

# 【华为OD机考 统一考试机试C卷】密码解密（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。

**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

给定一段“密文”字符串 s，其中字符都是经过“密码本”映射的，现需要将“密文”解密并输出。

映射的规则（‘a’~‘i’）分别用（‘1’~‘9’）表示；（‘j’~‘z’）分别用（“10\*”~“26\*”）表示。

约束：映射始终唯一。

## 输入描述

“密文”字符串

## 输出描述

明文字符串

备注：翻译后的文本长度在100以内

## 用例

输入	20*19*20*
输出	tst
说明	无

解题思路

暴力替换，需要注释的是先从"10\*" ~ "26\*"开始映射替换

```
1 s=s.replace("10*","j")
2 s=s.replace("11*","k")
3 s=s.replace("12*","l")
4 s=s.replace("13*","m")
5 s=s.replace("14*","n")
6 s=s.replace("15*","o")
7 s=s.replace("16*","p")
8 s=s.replace("17*","q")
9 s=s.replace("18*","r")
10 s=s.replace("19*","s")
11 s=s.replace("20*","t")
12 s=s.replace("21*","u")
13 s=s.replace("22*","v")
14 s=s.replace("23*","w")
15 s=s.replace("24*","x")
16 s=s.replace("25*","y")
17 s=s.replace("26*","z")
18 s=s.replace("1","a")
19 s=s.replace("2","b")
20 s=s.replace("3","c")
21 s=s.replace("4","d")
22 s=s.replace("5","e")
23 s=s.replace("6","f")
24 s=s.replace("7","g")
25 s=s.replace("8","h")
26 s=s.replace("9","i")
```

```

1  #include <iostream>
2  #include <string>
3  #include <regex>
4
5  using namespace std;
6  int main() {
7
8
9      // 定义字符串变量s, 用于存储用户输入的密文
10     string s;
11     // 从标准输入读取一行数据存入s
12     getline(cin, s);
13
14     // 从26开始递减到1, 创建映射并立即使用正则表达式进行全局替换
15     for (int i = 26; i >= 1; --i) {
16         // 构造映射的键: 对于10到26, 添加'*'; 否则使用数字本身
17         string key = to_string(i) + (i >= 10 ? "\\*" : "");
18         // 构造映射的值: ASCII码97对应'a', 因此96+i对应的字符
19         char value = static_cast<char>(96 + i);
20
21         // 使用正则表达式和regex_replace方法, 将密文中的每个加密字符 (键) 替换为对应的字母 (值)
22         s = regex_replace(s, regex(key), string(1, value));
23     }
24
25     // 打印解密后的明文字符串
26     cout << s << endl;
27
28     // 主函数结束
29     return 0;
30 }

```

## Java

```

1  import java.util.HashMap;
2  import java.util.Map;
3  import java.util.Scanner;
4
5  public class Main {
6      public static void main(String[] args) {
7          Scanner scanner = new Scanner(System.in);

```

```

8      // 从用户输入接收密文字符串
9      String s = scanner.nextLine();
10
11     // 创建一个映射HashMap, 用于将加密的'10*'到'26*'映射到对应的字母'j'到'z'
12     Map<String, Character> mapping = new HashMap<>();
13     for (int i = 10; i <= 26; i++) {
14         // 将数字和'*'组合成字符串作为键, 将ASCII码转换得到的字母作为值
15         mapping.put(i + "*", (char) (96 + i));
16     }
17
18     // 更新映射HashMap, 将'1'到'9'映射到'a'到'i'
19     for (int i = 1; i <= 9; i++) {
20         // 将数字转换成字符串作为键, 将ASCII码转换得到的字母作为值
21         mapping.put(String.valueOf(i), (char) (96 + i));
22     }
23
24     // 遍历映射HashMap中的每一对键值对
25     for (Map.Entry<String, Character> entry : mapping.entrySet()) {
26         // 获取键和值
27         String key = entry.getKey();
28         Character value = entry.getValue();
29         // 使用字符串的replace方法, 将密文中的每个加密字符(键)替换为对应的字母(值)
30         s = s.replace(key, value.toString());
31     }
32
33     // 打印解密后的明文字符串
34     System.out.println(s);
35
36     // 关闭扫描器
37     scanner.close();
38 }
39 }

```

## JavaScript

```

1 // 引入readline模块, 用于从命令行读取输入
2 const readline = require('readline');
3
4 // 创建readline.Interface实例, 用于读取标准输入
5 const rl = readline.createInterface({
6

```

```

6   input: process.stdin,
7   output: process.stdout
8   });
9
10  // 提示用户输入密文字符串
11  rl.on('line', (s) => {
12    // 从26开始递减到1, 创建映射并立即使用正则表达式进行全局替换
13    for (let i = 26; i >= 1; i--) {
14      // 构造映射的键: 对于10到26, 添加'*'; 否则使用数字本身
15      const key = i + (i >= 10 ? '*' : '');
16      // 构造映射的值: ASCII码97对应'a', 因此96+i对应的字符
17      const value = String.fromCharCode(96 + i);
18      // 使用字符串的replace方法, 将密文中的每个加密字符(键)替换为对应的字母(值)
19      s = s.replaceAll(key, value);
20    }
21
22    // 打印解密后的明文字符串
23    console.log(s);
24
25    // 关闭readline.Interface实例
26    rl.close();
27  });

```

## Python

```

1  # 从用户输入接收密文字符串
2  s = input()
3
4  # 创建一个映射字典, 用于将'1'到'9'映射到'a'到'i', 以及'10*'到'26*'映射到'j'到'z'
5  # 对于26到1之间的每个数字i, 如果i小于10, 则键是字符串形式的i;
6  # 如果i大于等于10, 则键是字符串形式的i加上'*';
7  # 对应的值是通过ASCII码转换得到的字母 (ASCII码97是'a', 所以96+i就是对应的字母)
8  mapping = {str(i) + ('*' if i >= 10 else ''): chr(96 + i) for i in range(26, 0, -1)}
9
10 # 遍历映射字典中的每一对键值对
11 for key, value in mapping.items():
12     # 使用字符串的replace方法, 将密文中的每个加密字符(键)替换为对应的字母(值)
13     s = s.replace(key, value)
14
15 # 打印解密后的明文字符串
16

```

```
16 print(s)
17
18
19 上面代码等同下
20
21 s=input()
22 s=s.replace("10*", "j")
23 s=s.replace("11*", "k")
24 s=s.replace("12*", "l")
25 s=s.replace("13*", "m")
26 s=s.replace("14*", "n")
27 s=s.replace("15*", "o")
28 s=s.replace("16*", "p")
29 s=s.replace("17*", "q")
30 s=s.replace("18*", "r")
31 s=s.replace("19*", "s")
32 s=s.replace("20*", "t")
33 s=s.replace("21*", "u")
34 s=s.replace("22*", "v")
35 s=s.replace("23*", "w")
36 s=s.replace("24*", "x")
37 s=s.replace("25*", "y")
38 s=s.replace("26*", "z")
39 s=s.replace("1", "a")
40 s=s.replace("2", "b")
41 s=s.replace("3", "c")
42 s=s.replace("4", "d")
43 s=s.replace("5", "e")
44 s=s.replace("6", "f")
45 s=s.replace("7", "g")
46 s=s.replace("8", "h")
47 s=s.replace("9", "i")
48 print(s)
```

## C语言

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5
```

```

5 // 定义一个函数来替换字符串中的所有匹配项
6 char* replace_all(char *str, const char *old, const char *new_str) {
7     char *result;
8     int i, cnt = 0;
9     int new_len = strlen(new_str);
10    int old_len = strlen(old);
11
12    // 计算替换后的字符串长度
13    for (i = 0; str[i] != '\0'; i++) {
14        if (strstr(&str[i], old) == &str[i]) {
15            cnt++;
16            i += old_len - 1;
17        }
18    }
19
20    result = (char *)malloc(i + cnt * (new_len - old_len) + 1);
21
22    i = 0;
23    while (*str) {
24        if (strstr(str, old) == str) {
25            strcpy(&result[i], new_str);
26            i += new_len;
27            str += old_len;
28        }
29        else
30            result[i++] = *str++;
31    }
32
33    result[i] = '\0';
34    return result;
35 }
36
37 int main() {
38     char s[1000]; // 定义字符数组s，用于存储用户输入的密文
39
40     scanf("%[^\n]*c", s); // 从标准输入读取一行数据存入s
41
42     // 创建映射，将加密的'10*'到'26*'映射到对应的字母'j'到'z'
43     for (int i = 10; i <= 26; i++) {
44         char key[4]; // 定义一个字符串来存储键
45

```

```

46     sprintf(key, "%d*", i); // 将数字和'*'组合成字符串作为键
47     char value[2] = {96 + i, '\0'}; // 将ASCII码转换得到的字母作为值
48
49     char *new_s = replace_all(s, key, value); // 替换字符串中的匹配项
50     strcpy(s, new_s); // 将替换后的字符串复制回s
51     free(new_s); // 释放临时字符串
52 }
53
54 // 更新映射, 将'1'到'9'映射到'a'到'i'
55 for (int i = 1; i <= 9; i++) {
56     char key[2] = {i + '0', '\0'}; // 将数字转换成字符串作为键
57     char value[2] = {96 + i, '\0'}; // 将ASCII码转换得到的字母作为值
58
59     char *new_s = replace_all(s, key, value); // 替换字符串中的匹配项
60     strcpy(s, new_s); // 将替换后的字符串复制回s
61     free(new_s); // 释放临时字符串
62 }
63
64 printf("%s\n", s); // 打印解密后的明文字符串
65
66 return 0;
}

```

## 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](#)



# 机考真题 华为OD



CSDN @算法大师