

2020—2021 学年第一学期

年级/班级: _____

[illegible]

```
def(i) = {s, low, high}
```

的数据流图

分，定义使用标注约 1 分/

```
graph TD; 1((1)) --> 2((2)); 2((2)) --> 3((3)); 2((2)) --> 6((6)); 3((3)) --> 4((4)); 3((3)) --> 5((5)); 4((4)) --> 6((6)); 5((5)) --> 6((6)); 6((6)) --> 7((7));
```

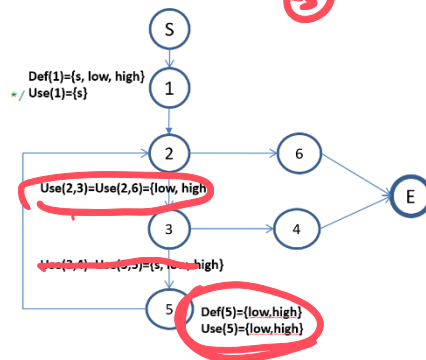
Subs D.

123523
5234

```

1  #include <stdio.h>
2  #include <string.h>
3
4  /** Check if a string is a palindrome */
5  int isPalindrome(char s[]) {
6      /* The index of the first character in the string */
7      int low = 0;
8
9      /* The index of the last character in the string */
10     int high = strlen(s) - 1;
11
12     while (low < high) {
13         if (s[low] != s[high])
14             return 0; /* Not a palindrome */
15
16         low++;
17         high--;
18     }
19
20     return 1; /* The string is a palindrome */
21 }

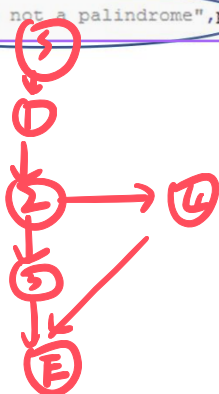
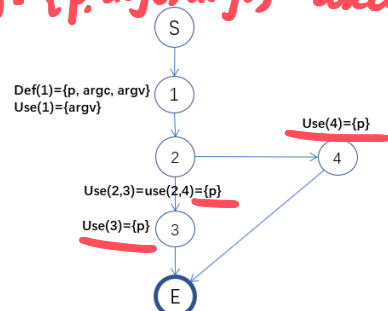
```


$$\text{Def}(u) = \{p, \arg c, \arg v\} \quad \text{use}(u) = \{\arg v\}$$

```

23      /** Main method */
24      void main(int argc, char *argv[]) {
25          char *p = argv[1];
26
27          if (isPalindrome(p))
28              printf("%s is a palindrome", p);
29          else
30              printf("%s is not a palindrome", p);
31      }
32
33

```



Ex: 1.5, 使用: (2,3)(2,6)(3,4)(3,5), 5

$du(1, (2,3), high) = \{(1,2,3)\}$

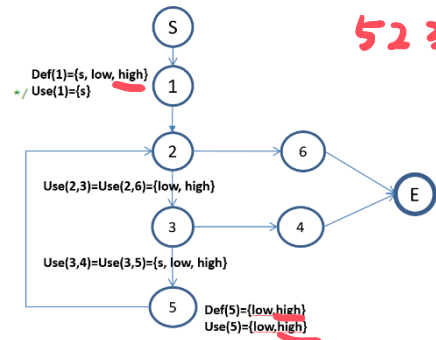
1 2 6
1 2 3 4
1 2 3 5
5 2 3
5 2 6

b) 给出变量 high 的定义使用路径集合 (du-paths) (12 分)

评分标准: 8 条定义使用路径, 1.5 分/条

high 的定义节点为 1, 5, 因此 high 的定义使用路径集合为: $du(1, high) \cup du(5, high)$

- 1) $du(1,3,high) = \{(1,2,3)\}$
- 2) $du(1,6,high) = \{(1,2,6)\}$
- 3) $du(1,4,high) = \{(1,2,3,4)\}$
- 4) $du(1,5,high) = \{(1,2,3,5)\}$
- 5) $du(5,3,high) = \{(5,2,3)\}$
- 6) $du(5,6,high) = \{(5,2,6)\}$
- 7) $du(5,4,high) = \{(5,2,3,4)\}$
- 8) $du(5,5,high) = \{(5,2,3,5)\}$



c) 给出满足变量 high 的全定义使用路径覆盖 (all du-paths) 的完全路径集合及其相应的测试集合 (16 points)

评分标准, 答案不唯一

- 1) 完全路径集合 8 分, 没有覆盖的定义使用路径, 缺 1 条扣 1 分
- 2) 测试集合 8 分, 没有覆盖的定义使用路径, 缺 1 条扣 1 分

针对 high, 满足 all-du-path 的测试需求 $TR = \{(1,2,3), (1,2,6), (1,2,3,4), (1,2,3,5), (5,2,3), (5,2,6), (5,2,3,4), (5,2,3,5)\}$

构造完全路径满足遍历满足 all-du-path 的测试需求, 在上述 TR 中, 除去 $(1,2,3), (5,2,3)$

以外, 其余路径之间互不包含, 因此, 可以构造计算测试用例的完备路径 $CP = \{(1,2,6),$

$(1,2,3,4), (1,2,3,5,2,3,5,2,3,4), (1,2,3,5,2,6)\}$, 可求得测试集合

- 1) 遍历 $(1,2,6)$ 的 Test Case 1: Test Input: s, Expect Result: 1
- 2) 遍历 $(1,2,3,4)$ 的 Test Case 2: Test Input: su, Expect Result: 0
- 3) 遍历 $(1,2,3,5,2,6)$ 的 Test Case 3: Test Input: ss, Expect Result: 1
- 4) 遍历 $(1,2,3,5,2,3,5,2,3,4)$ 的 Test Case 4: Test Input: suss, Expect Result: 0

- d) 若使用 tempPalResult 和 tempMain 分别表示 isPalindrome 的返回临时变量和 main 中返回结果临时变量，请给出给出 main 和 isPalindrome 的耦合定义使用对（9 分）

评分标准：每个耦合定义使用对 3 分

- 1) (main(), p, 1) - (isPalindrome, s, 1)
- 2) (isPalindrome, tempPalResult, 6) - (main(), tempMain, 2)
- 3) (isPalindrome, tempPalResult, 4) - (main(), tempMain, 2)

$TC_2: \{1, 6\} \quad sum = 7$

- e) 给出满足全耦合定义使用路径覆盖的测试集合。（6 points）

评分标准：每个耦合定义使用对 3 分

$TC_1: \quad array = [] \quad sum = 7$

- 1、Test case 1: Test Input: ab, Expected Result: ab is not a palindrome
- 2、Test case 2: Test Input: aa, Expected Result: aa is a palindrome

2. FindNumbersWithSum 在降序排列的

链表数组中查询与其和与 sum 相同的两个元素，其源代码如右图所示。

表 1 是其 3 个一阶变异体，表 2 是 FindNumbersWithSum 某个待判定质量的测试集合。

```

2 vector<int> FindNumbersWithSum(vector<int> array, int sum)
3 {
4     int length = array.size();
5     vector<int> ret;
6     if (length < 2)
7         return ret;
8     int i = 0;
9     int j = length - 1;
10    while (i < j)
11    {
12        int asum = array[i] + array[j];
13        if (asum == sum)
14        {
15            ret.push_back(array[i]);
16            ret.push_back(array[j]);
17            break;
18        }
19        else if (asum < sum)
20            i++;
21        else
22            j--;
23    }
24    return ret;
25 }

```

	TC_1	TC_2	D
P	{}	{1, 6}	
P_1	{}	{}	P_1
P_2	{}	{}	P_1, P_2
P_3	{}	X	P_1, P_2, P_3

- a) 请计算该测试集合相对于表 1 的变异体集合的变异得分并给出计算过程。（22 分）

$\{2, 4\} \quad 8 \quad 3 \quad \frac{3}{3} = 1$

- b) 请构造一个一阶非等价变异体，该变异体无法被表 2 中的测试集合杀死（10 分）
- c) 针对 b) 中构造的变异体，补充能杀死该变异体的测试用例（10 分）

表格 1 FindNumbersWithSum 的变异体

Mutant ID	Mutated Position	Mutated code (original code → mutated code)
P_1	Line 6	if (length < 2) → if (length <= 2)
P_2	Line 8	int i = 0; → int i = 1;
P_3	Line 9	int j = length - 1; → int j = length;

表格 2 测试集合 1

Test Case ID	Test Case
TC_1	Test Inputs: array = {}, sum = 7; Excepted Result: ret = {}

TC ₂	Test Inputs:array={1,6},sum = 7;Expected Result: ret={1, 6}
-----------------	---

1) 评分标准: 每行 4 分, 变异得分 6 分

array = {}, sum = 7
→ {1, 6}, sum = 7

	Test Set		
Program	TC1	TC2	D
P	{}	{1,6}	
P1	{}	null	p1
P2	{}	null	p1,p2
P3	{}	出错	p1,p2,p3

$$\frac{|D|}{(|L| + |D|)}$$

$$MS = |D|/(|L|+|D|) = 3/(0+3) = 1$$

2) 满足题意得变异体需要在 TC1 和 TC2 下输出与原代码相同的结果, 且语义不相同, 例如

`while (i<=j)`

3) 针对 `while (i<=j)`, 补充测试用例 TC3: array = {1, 2, 6}, sum=12, expect result: null

(在变异体下, 返回结果为{6, 6})