (int i = 2; i <= Math.sqrt(num); i++) {
29         if (num % i == 0) {
30             int j = num / i;
31             // 判断 i 和 j 是否都是素数
32             if (isPrime(i) && isPrime(j)) {
33                 System.out.println(i < j ? i + " " + j : j + " " + i);
34                 return;
35             }
36         }
37     }
38     System.out.println("-1 -1");
39 }
40 }
41 }
42

```

JavaScript

```

1  const readline = require('readline').createInterface({
2      input: process.stdin,
3      output: process.stdout
4  });
5
6  // 函数: 检查一个数是否为素数
7  function isPrime(num) {
8      if (num <= 3) {
9          return num > 1;
10     }
11     if (num % 6 !== 1 && num % 6 !== 5) {
12         return false;
13     }
14     for (let i = 5; i <= Math.sqrt(num); i += 6) {
15         if (num % i == 0 || num % (i + 2) == 0) {
16             return false;
17         }
18     }
19 }

```

```

19     return true;
20 }
21
22 readline.on('line', num => {
23     num = parseInt(num);
24     // 如果数字本身就是素数, 那么它不能被分解
25     if (isPrime(num)) {
26         console.log("-1 -1");
27         readline.close();
28         return;
29     }
30
31     // 分解数字
32     for (let i = 2; i <= Math.sqrt(num); i++) {
33         if (num % i == 0) {
34             let j = num / i;
35             // 检查 i 和 j 是否都是素数
36             if (isPrime(i) && isPrime(j)) {
37                 console.log(i < j ? i + " " + j : j + " " + i);
38                 readline.close();
39                 return;
40             }
41         }
42     }
43     console.log("-1 -1");
44     readline.close();
45 });
46
47

```

Python

```

1 import math
2
3 # 函数: 检查一个数是否为素数
4 def is_prime(num):
5     if num <= 3:
6         return num > 1
7     if num % 6 != 1 and num % 6 != 5:
8         return False
9

```

```

9     for i in range(5, int(math.sqrt(num)) + 1, 6):
10         if num % i == 0 or num % (i + 2) == 0:
11             return False
12     return True
13
14 num = int(input())
15
16 # 如果数字本身就是素数, 那么它不能被分解
17 if is_prime(num):
18     print("-1 -1")
19 else:
20     # 分解数字
21     for i in range(2, int(math.sqrt(num)) + 1):
22         if num % i == 0:
23             j = num // i
24             # 检查 i 和 j 是否都是素数
25             if is_prime(i) and is_prime(j):
26                 print(f"{min(i, j)} {max(i, j)}")
27                 break
28     else:
29         print("-1 -1")
30
31

```

C语言

```

1  #include <stdio.h>
2  #include <math.h>
3
4  // 函数: 检查一个数是否为素数
5  int isPrime(int num) {
6      if (num <= 3) {
7          return num > 1;
8      }
9      if (num % 6 != 1 && num % 6 != 5) {
10         return 0;
11     }
12     for (int i = 5; i <= sqrt(num); i += 6) {
13         if (num % i == 0 || num % (i + 2) == 0) {
14             return 0;
15         }
16     }
17     return 1;
18 }
19
20 int main() {
21     int num;
22     scanf("%d", &num);
23     if (isPrime(num)) {
24         printf("是素数\n");
25     } else {
26         printf("不是素数\n");
27     }
28     return 0;
29 }

```

```

15     }
16 }
17 return 1;
18 }
19
20 int main() {
21     int num;
22     scanf("%d", &num); // 读取输入
23
24     // 如果数字本身就是素数, 那么它不能被分解
25     if (isPrime(num)) {
26         printf("-1 -1\n");
27         return 0;
28     }
29
30     // 分解数字
31     for (int i = 2; i <= sqrt(num); i++) {
32         if (num % i == 0) {
33             int j = num / i;
34             // 检查 i 和 j 是否都是素数
35             if (isPrime(i) && isPrime(j)) {
36                 printf("%d %d\n", i < j ? i : j, i < j ? j : i);
37                 return 0;
38             }
39         }
40     }
41     printf("-1 -1\n");
42     return 0;
43 }

```

完整用例

用例1

13

用例2

49

用例3

77

用例4

35

用例5

1

用例6

2147483647

用例7

4

用例8

2999

用例9

38

用例10

589

文章目录

- 华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷
- 题目描述
- 输入描述
- 输出描述
- 用例
- 解题思路

C++

Java

JavaScript

Python

C语言

完整用例

用例1

用例2

用例3

用例4

用例5

用例6

用例7

用例8

用例9

用例10

机考真题 华为OD



CSDN @算法大师

