

## 第 1 题

某市去年全年举行某疫苗全民接种, 现今民众可以通过接种号对疫苗接种情况进行查询。接种号为 10 位, 首位大写字母表示 A,B,C 三个接种点, 2~3 位表示接种月份, 4~5 位表示疫苗的种类 (07, 13, 23), 后 5 位为接种者的数字编号。

请用**等价类划分**方法对接种号输入是否合法进行测试。要求给出**等价类表**, 和具体的**覆盖数据**。(每种无效等价类都独立给出一个覆盖数据, 并给出一个正确输入)

参考格式: [《等价类划分》-有这篇就够了 - 知乎 \(zhihu.com\)](#)

## 第 2 题

给出以下两个代码的**环路复杂度**, 并给出所有的**独立路径**, 同时对于每条独立路径给出**完整测试用例以及对应输出**。(流图中的结点标识最好能说明代表程序中的哪些代码段, 或者用相应代码行号作为结点标识, 若代码中存在循环, 可将循环体视为只执行一次, 这样方便写独立路径。测试用例不唯一, 满足路径即可, 对于实际不存在的路径, 进行说明后可不进行测试。)

代码一:

```
1. void lcs(int i, int j_initial, char a[], char b[])
2. {
3.     length1 = strlen(a);
4.     length2 = strlen(b);
5.     while (i <= length1)
6.     {
7.         j = j_initial;
8.         while (j <= length2)
9.         {
10.            if (a[i - 1] == b[j - 1])
11.            {
12.                arr[i][j] = arr[i - 1][j - 1] + 1; //int arr[max][max]={0}
13.            }
14.            else
15.            {
16.                if (arr[i - 1][j] > arr[i][j - 1])
17.                    arr[i][j] = arr[i - 1][j];
18.                else
19.                    arr[i][j] = arr[i][j - 1];
20.            }
21.            j++;
22.        }
23.        i++;
24.    }
25.    printf("%d\n", arr[length1][length2]);
26. }
```

代码二:

```
1. int Isprime(int n){
2.     int i;
3.     if (n < 2)
4.         printf("%d 不是素数", n);
5.     else {
6.         for (i = 2; i < n; i++) {
7.             if (n%i == 0)
8.                 break;
9.         }
10.        if (i < n) {
11.            printf("%d 不是素数\n", n);
12.        }
13.        else
14.            printf("%d 是素数\n", n);
15.    }
16.    return 0;
17. }
```

参考格式

独立路径	测试用例	结果
路径 1: 1, 2……	n = …	
路径 2: …		
路径 3: …		

第三题

请同学们安装和使用 SonarQube，对自己以前写得的程序代码进行检测。找到 5 种不同严重程度的 **bug 例子**并给出其相应的**修复方案与修复结果**。(提供截图和相应的文本描述)

SonarQube 的安装手册另见附件

答案：

第 1 题

某市去年全年举行某疫苗全民接种, 现今民众可以通过接种号对疫苗接种情况进行查询。接种号为 10 位, 首位大写字母表示 A,B,C 三个接种点, 2~3 位表示接种月份, 4~5 位表示疫苗的种类 (07, 13, 23), 后 5 位为接种者的数字编号。

请用**等价类划分**方法对接种号输入是否合法进行测试。要求给出**等价类表**, 和具体的**覆盖数据**。(每种无效等价类都独立给出一个覆盖数据, 并给出一个正确输入)

等价类表：

输入条件	有效等价类	编号	无效等价类	编号
接种号	10 位	1	不为 10 位	2
	首字母为 A,B,C	3	首字母不为 A,B,C	4
	月份合法	5	月份不合法	6
	疫苗种类正确	7	疫苗种类不正确	8
	后 5 位为数字	9	后 5 位包含非数字	10

覆盖数据：

编号	输入数据	覆盖等价类	预期输出
1	A122356465	1,3,5,7,9	OK
2	D110732142	1,4,5,7,9	ERROR
3	B151356421	1,3,6,7,9	ERROR
4	C120112345	1,3,5,8,9	ERROR
5	A0513123&&	1,3,5,7,10	ERROR
6	A091355644445	2,3,5,7,9	ERROR

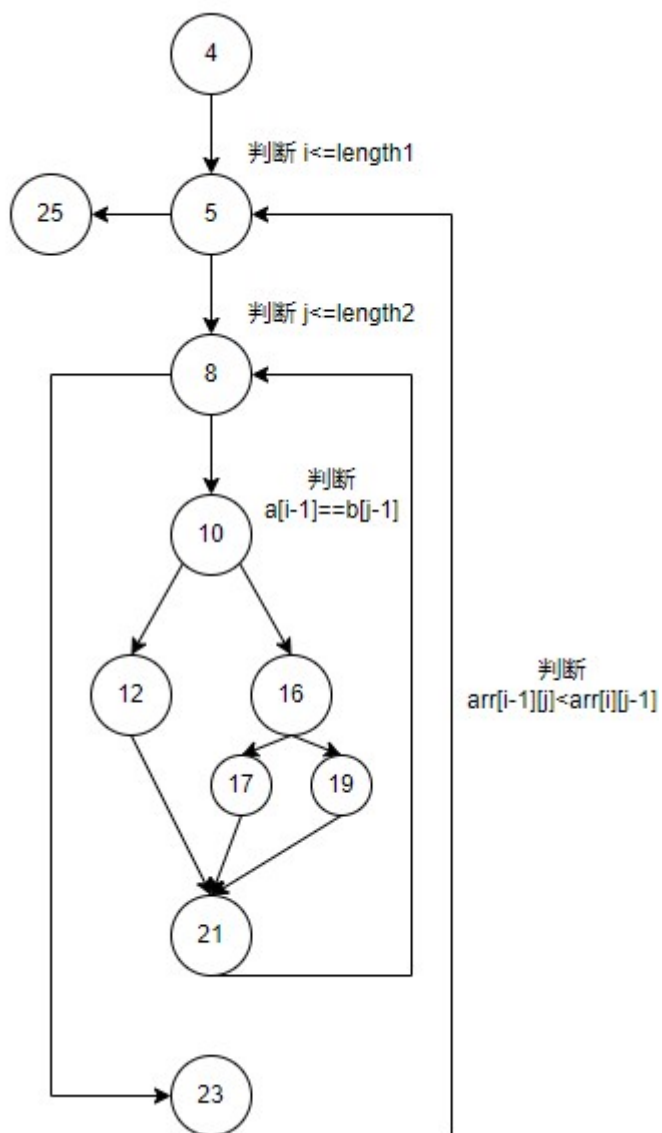
第 2 题

给出以下两个代码的**环路复杂度**, 并给出所有的**独立路径**, 同时对于每条独立路径给出**完整测试用例以及对应输出**。(流图中的结点标识最好能说明代表程序中的哪些代码段, 或者用相应代码行号作为结点标识, 若代码中存在循环, 可将循环体视为只执行一次, 这样方便写独立路径。测试用例不唯一, 满足路径即可, 对于实际不存在的路径, 进行说明后可不进行测试。)

代码一：

```
1. void lcs(int i, int j_initial, char a[], char b[])
2. {
3.     length1 = strlen(a);
4.     length2 = strlen(b);
5.     while (i <= length1)
6.     {
```

```
7.         j = j_initial;
8.         while (j <= length2)
9.         {
10.            if (a[i - 1] == b[j - 1])
11.            {
12.                arr[i][j] = arr[i - 1][j - 1] + 1; //int arr[max][max]={0}
13.            }
14.            else
15.            {
16.                if (arr[i - 1][j] > arr[i][j - 1])
17.                    arr[i][j] = arr[i - 1][j];
18.                else
19.                    arr[i][j] = arr[i][j - 1];
20.            }
21.            j++;
22.        }
23.        i++;
24.    }
25.    printf("%d\n",arr[length1][length2]);
26. }
```

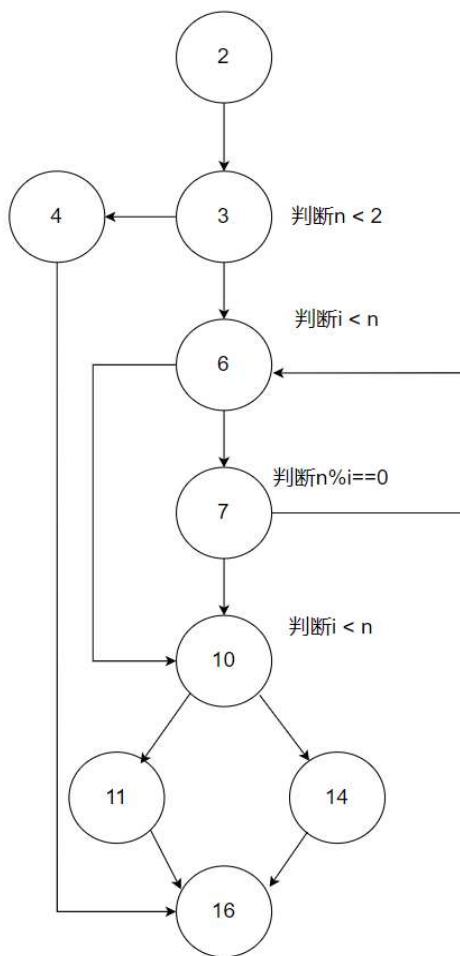


独立路径	输入用例	结果
4→5→25	i=4;j_initial=0;a="123";b="123"	0
4→5→8→23→5→25	i=0;j_initial=4;a="123";b="123"	0
4→5→8→10→12→21→8→23→5→25	i=0;j_initial=0; a="123";b="123"	3
4→5→8→10→12→21→8→10→16→17→21→8→23→5→25	i=0;j_initial=0;a="BD";b="B"	1
4→5→8→10→16→19→21→8→23→5→25	i=0;j_initial=0;a="1";b="2"	0

注：在路径 4 中，因为判断需要满足条件  $arr[i-1][j] > arr[i][j-1]$  则程序至少已经通过行 11 进行过+1。

代码二:

```
1. int Isprime(int n){
2.     int i;
3.     if (n < 2)
4.         printf("%d 不是素数", n);
5.     else {
6.         for (i = 2; i < n; i++) {
7.             if (n%i == 0)
8.                 break;
9.         }
10.        if (i < n) {
11.            printf("%d 不是素数\n", n);
12.        }
13.        else
14.            printf("%d 是素数\n", n);
15.    }
16.    return 0;
17. }
```



环路复杂度：  $V(G) = 4(\text{判定节点数}) + 1 = 5$

独立路径	测试用例	结果
路径 1: 2,3,4,16	$n = 0$	输出"0 不是素数"
路径 2: 2,3,6,10,14,16	$n = 2$	输出"2 是素数"
路径 3: 2,3,6,7,10,11,16	$n = 4$	输出"4 不是素数"
路径 4: 2,3,6,7,6,10,11,16	$n = 6$	输出"6 不是素数"
路径 5: 2,3,6,7,6,10,14,16	$n = 7$	输出"7 是素数"

独立路径	测试用例	结果
路径 1: 1, 2……	$n = \dots$	
路径 2: …		
路径 3: …		