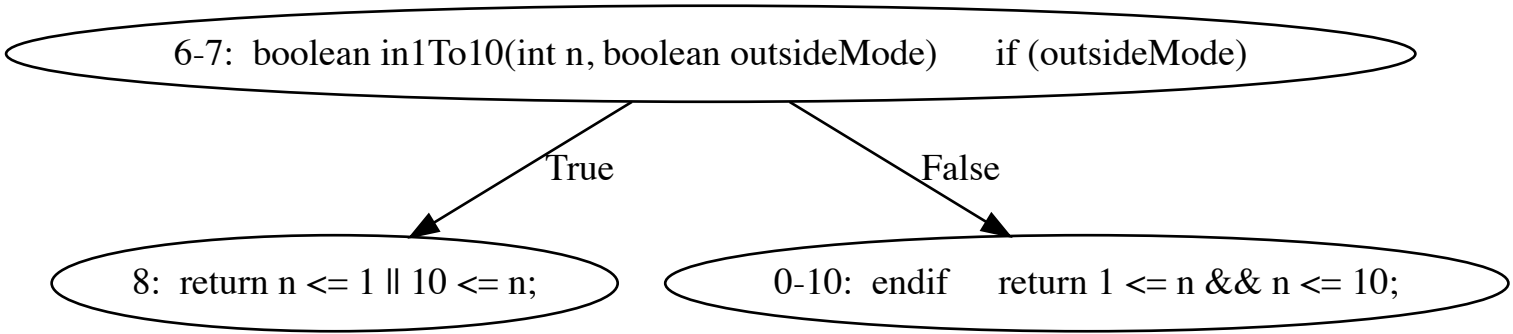
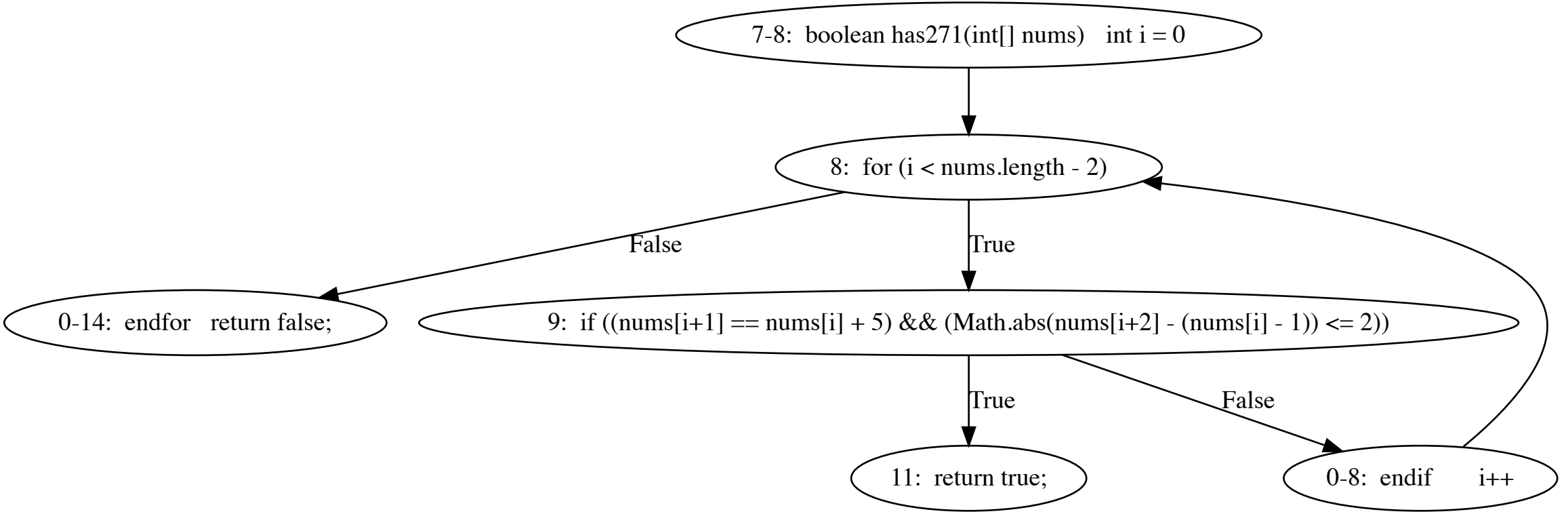
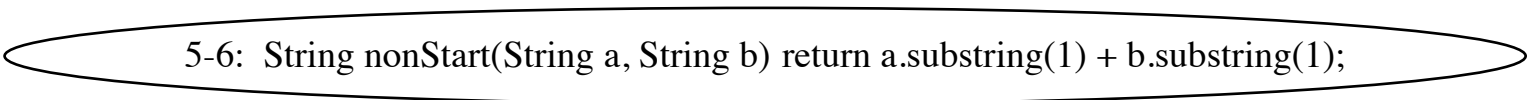
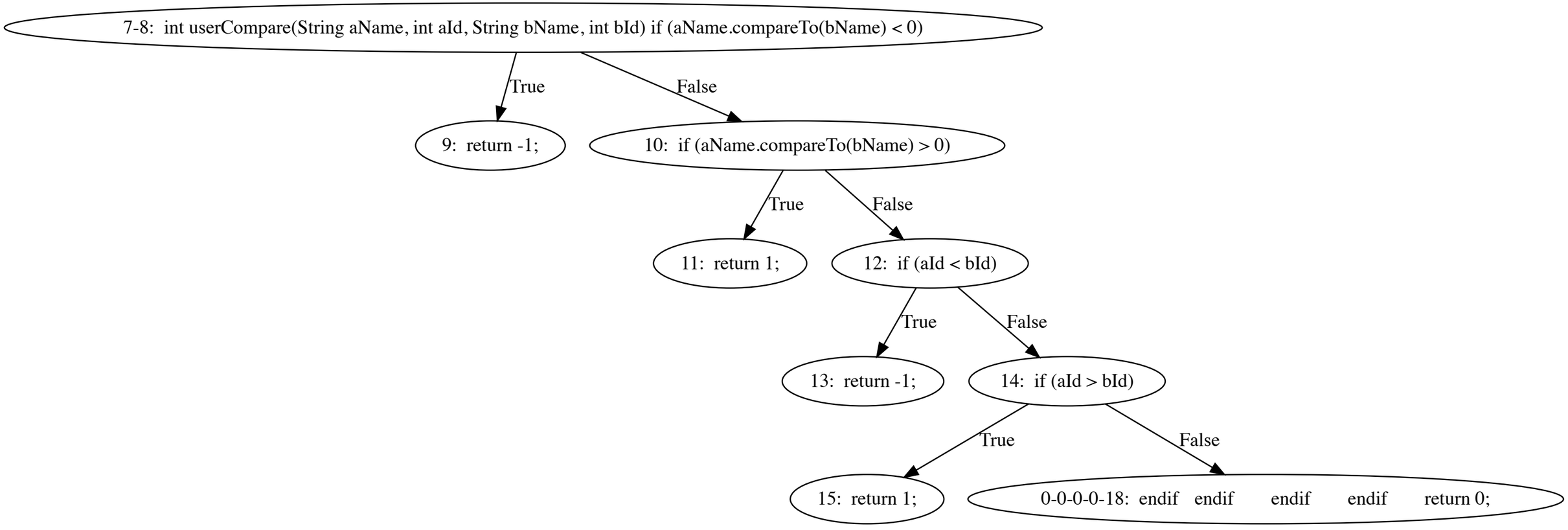
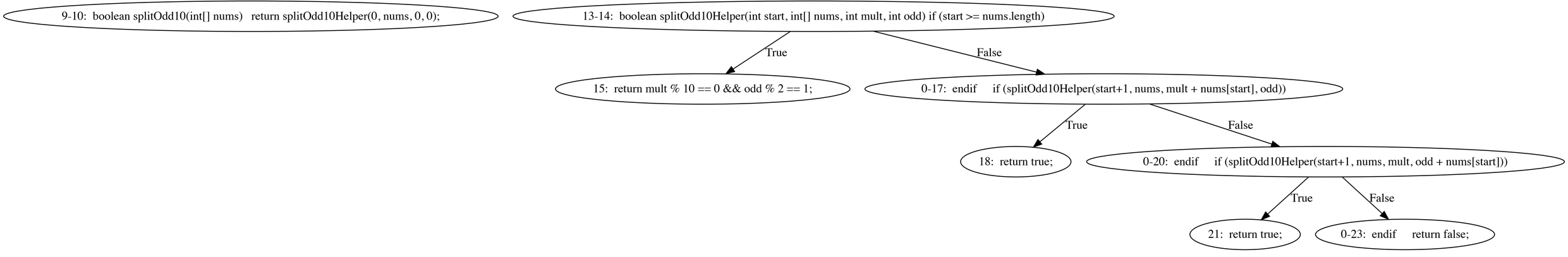
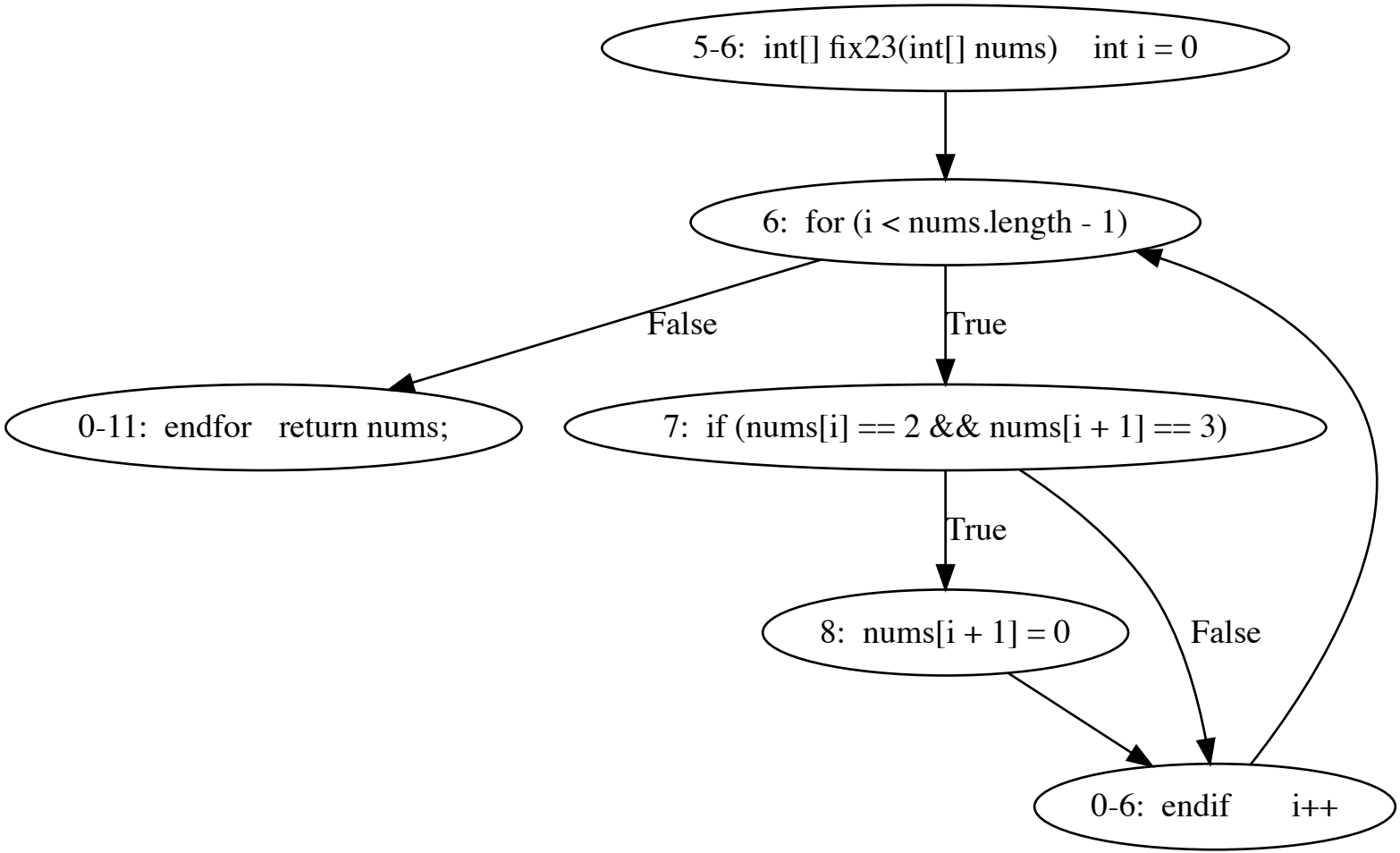

6-7: boolean in1To10(int n, boolean outsideMode) if (outsideMode)
True
False
8: return n <= 1 || 10 <= n;
0-10: endif return 1 <= n && n <= 10;

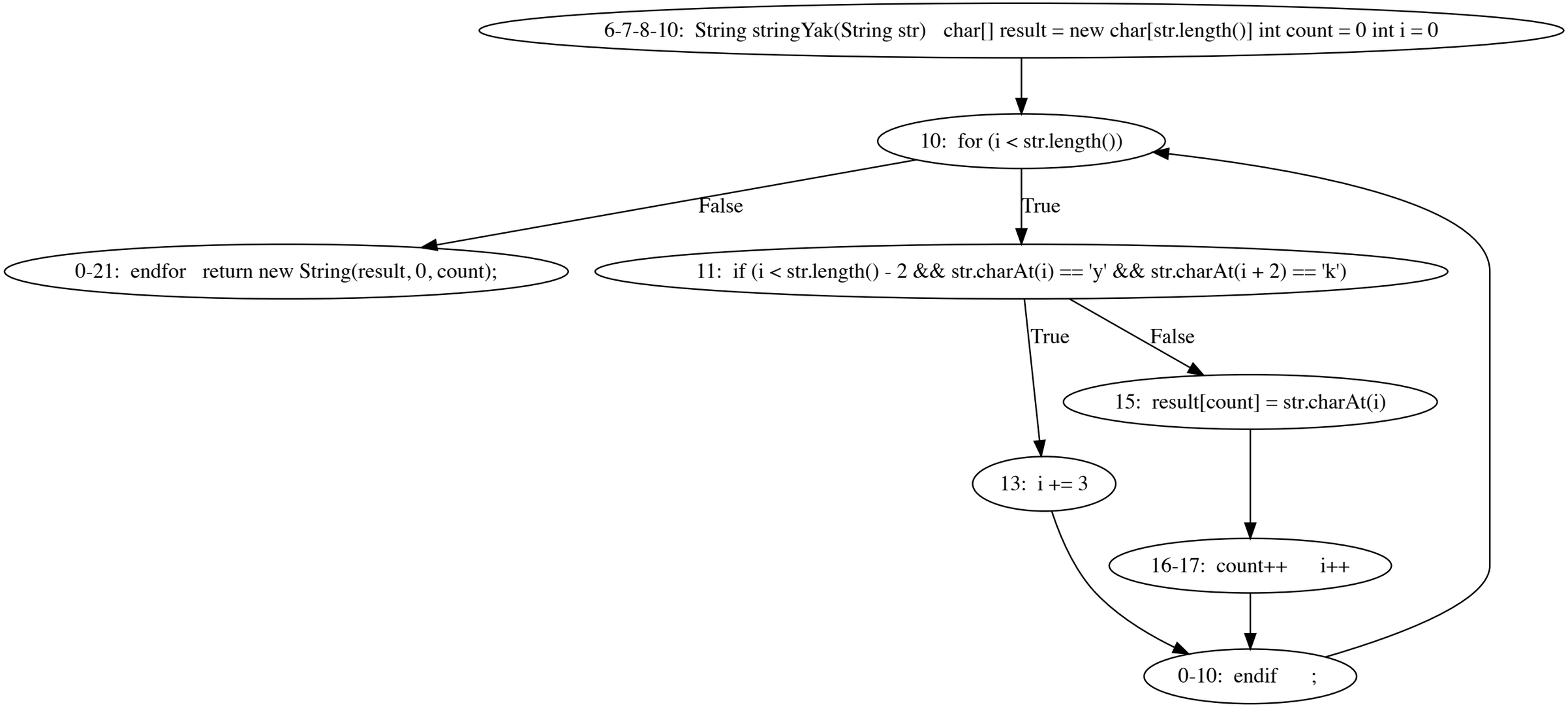

7-8: boolean has271(int[] nums) int i = 0
8: for (i < nums.length - 2)
False
True
0-14: endfor return false;
9: if ((nums[i+1] == nums[i] + 5) && (Math.abs(nums[i+2] - (nums[i] - 1)) <= 2))
True
False
11: return true;
0-8: endif i++

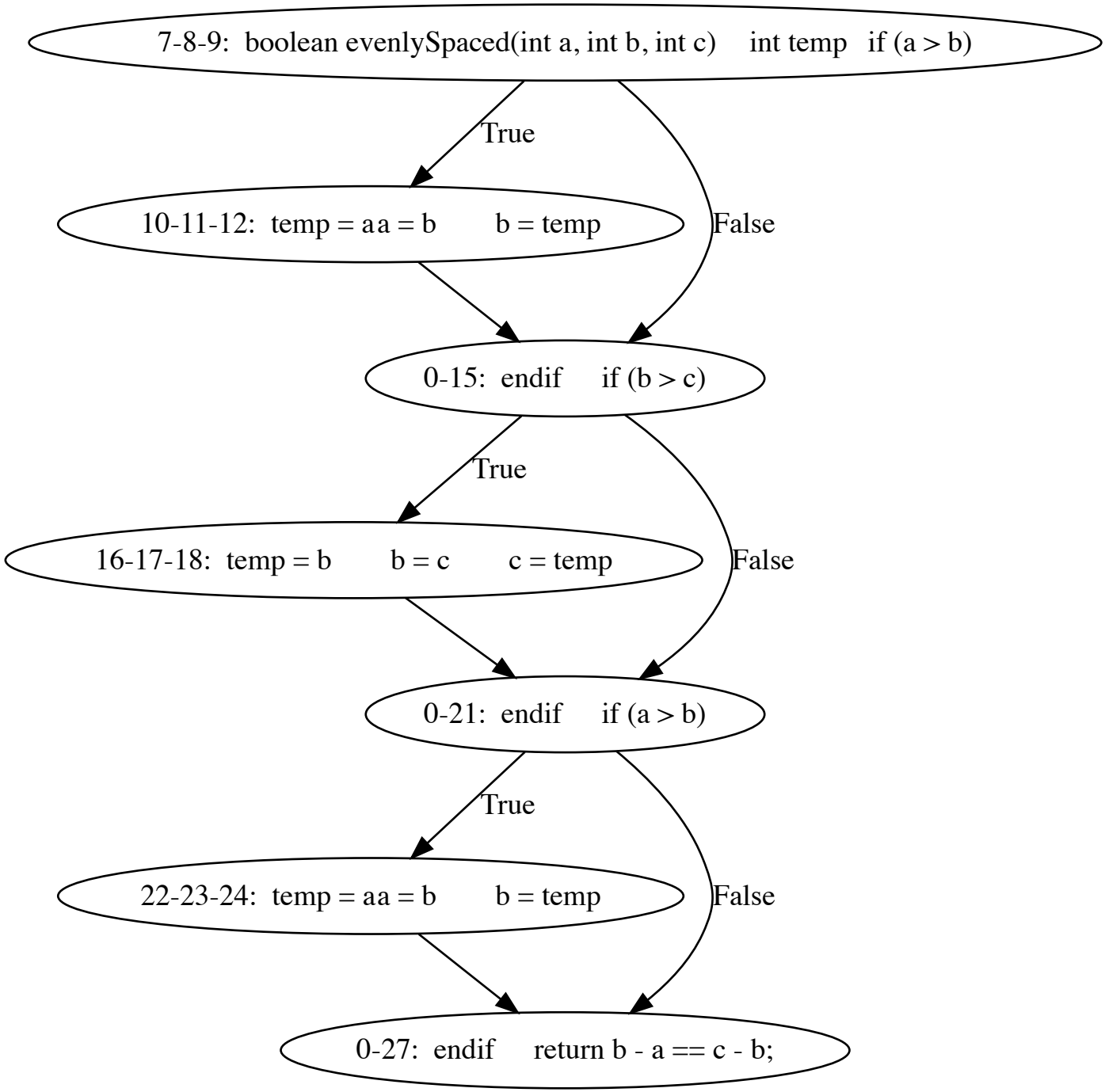

5-6: String nonStart(String a, String b) return a.substring(1) + b.substring(1);

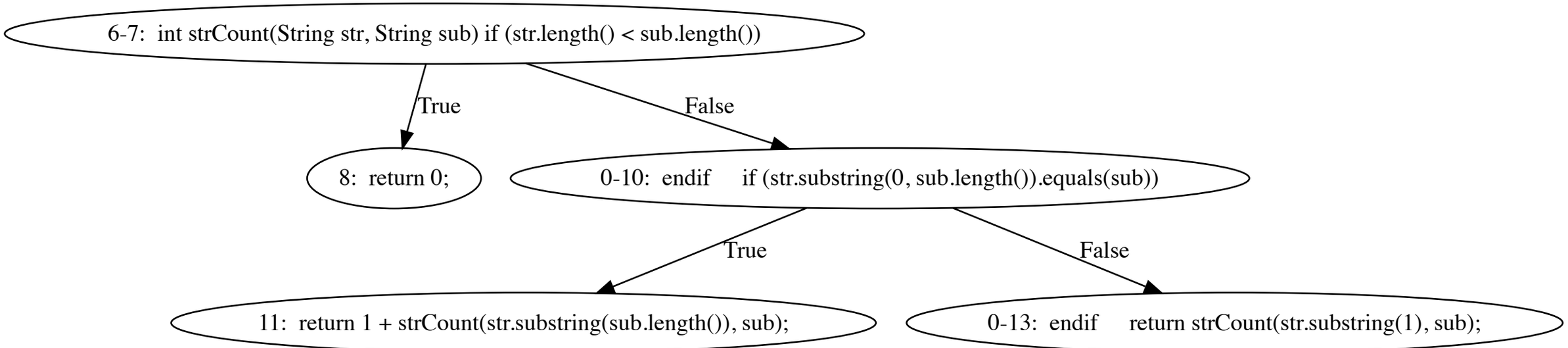

7-8: int userCompare(String aName, int aId, String bName, int bId) if (aName.compareTo(bName) < 0)
True
False
9: return -1;
10: if (aName.compareTo(bName) > 0)
True
False
11: return 1;
12: if (aId < bId)
True
False
13: return -1;
14: if (aId > bId)
True
False
15: return 1;
0-0-0-0-18: endif endif endif endif return 0;

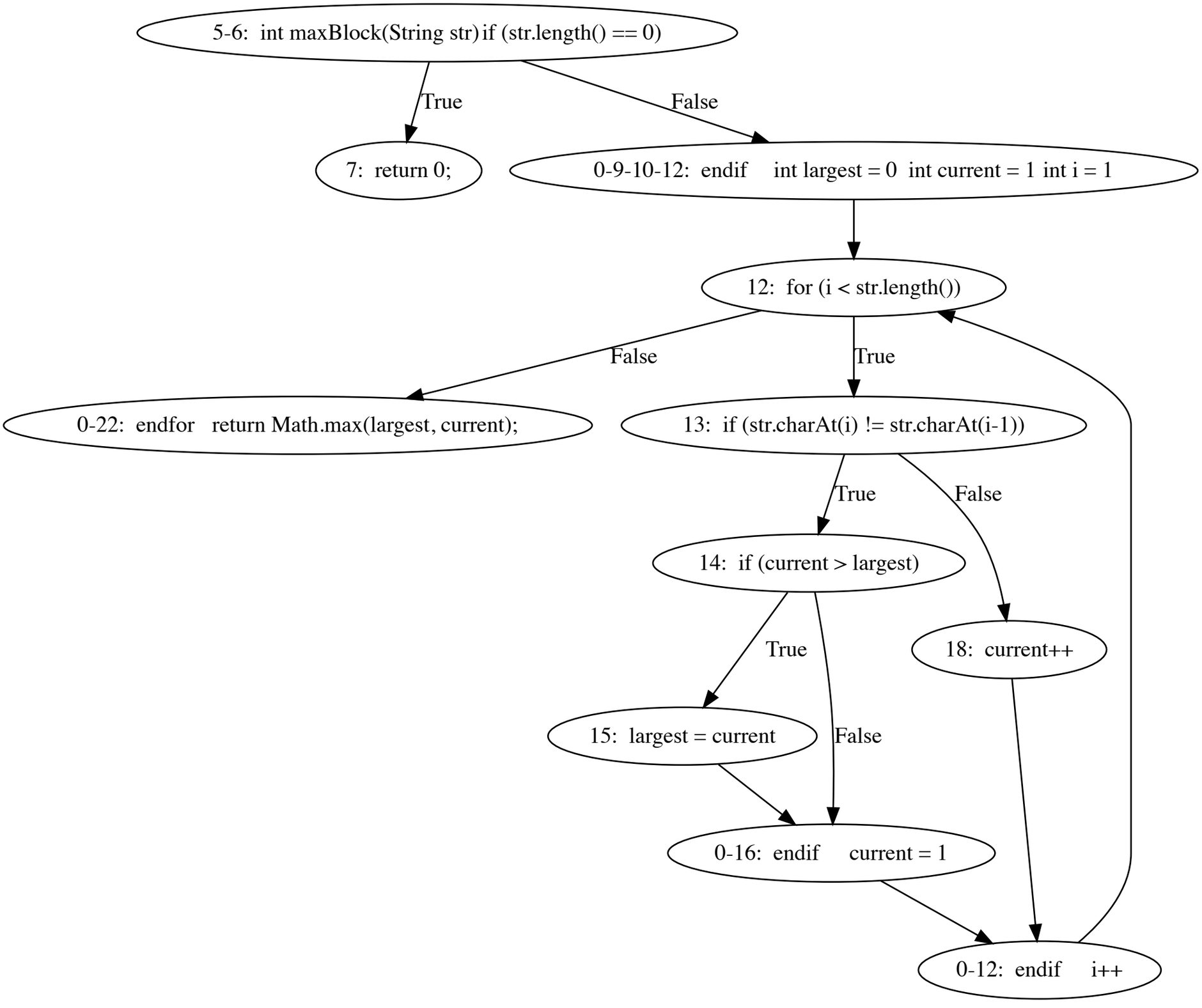

9-10: boolean splitOdd10(int[] nums) return splitOdd10Helper(0, nums, 0, 0);
13-14: boolean splitOdd10Helper(int start, int[] nums, int mult, int odd) if (start >= nums.length)
True
False
15: return mult % 10 == 0 && odd % 2 == 1;
0-17: endif if (splitOdd10Helper(start+1, nums, mult + nums[start], odd))
True
False
18: return true;
0-20: endif if (splitOdd10Helper(start+1, nums, mult, odd + nums[start]))
True
False
21: return true;
0-23: endif return false;

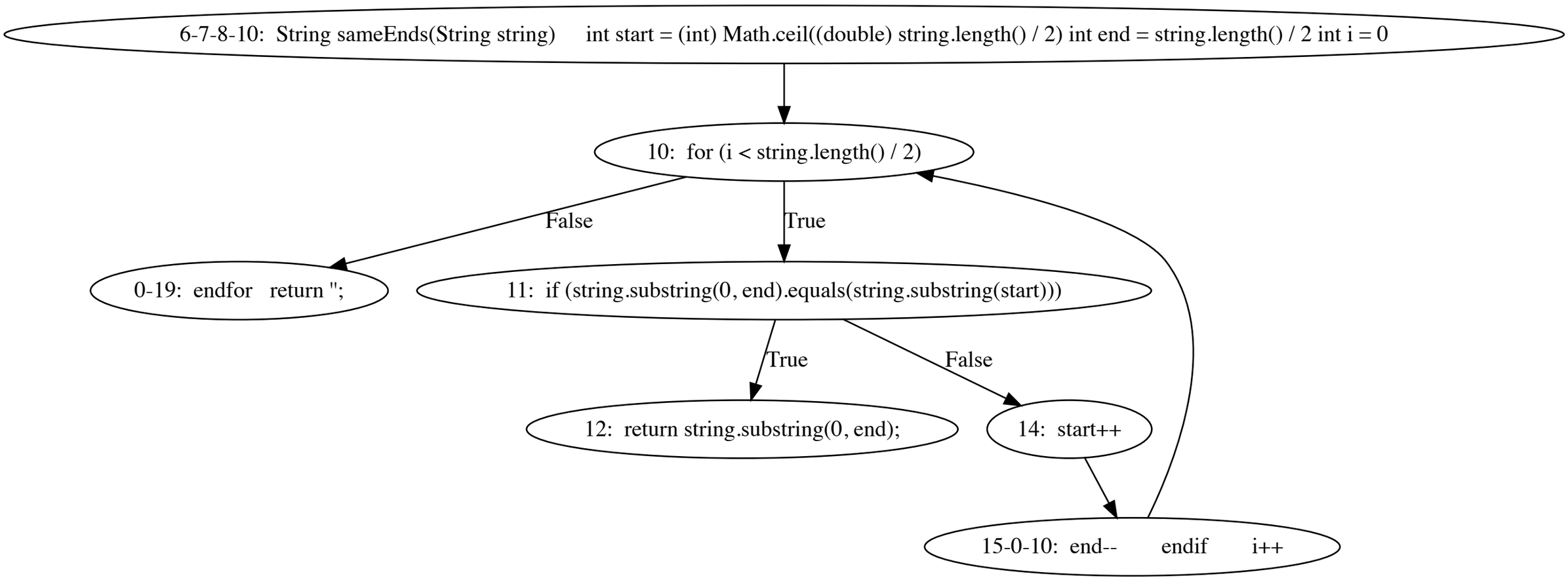

5-6: int[] fix23(int[] nums) int i = 0
6: for (i < nums.length - 1)
False
True
0-11: endfor return nums;
7: if (nums[i] == 2 && nums[i + 1] == 3)
True
8: nums[i + 1] = 0
False
0-6: endif i++

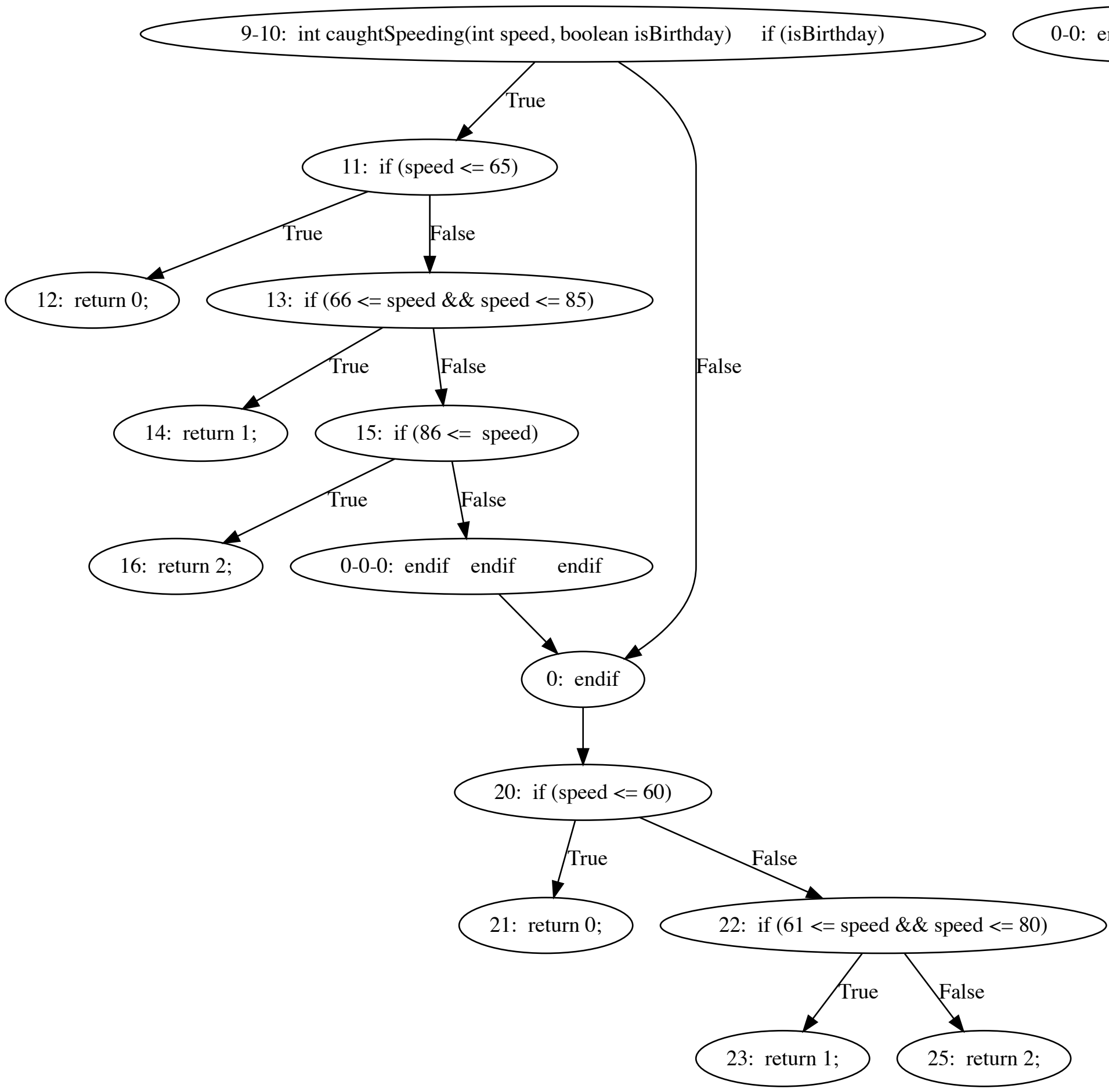

6-7-8-10: String stringYak(String str) char[] result = new char[str.length()] int count = 0 int i = 0
10: for (i < str.length())
False
True
0-21: endfor return new String(result, 0, count);
11: if (i < str.length() - 2 && str.charAt(i) == 'y' && str.charAt(i + 2) == 'k')
True
False
15: result[count] = str.charAt(i)
13: i += 3
16-17: count++ i++
0-10: endif ;

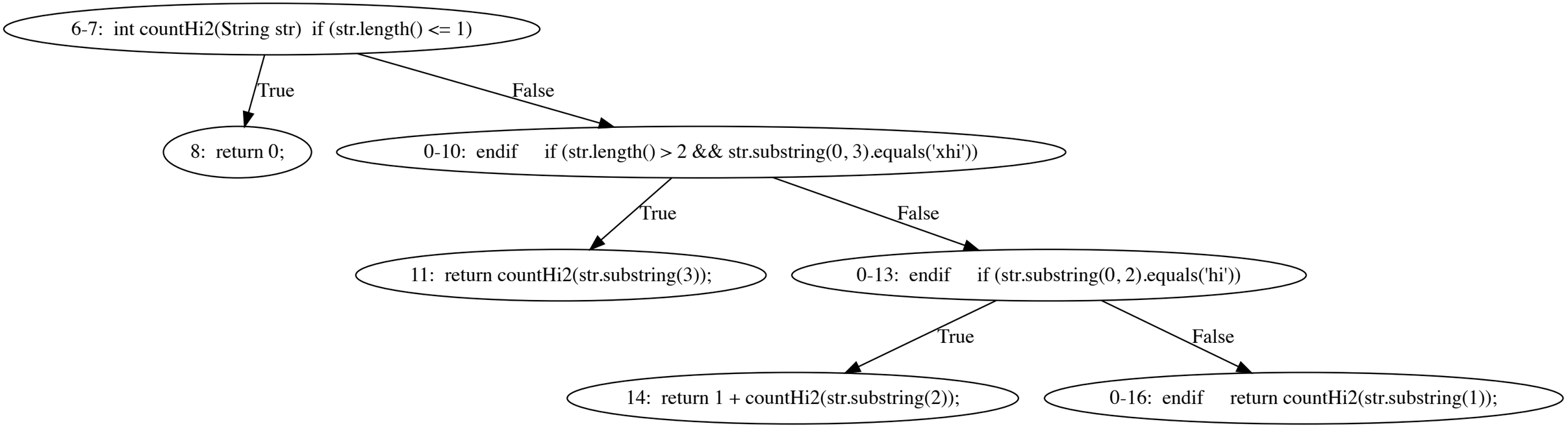

7-8-9: boolean evenlySpaced(int a, int b, int c) int temp if (a > b)
True
False
10-11-12: temp = aa = b b = temp
0-15: endif if (b > c)
True
False
16-17-18: temp = b b = c c = temp
0-21: endif if (a > b)
True
False
22-23-24: temp = aa = b b = temp
0-27: endif return b - a == c - b;

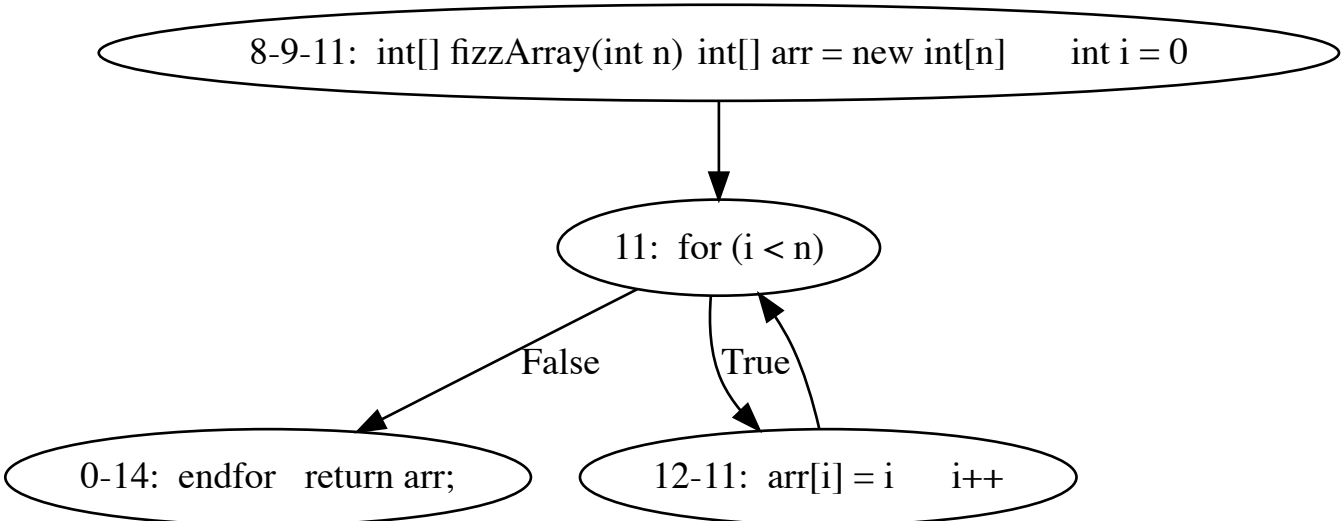

6-7: int strCount(String str, String sub) if (str.length() < sub.length())
True
False
8: return 0;
0-10: endif if (str.substring(0, sub.length()).equals(sub))
True
False
11: return 1 + strCount(str.substring(sub.length()), sub);
0-13: endif return strCount(str.substring(1), sub);

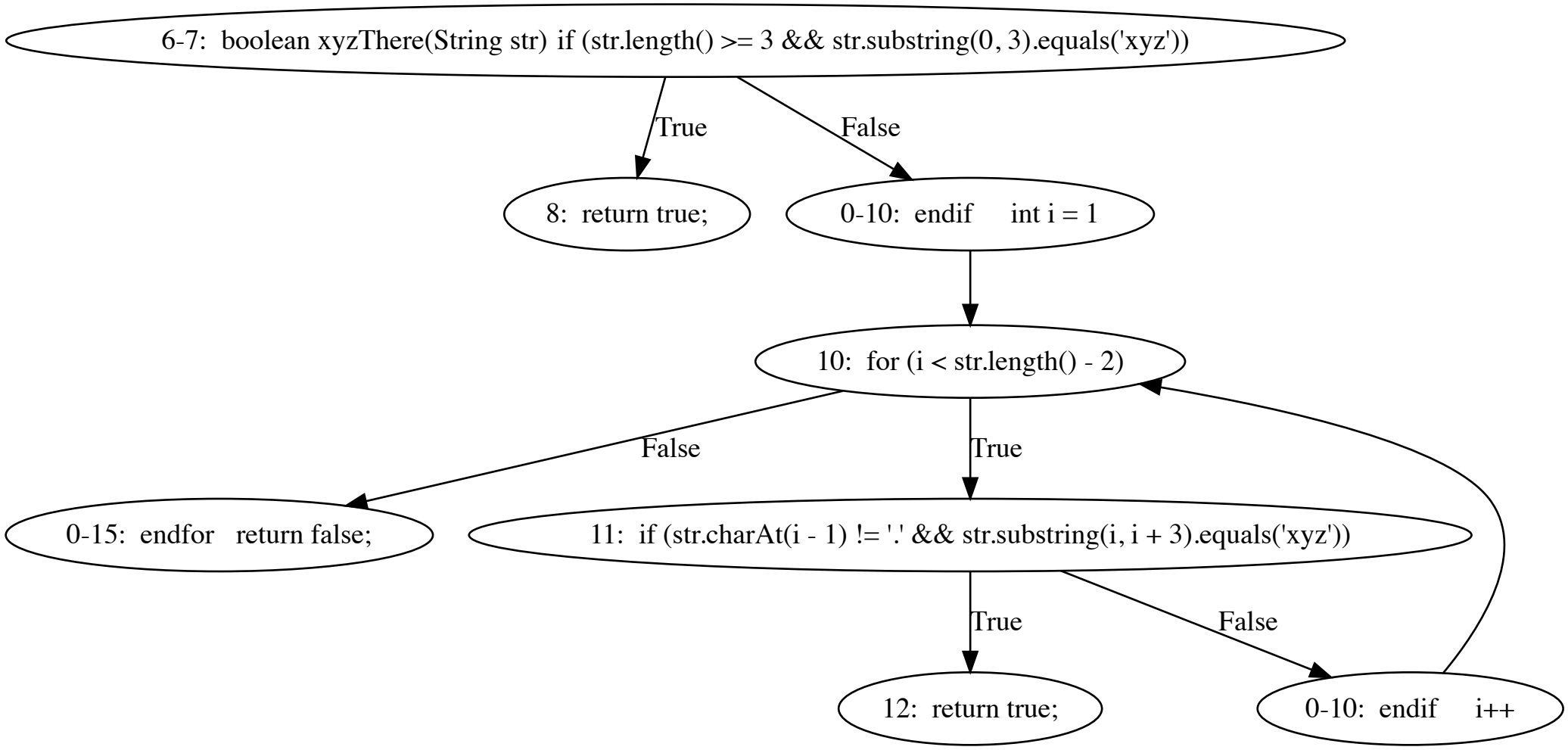

5-6: int maxBlock(String str) if (str.length() == 0)
True
False
7: return 0;
0-9-10-12: endif int largest = 0 int current = 1 int i = 1
12: for (i < str.length())
False
True
0-22: endfor return Math.max(largest, current);
13: if (str.charAt(i) != str.charAt(i-1))
True
False
14: if (current > largest)
18: current++
True
False
15: largest = current
0-16: endif current = 1
0-12: endif i++

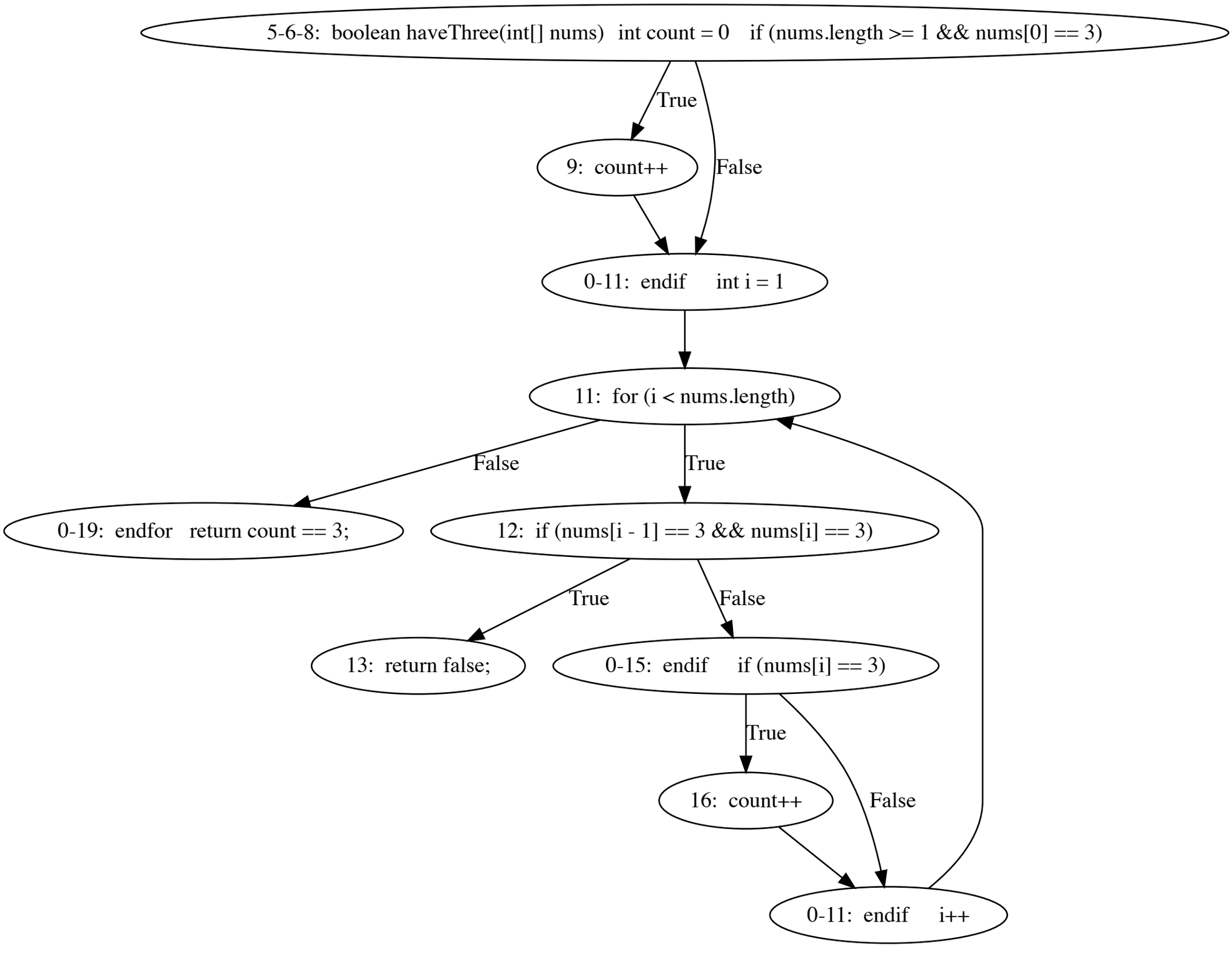

6-7-8-10: String sameEnds(String string) int start = (int) Math.ceil((double) string.length() / 2) int end = string.length() / 2 int i = 0
10: for (i < string.length() / 2)
False
True
0-19: endfor return '';
11: if (string.substring(0, end).equals(string.substring(start)))
True
False
12: return string.substring(0, end);
14: start++
15-0-10: end-- endif i++

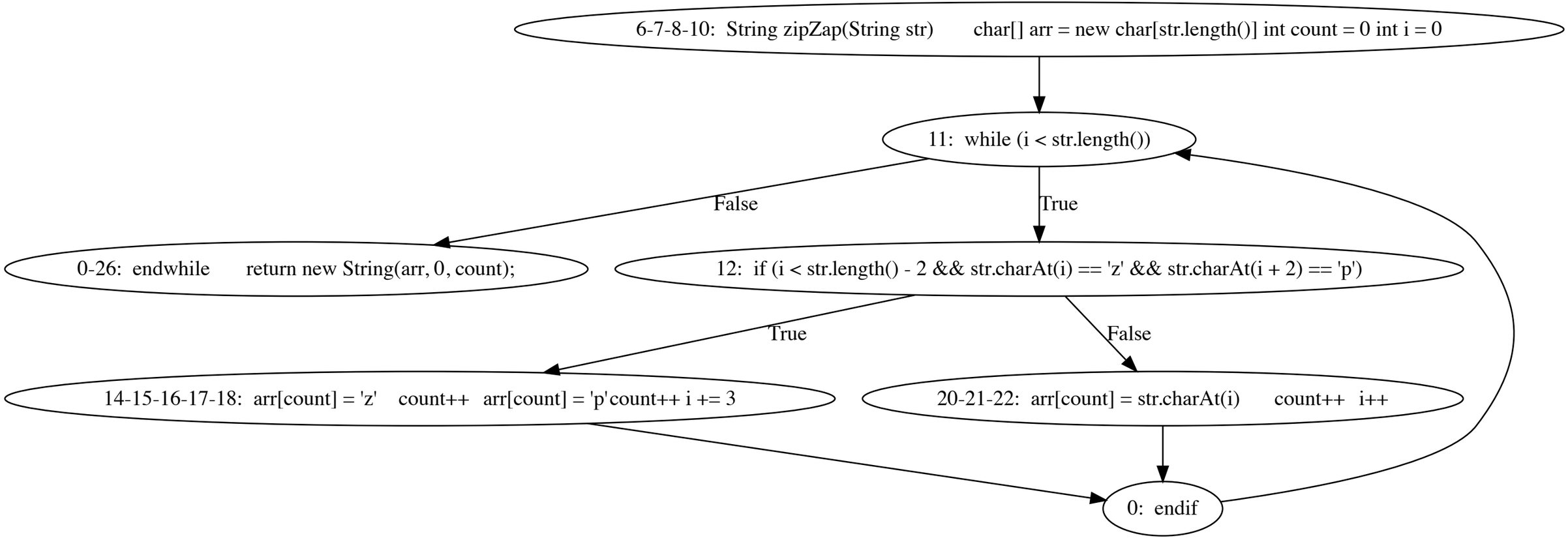

9-10: int caughtSpeeding(int speed, boolean isBirthday) if (isBirthday)
0-0: endif endif
True
11: if (speed <= 65)
True
False
12: return 0;
13: if (66 <= speed && speed <= 85)
True
False
False
14: return 1;
15: if (86 <= speed)
True
False
16: return 2;
0-0-0: endif endif endif
0: endif
20: if (speed <= 60)
True
False
21: return 0;
22: if (61 <= speed && speed <= 80)
True
False
23: return 1;
25: return 2;

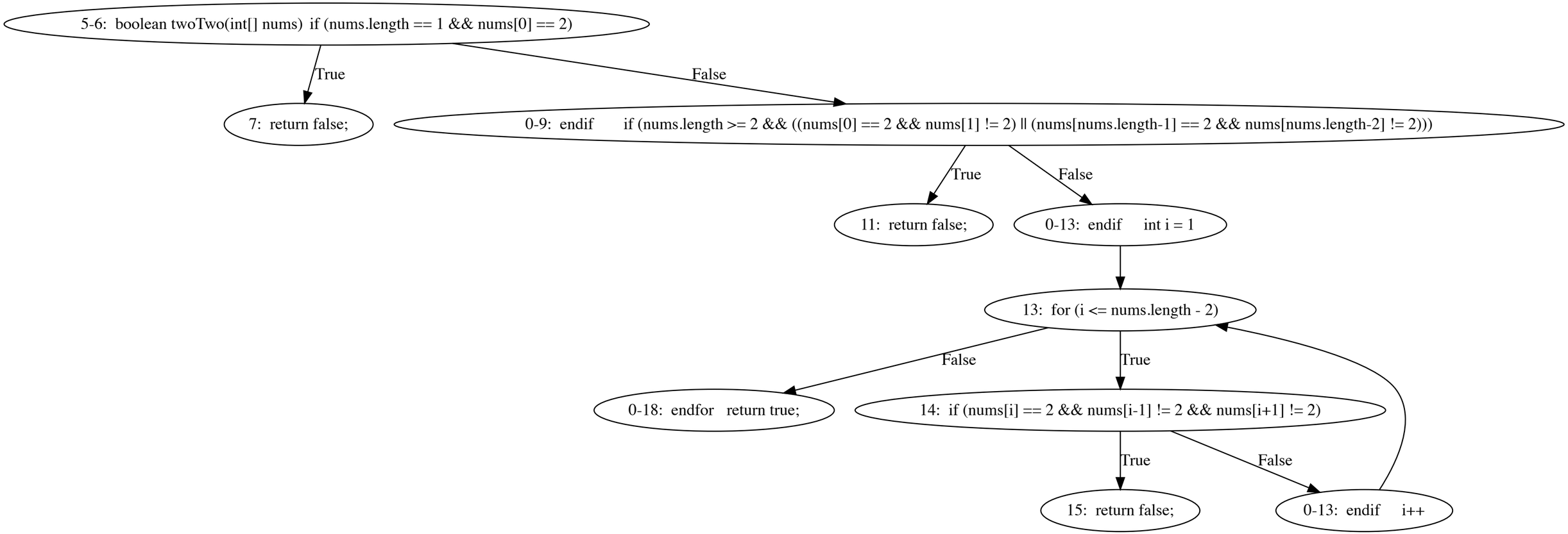

6-7: int countHi2(String str) if (str.length() <= 1)
True
False
8: return 0;
0-10: endif if (str.length() > 2 && str.substring(0, 3).equals('xhi'))
True
False
11: return countHi2(str.substring(3));
0-13: endif if (str.substring(0, 2).equals('hi'))
True
False
14: return 1 + countHi2(str.substring(2));
0-16: endif return countHi2(str.substring(1));

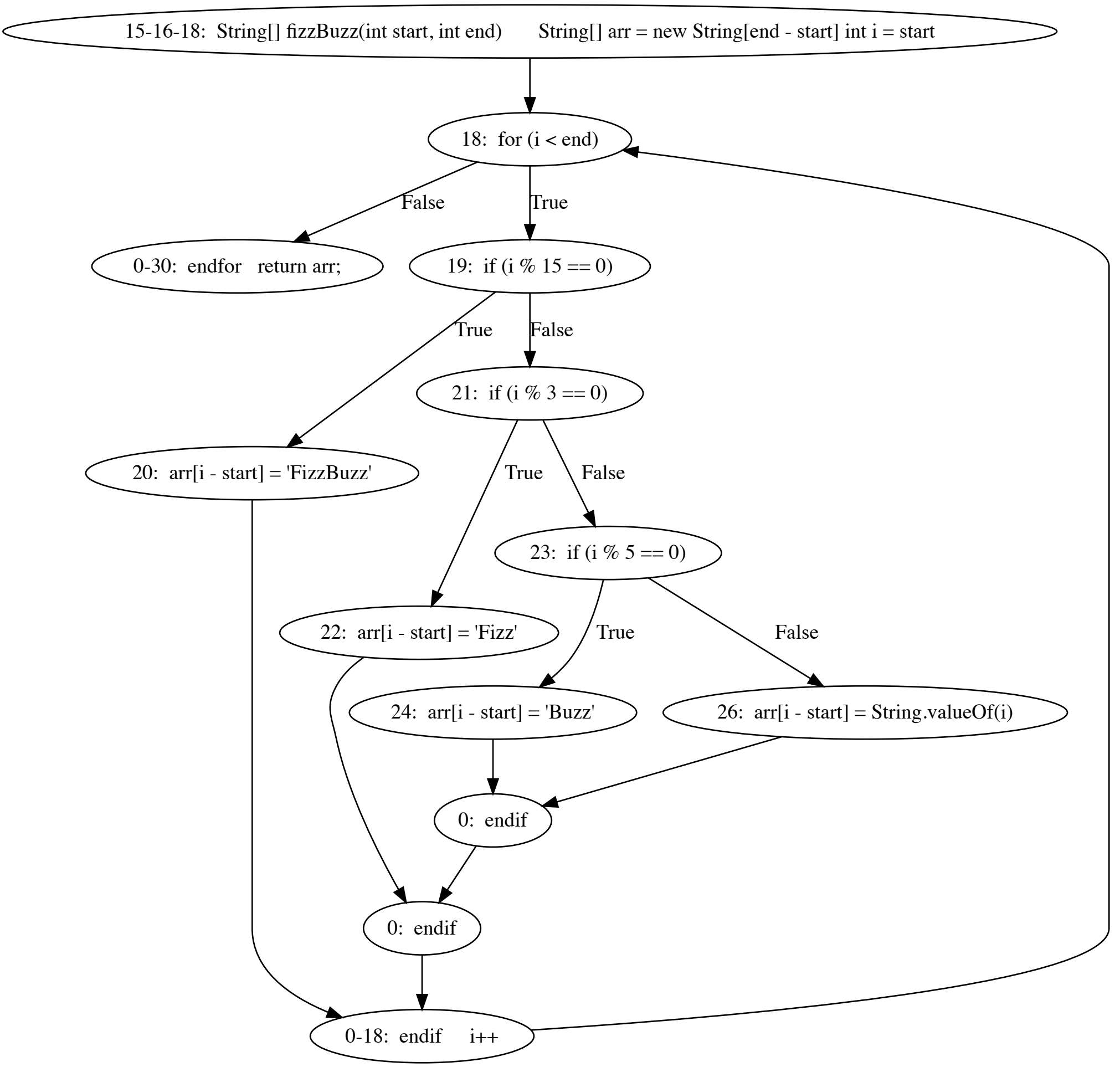

8-9-11: int[] fizzArray(int n) int[] arr = new int[n] int i = 0
11: for (i < n)
False
True
0-14: endfor return arr;
12-11: arr[i] = i i++

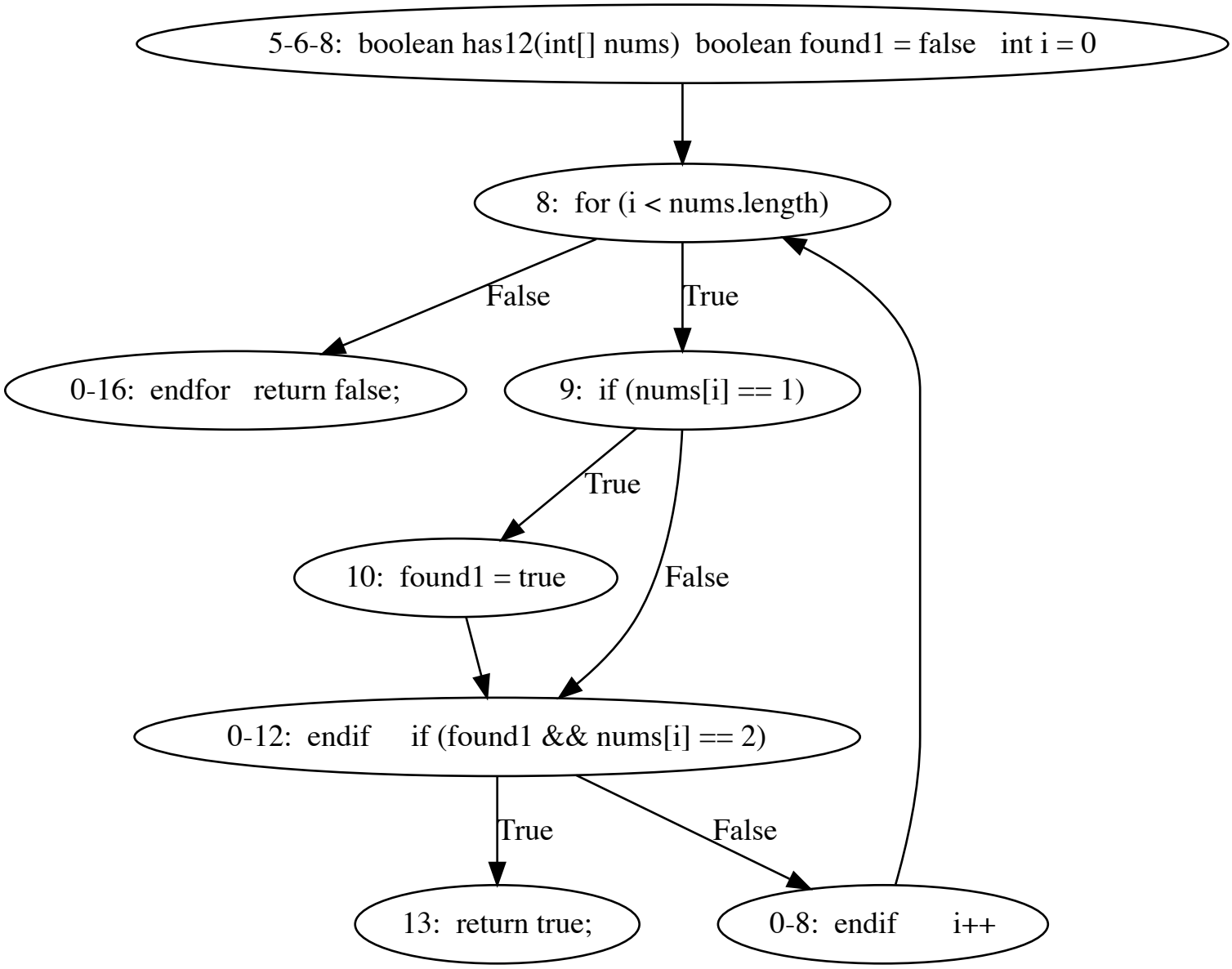

6-7: boolean xyzThere(String str) if (str.length() >= 3 && str.substring(0, 3).equals('xyz'))
True
False
8: return true;
0-10: endif int i = 1
10: for (i < str.length() - 2)
False
True
0-15: endfor return false;
11: if (str.charAt(i - 1) != '.' && str.substring(i, i + 3).equals('xyz'))
True
False
12: return true;
0-10: endif i++

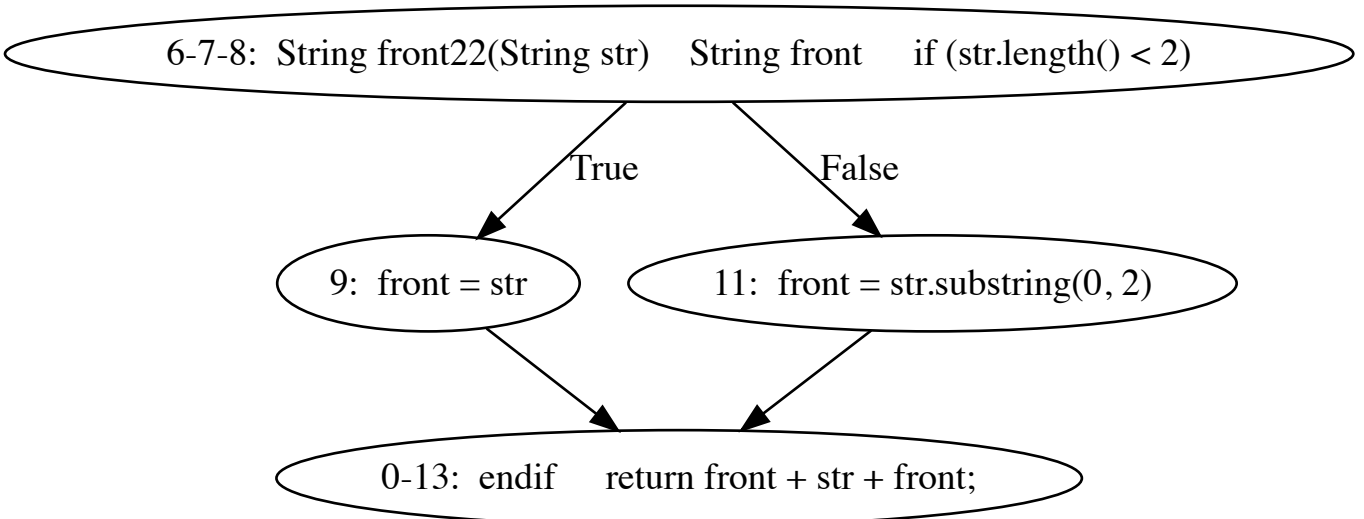

5-6-8: boolean haveThree(int[] nums) int count = 0 if (nums.length >= 1 && nums[0] == 3)
True
False
9: count++
0-11: endif int i = 1
11: for (i < nums.length)
False
True
0-19: endfor return count == 3;
12: if (nums[i - 1] == 3 && nums[i] == 3)
True
False
13: return false;
0-15: endif if (nums[i] == 3)
True
False
16: count++
0-11: endif i++

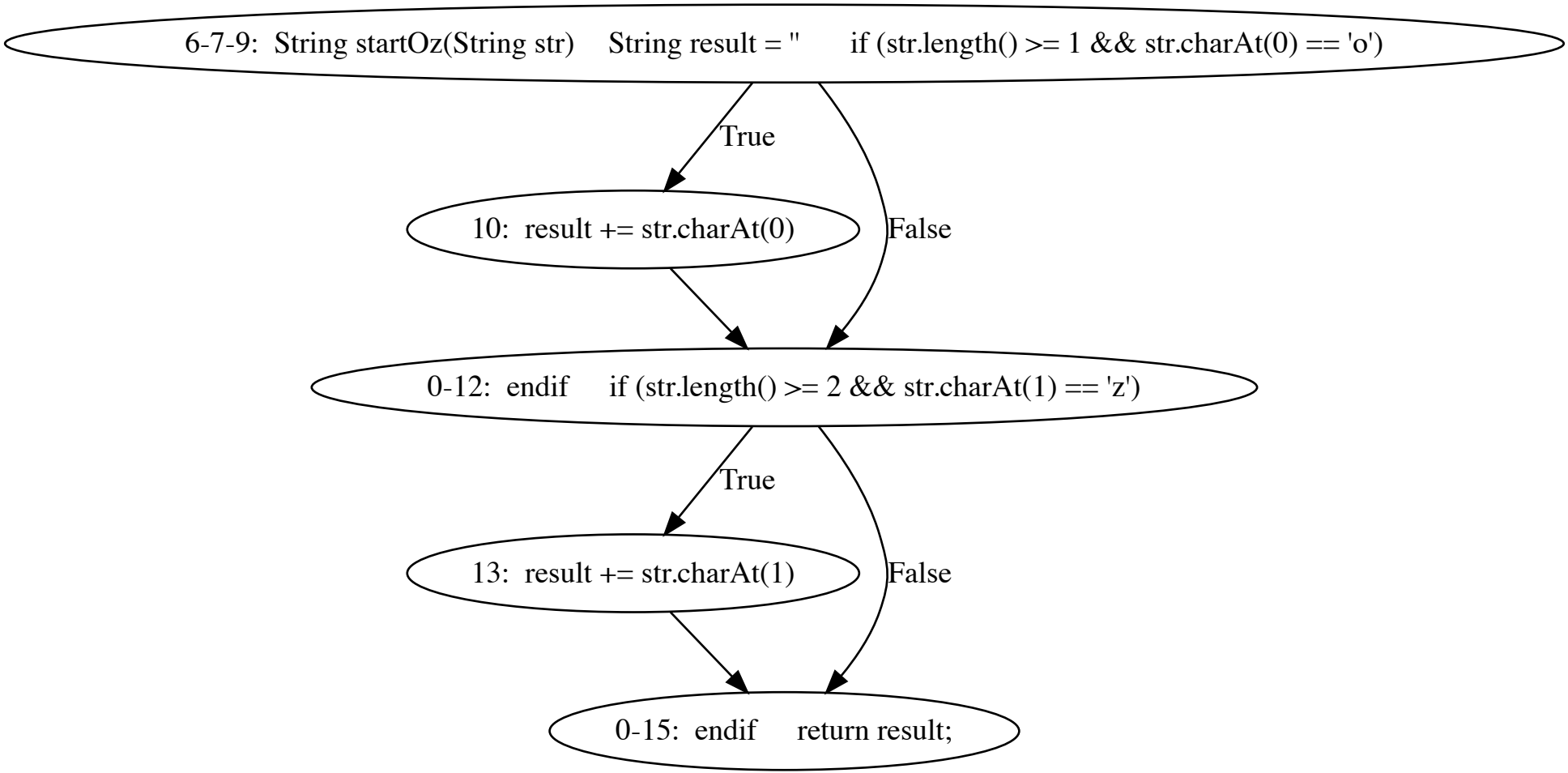

6-7-8-10: String zipZap(String str) char[] arr = new char[str.length()] int count = 0 int i = 0
11: while (i < str.length())
False
True
0-26: endwhile return new String(arr, 0, count);
12: if (i < str.length() - 2 && str.charAt(i) == 'z' && str.charAt(i + 2) == 'p')
True
False
14-15-16-17-18: arr[count] = 'z' count++ arr[count] = 'p'count++ i += 3
20-21-22: arr[count] = str.charAt(i) count++ i++
0: endif

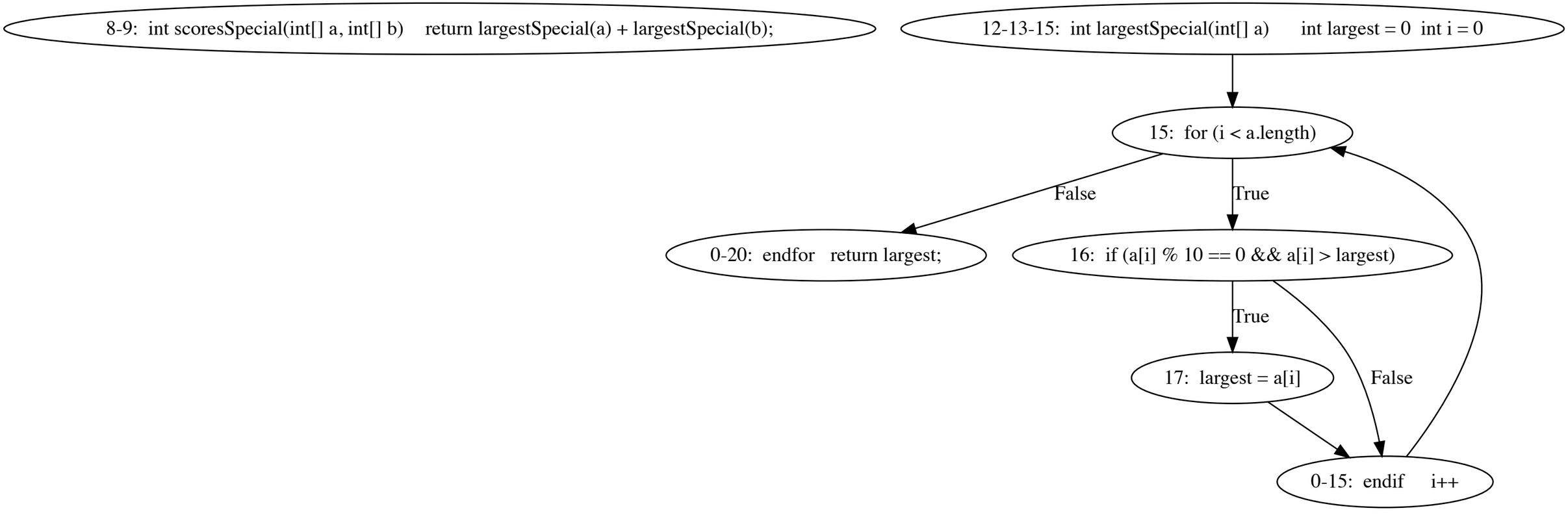

5-6: boolean twoTwo(int[] nums) if (nums.length == 1 && nums[0] == 2)
True
False
7: return false;
0-9: endif if (nums.length >= 2 && ((nums[0] == 2 && nums[1] != 2) || (nums[nums.length-1] == 2 && nums[nums.length-2] != 2)))
True
False
11: return false;
0-13: endif int i = 1
13: for (i <= nums.length - 2)
False
True
0-18: endfor return true;
14: if (nums[i] == 2 && nums[i-1] != 2 && nums[i+1] != 2)
True
False
15: return false;
0-13: endif i++

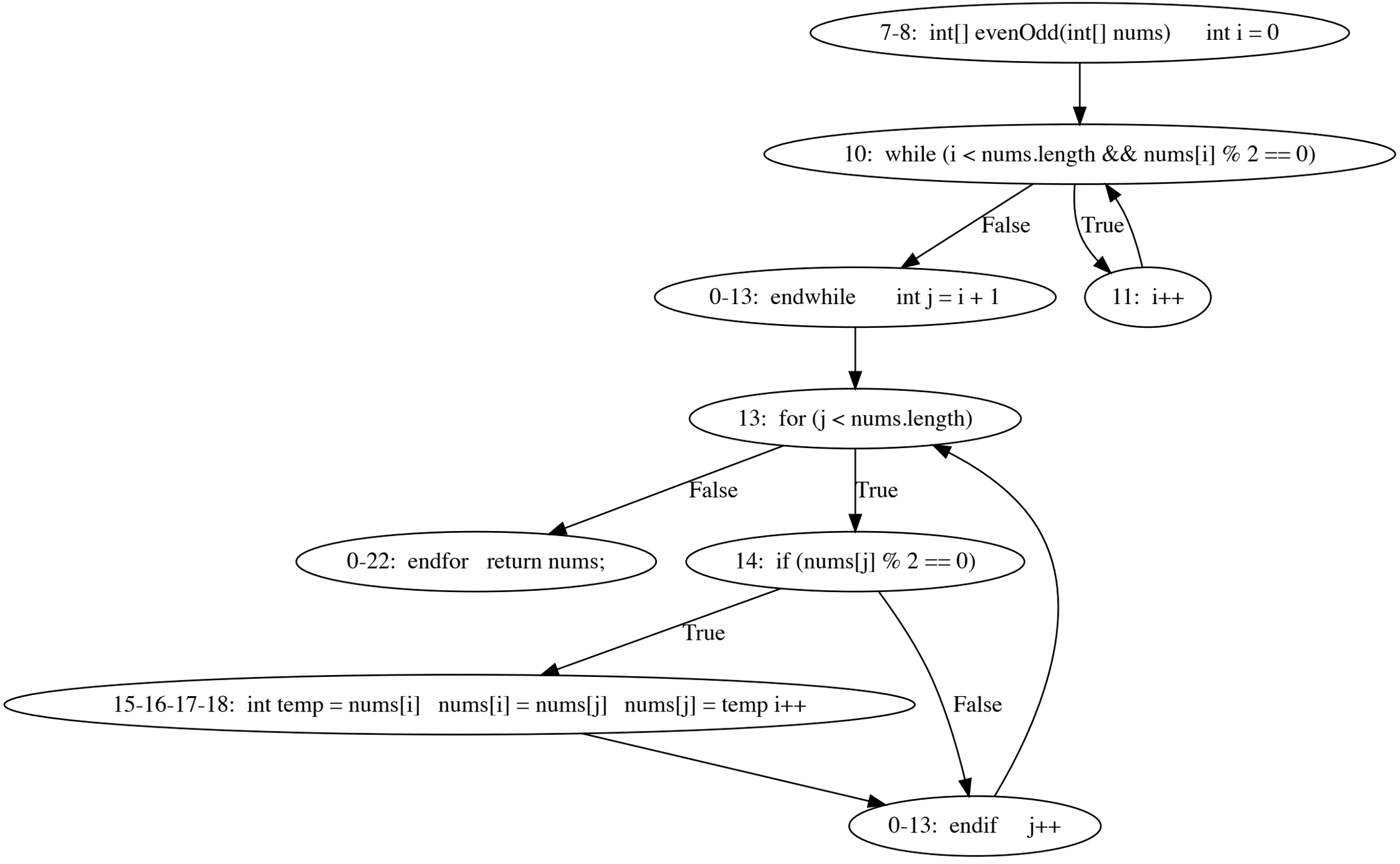

15-16-18: String[] fizzBuzz(int start, int end) String[] arr = new String[end - start] int i = start
18: for (i < end)
False
True
0-30: endfor return arr;
19: if (i % 15 == 0)
True
False
21: if (i % 3 == 0)
20: arr[i - start] = 'FizzBuzz'
True
False
23: if (i % 5 == 0)
22: arr[i - start] = 'Fizz'
True
False
24: arr[i - start] = 'Buzz'
26: arr[i - start] = String.valueOf(i)
0: endif
0: endif
0-18: endif i++

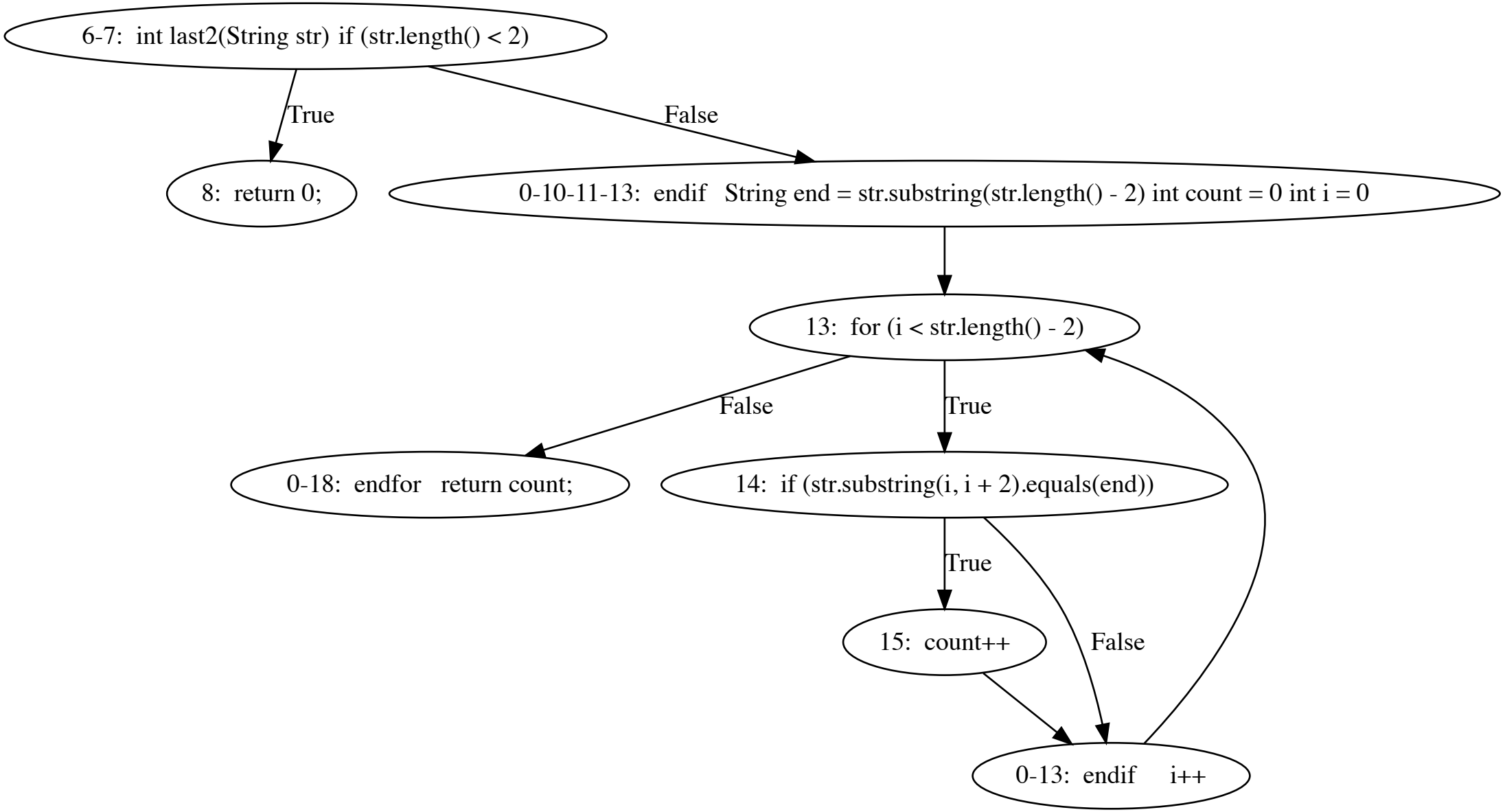

5-6-8: boolean has12(int[] nums) boolean found1 = false int i = 0
8: for (i < nums.length)
False
True
0-16: endfor return false;
9: if (nums[i] == 1)
True
False
10: found1 = true
0-12: endif if (found1 && nums[i] == 2)
True
False
13: return true;
0-8: endif i++

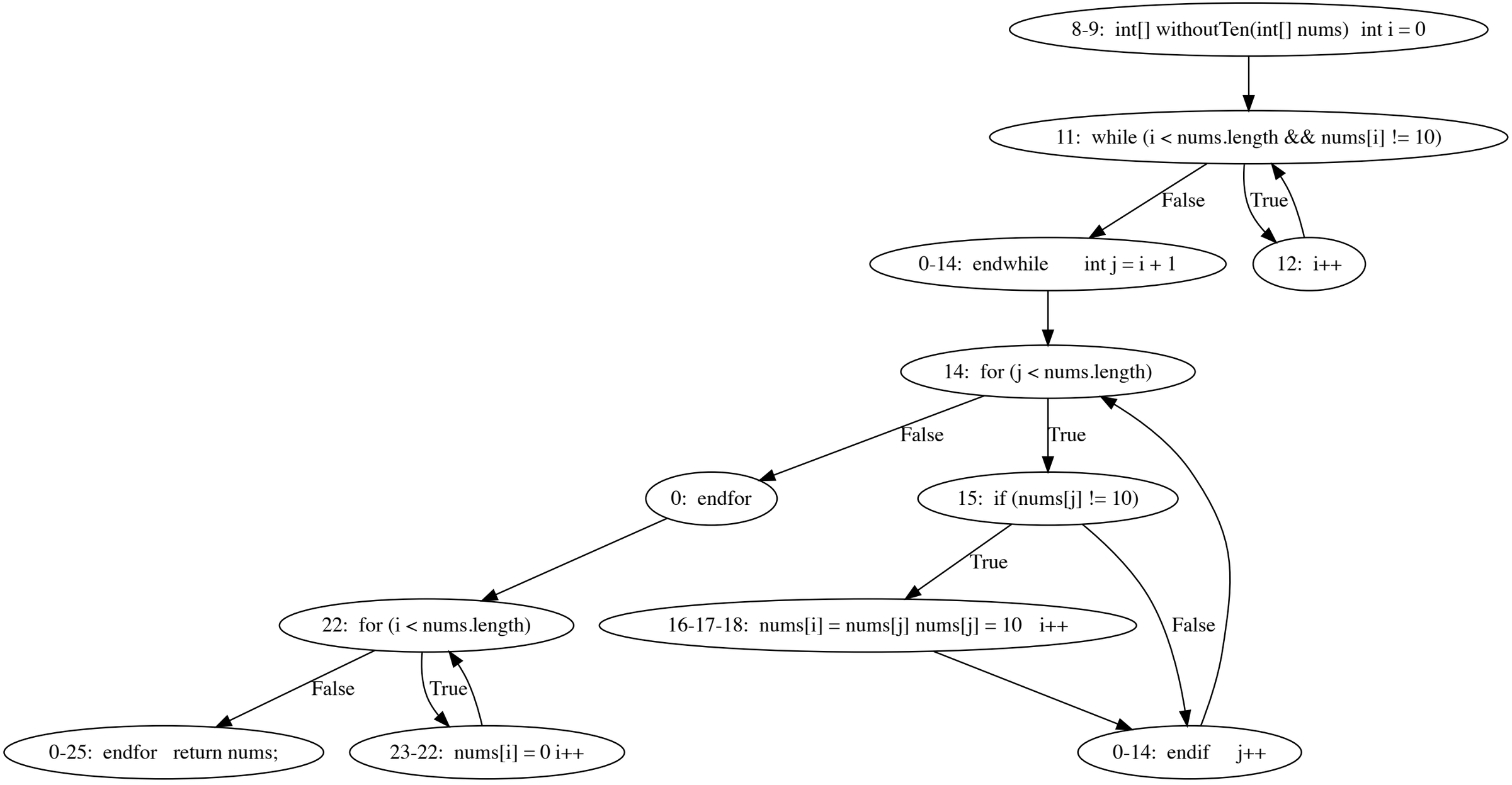

6-7-8: String front22(String str) String front if (str.length() < 2)
True
False
9: front = str
11: front = str.substring(0, 2)
0-13: endif return front + str + front;

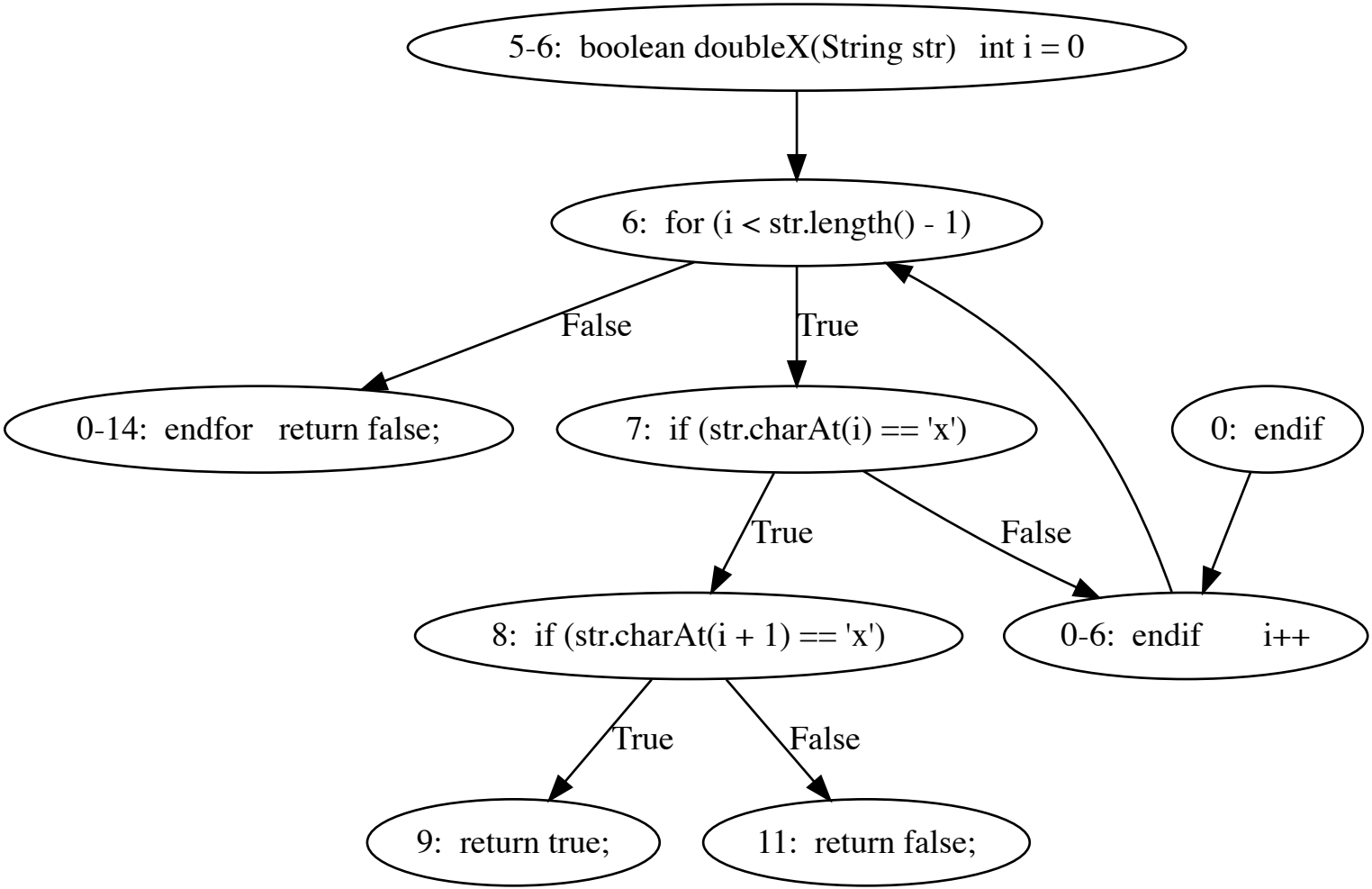

6-7-9: String startOz(String str) String result = '' if (str.length() >= 1 && str.charAt(0) == 'o')
True
10: result += str.charAt(0)
False
0-12: endif if (str.length() >= 2 && str.charAt(1) == 'z')
True
13: result += str.charAt(1)
False
0-15: endif return result;

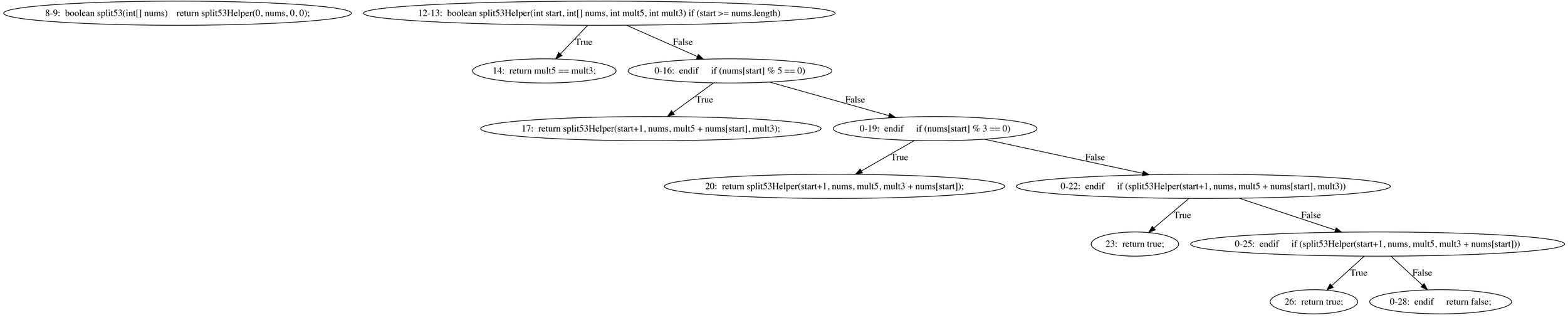

8-9: int scoresSpecial(int[] a, int[] b) return largestSpecial(a) + largestSpecial(b);
12-13-15: int largestSpecial(int[] a) int largest = 0 int i = 0
15: for (i < a.length)
False
True
0-20: endfor return largest;
16: if (a[i] % 10 == 0 && a[i] > largest)
True
17: largest = a[i]
False
0-15: endif i++

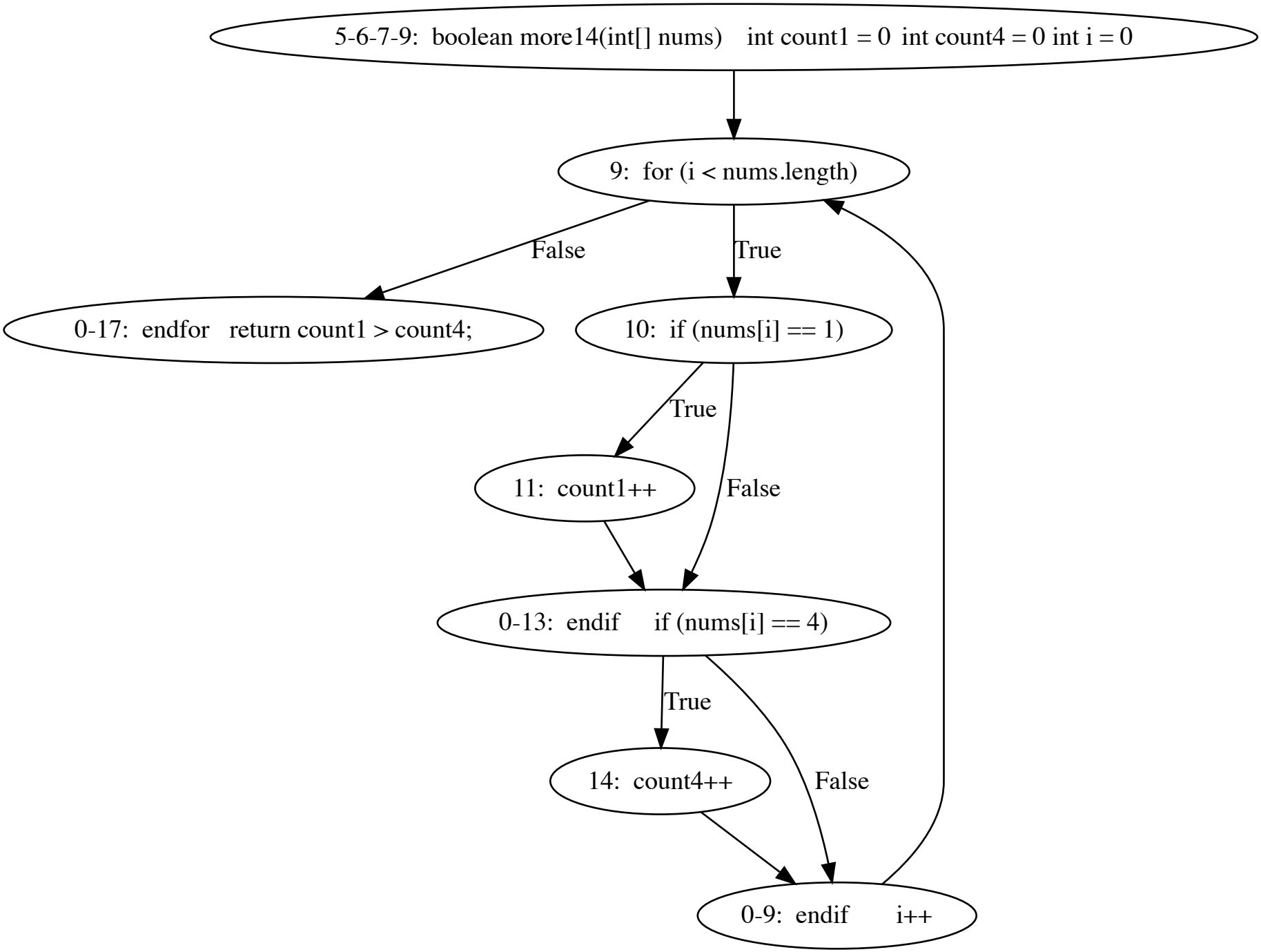

7-8: int[] evenOdd(int[] nums) int i = 0
10: while (i < nums.length && nums[i] % 2 == 0)
False
True
0-13: endwhile int j = i + 1
11: i++
13: for (j < nums.length)
False
True
0-22: endfor return nums;
14: if (nums[j] % 2 == 0)
True
False
15-16-17-18: int temp = nums[i] nums[i] = nums[j] nums[j] = temp i++
0-13: endif j++

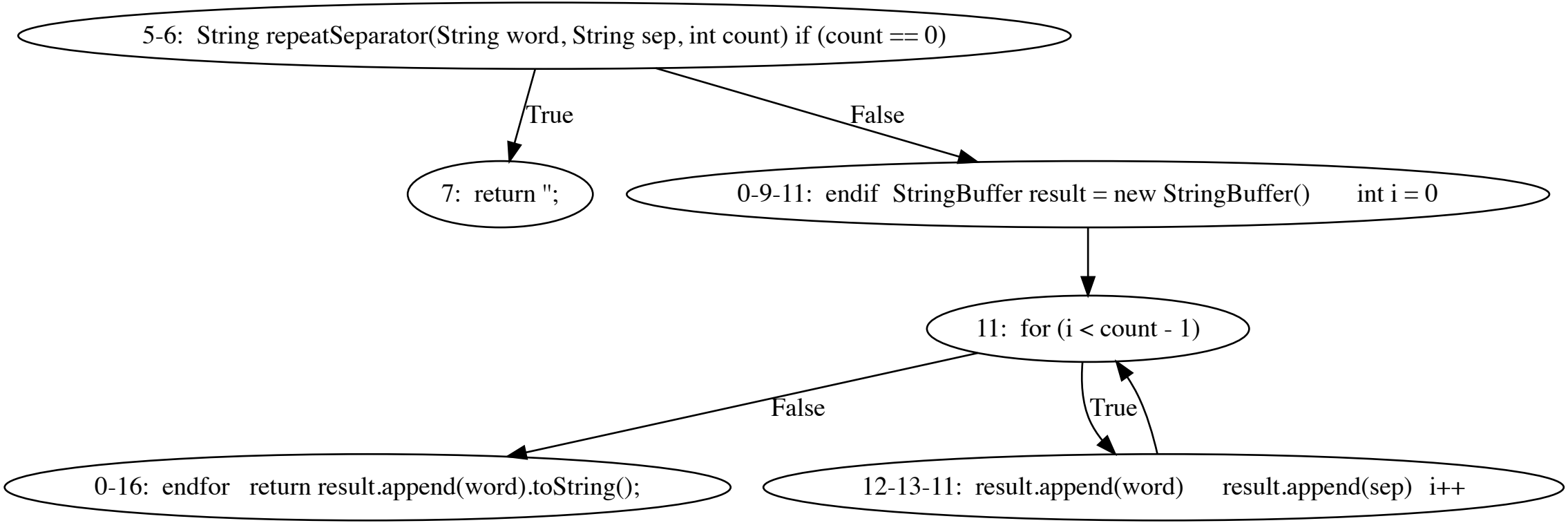

6-7: int last2(String str) if (str.length() < 2)
True
False
8: return 0;
0-10-11-13: endif String end = str.substring(str.length() - 2) int count = 0 int i = 0
13: for (i < str.length() - 2)
False
True
0-18: endfor return count;
14: if (str.substring(i, i + 2).equals(end))
True
False
15: count++
0-13: endif i++

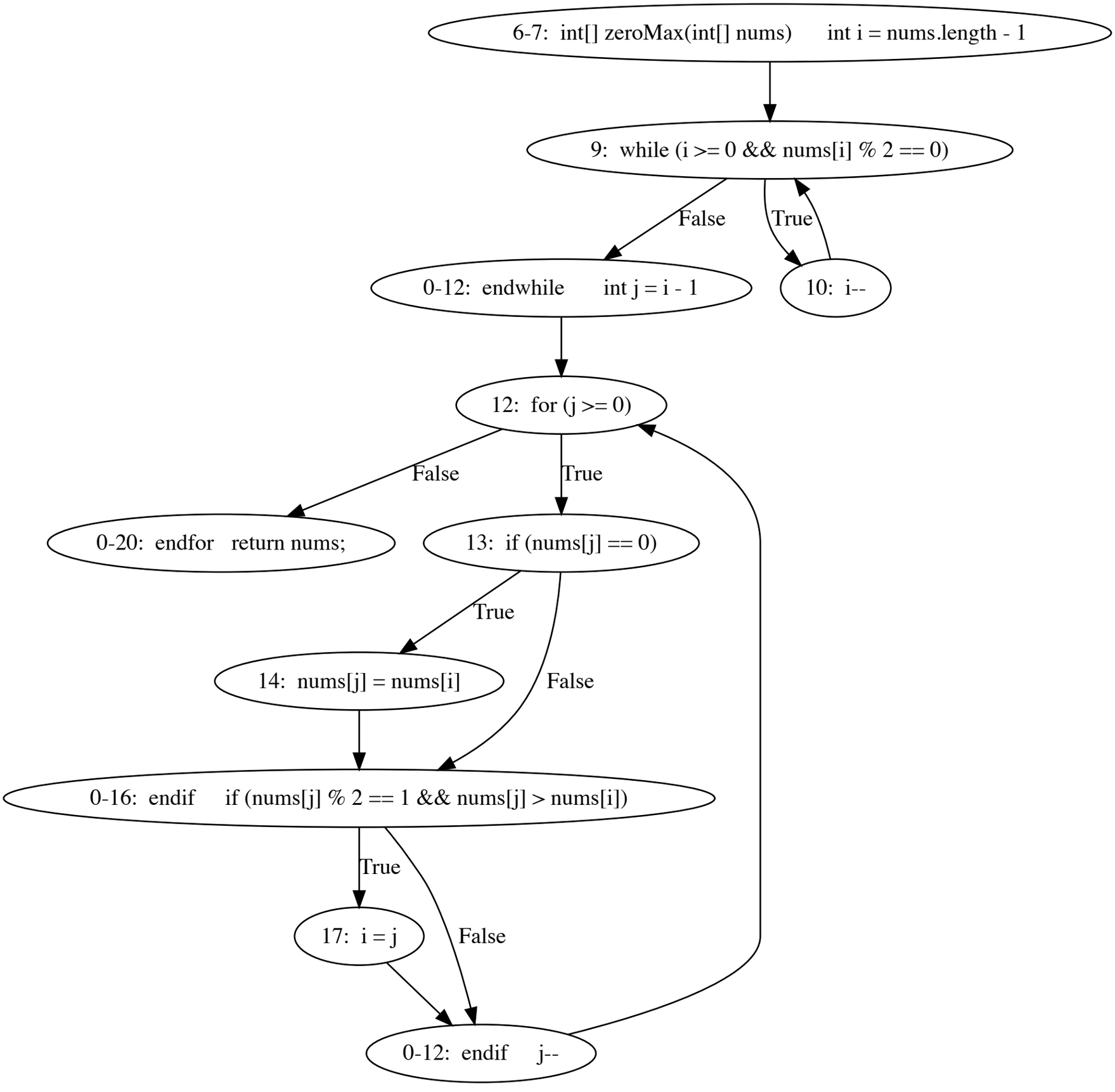

8-9: int[] withoutTen(int[] nums) int i = 0
11: while (i < nums.length && nums[i] != 10)
False
True
0-14: endwhile int j = i + 1
12: i++
14: for (j < nums.length)
False
True
0: endfor
15: if (nums[j] != 10)
True
False
22: for (i < nums.length)
16-17-18: nums[i] = nums[j] nums[j] = 10 i++
False
True
0-25: endfor return nums;
23-22: nums[i] = 0 i++
0-14: endif j++

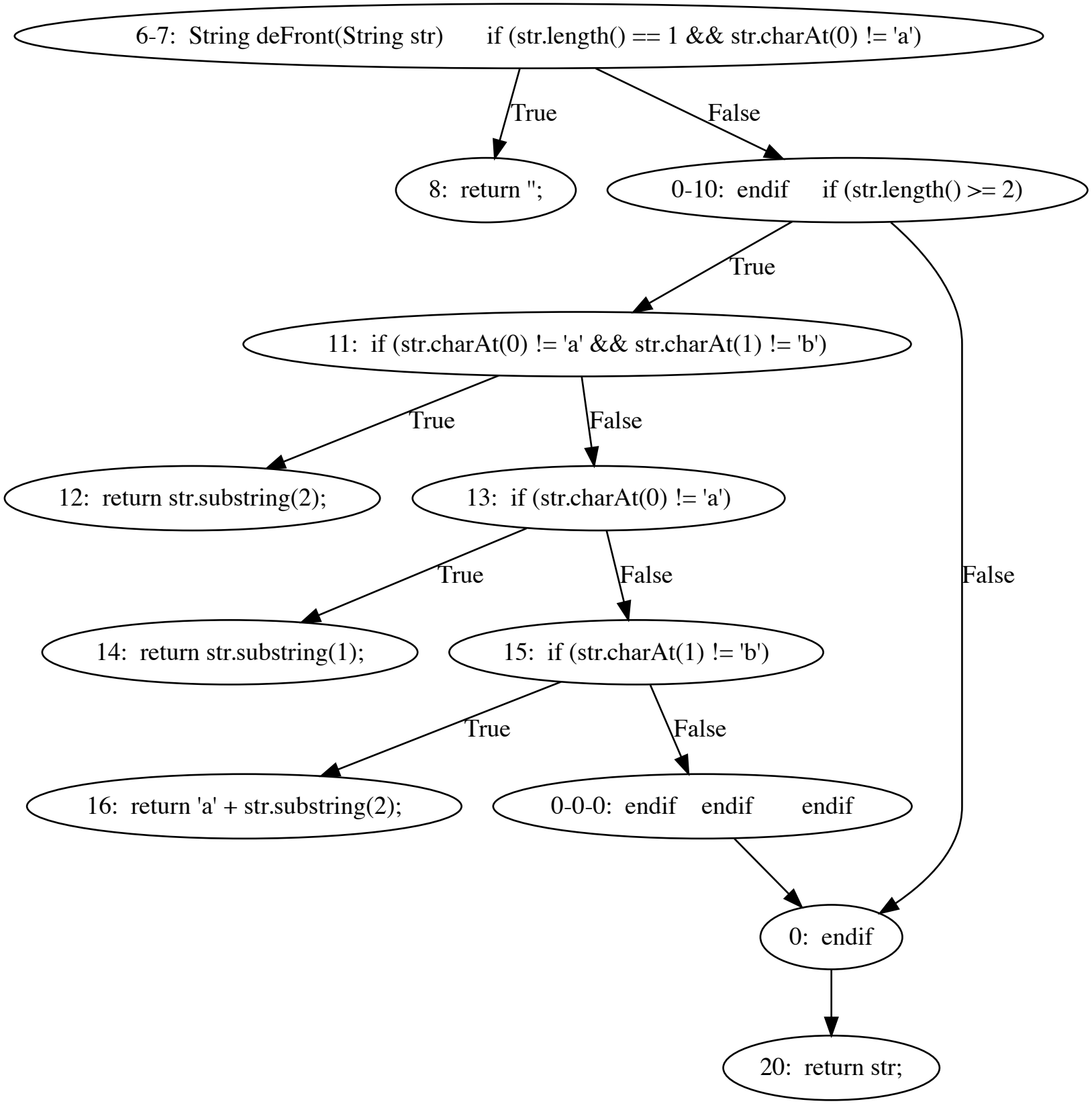

5-6: boolean doubleX(String str) int i = 0
6: for (i < str.length() - 1)
False
True
0-14: endfor return false;
7: if (str.charAt(i) == 'x')
0: endif
True
False
8: if (str.charAt(i + 1) == 'x')
0-6: endif i++
True
False
9: return true;
11: return false;

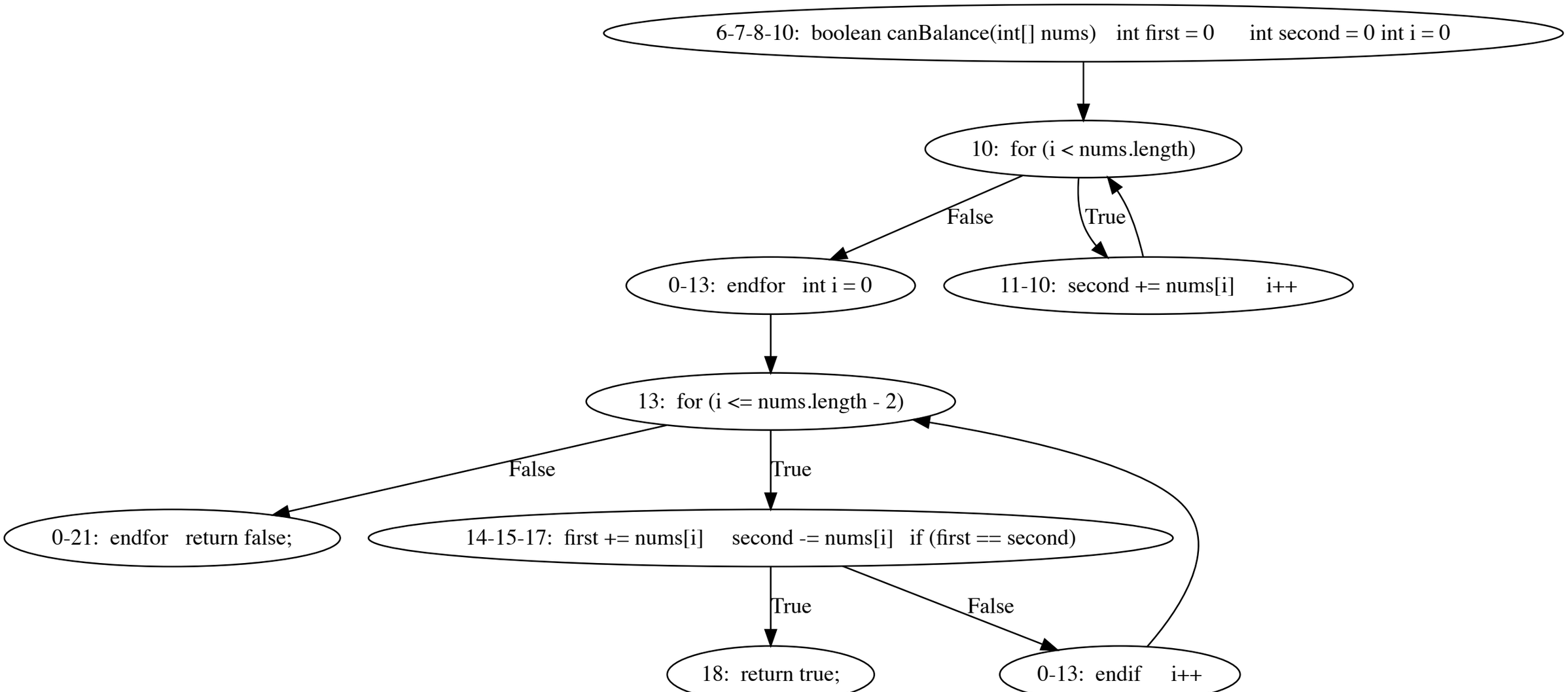

8-9: boolean split53(int[] nums) return split53Helper(0, nums, 0, 0);
12-13: boolean split53Helper(int start, int[] nums, int mult5, int mult3) if (start >= nums.length)
True
False
14: return mult5 == mult3;
0-16: endif if (nums[start] % 5 == 0)
True
False
17: return split53Helper(start+1, nums, mult5 + nums[start], mult3);
0-19: endif if (nums[start] % 3 == 0)
True
False
20: return split53Helper(start+1, nums, mult5, mult3 + nums[start]);
0-22: endif if (split53Helper(start+1, nums, mult5 + nums[start], mult3))
True
False
23: return true;
0-25: endif if (split53Helper(start+1, nums, mult5, mult3 + nums[start]))
True
False
26: return true;
0-28: endif return false;

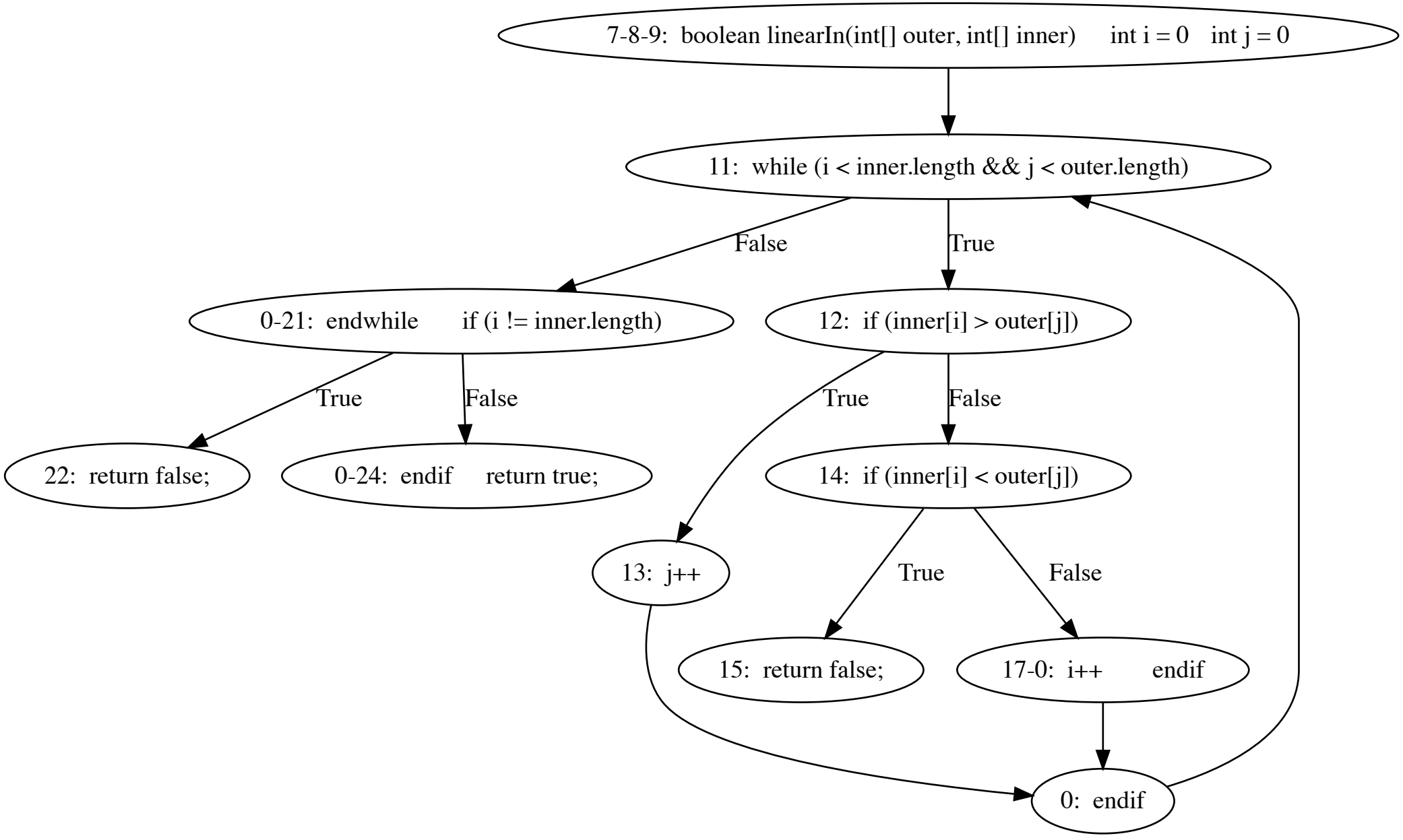

5-6-7-9: boolean more14(int[] nums) int count1 = 0 int count4 = 0 int i = 0
9: for (i < nums.length)
False
True
0-17: endfor return count1 > count4;
10: if (nums[i] == 1)
True
11: count1++
False
0-13: endif if (nums[i] == 4)
True
14: count4++
False
0-9: endif i++

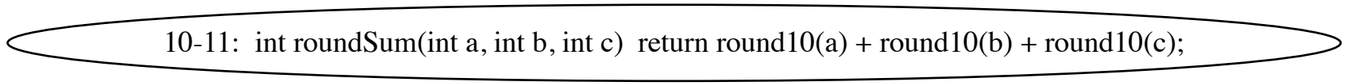

5-6: String repeatSeparator(String word, String sep, int count) if (count == 0)
True
False
7: return '';
0-9-11: endif StringBuffer result = new StringBuffer() int i = 0
11: for (i < count - 1)
False
True
0-16: endfor return result.append(word).toString();
12-13-11: result.append(word) result.append(sep) i++

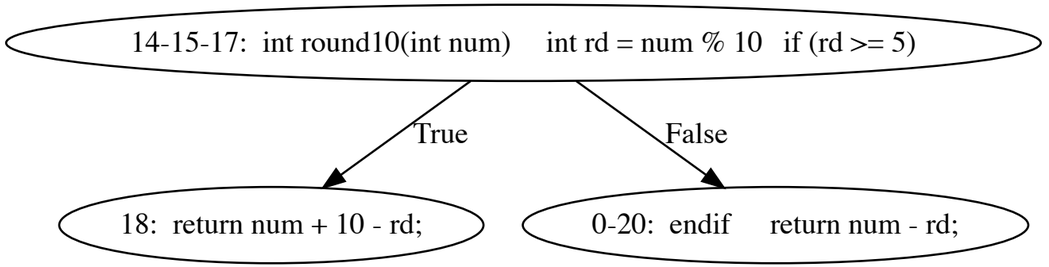

6-7: int[] zeroMax(int[] nums) int i = nums.length - 1
9: while (i >= 0 && nums[i] % 2 == 0)
False
True
0-12: endwhile int j = i - 1
10: i--
12: for (j >= 0)
False
True
0-20: endfor return nums;
13: if (nums[j] == 0)
True
14: nums[j] = nums[i]
False
0-16: endif if (nums[j] % 2 == 1 && nums[j] > nums[i])
True
17: i = j
False
0-12: endif j--

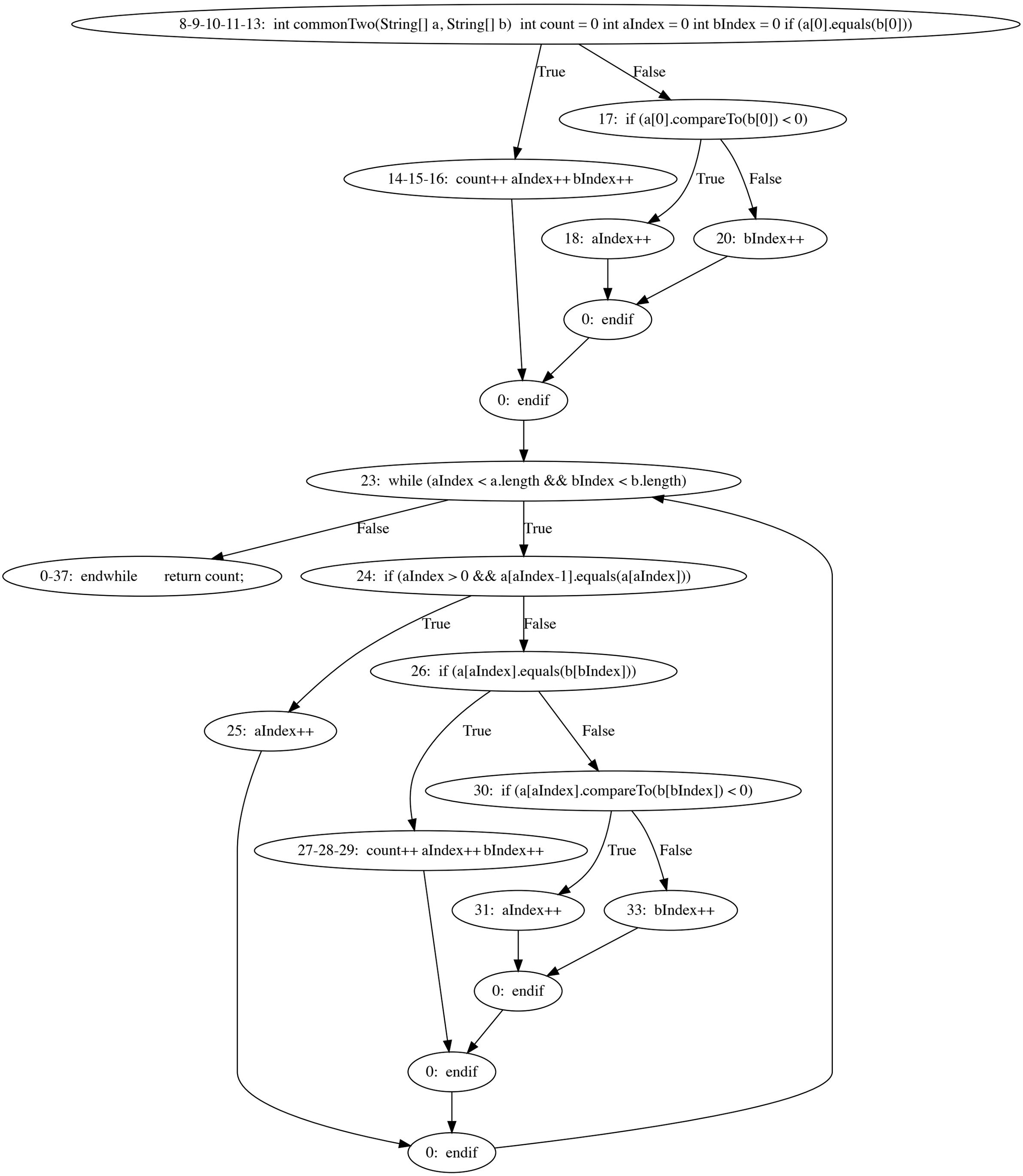

6-7: String deFront(String str) if (str.length() == 1 && str.charAt(0) != 'a')
True
False
8: return '';
0-10: endif if (str.length() >= 2)
True
11: if (str.charAt(0) != 'a' && str.charAt(1) != 'b')
True
False
12: return str.substring(2);
13: if (str.charAt(0) != 'a')
True
False
14: return str.substring(1);
15: if (str.charAt(1) != 'b')
True
False
16: return 'a' + str.substring(2);
0-0-0: endif endif endif
False
0: endif
20: return str;

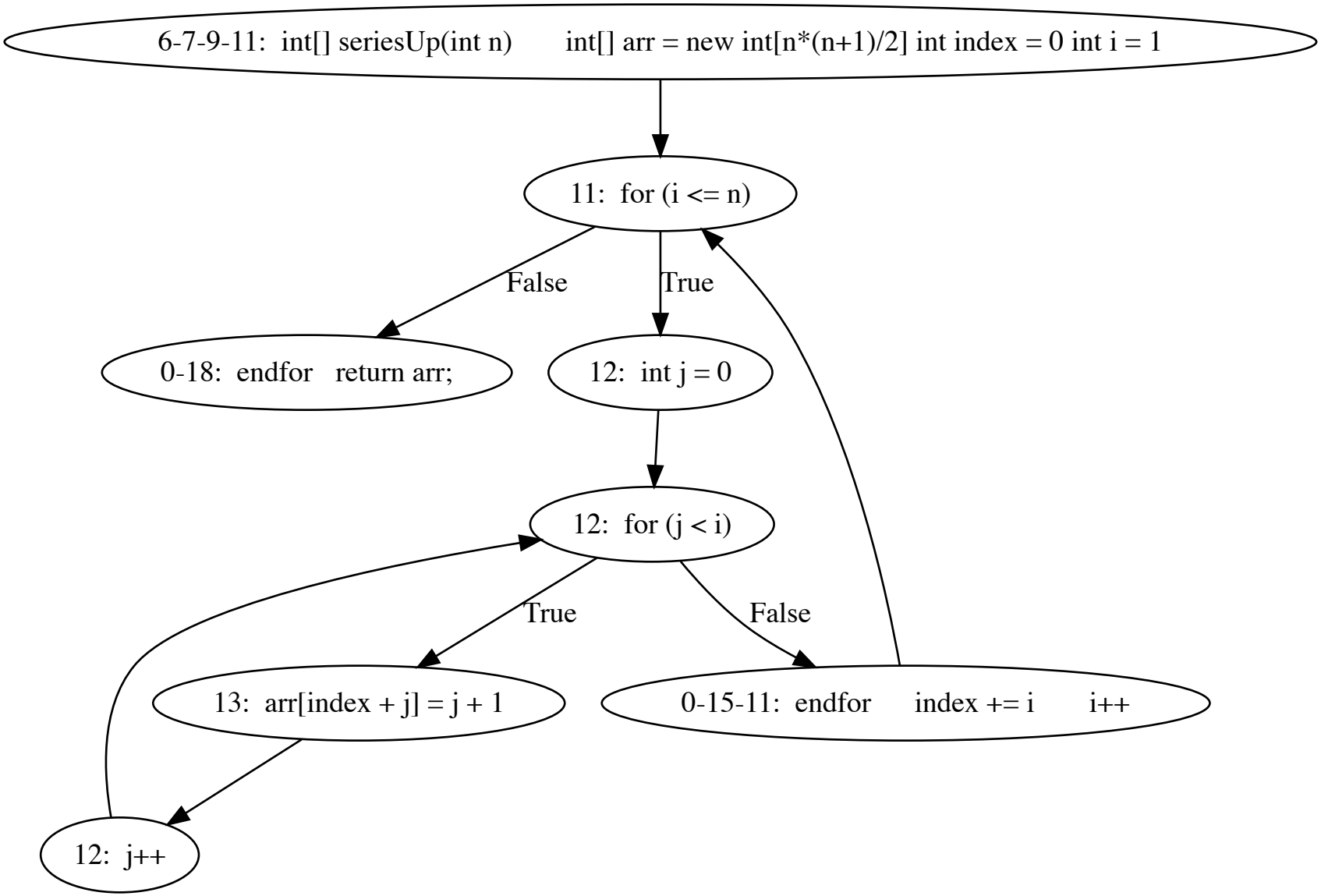

6-7-8-10: boolean canBalance(int[] nums) int first = 0 int second = 0 int i = 0
10: for (i < nums.length)
False
True
0-13: endfor int i = 0
11-10: second += nums[i] i++
13: for (i <= nums.length - 2)
False
True
0-21: endfor return false;
14-15-17: first += nums[i] second -= nums[i] if (first == second)
True
False
18: return true;
0-13: endif i++

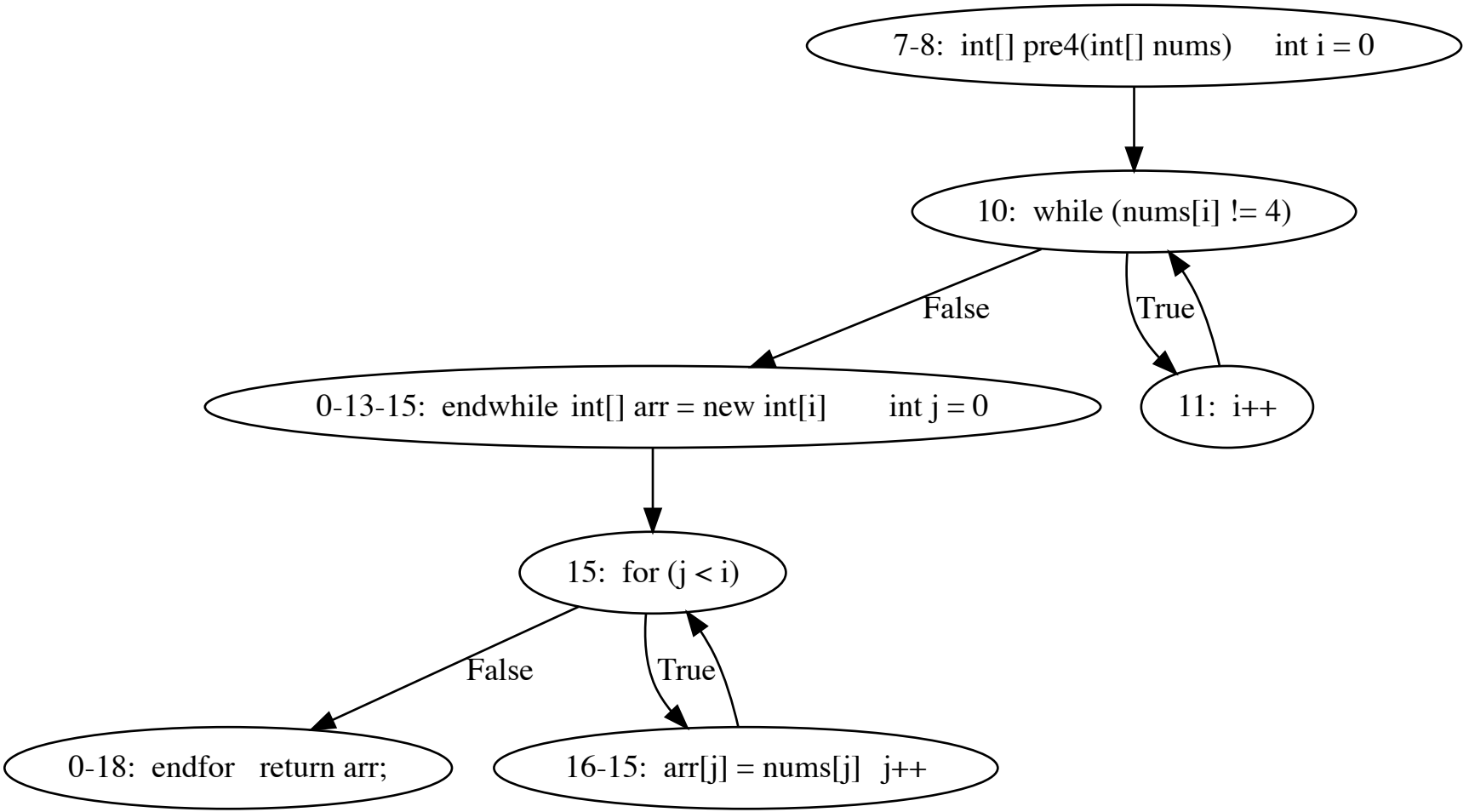

7-8-9: boolean linearIn(int[] outer, int[] inner) int i = 0 int j = 0
11: while (i < inner.length && j < outer.length)
False
True
0-21: endwhile if (i != inner.length)
12: if (inner[i] > outer[j])
True
False
True
False
22: return false;
0-24: endif return true;
14: if (inner[i] < outer[j])
13: j++
True
False
15: return false;
17-0: i++ endif
0: endif

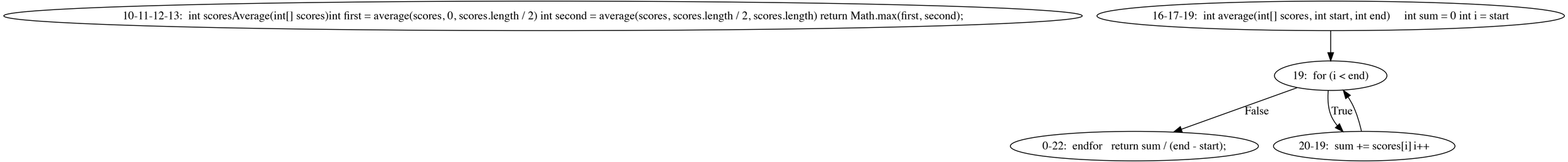

10-11: int roundSum(int a, int b, int c) return round10(a) + round10(b) + round10(c);

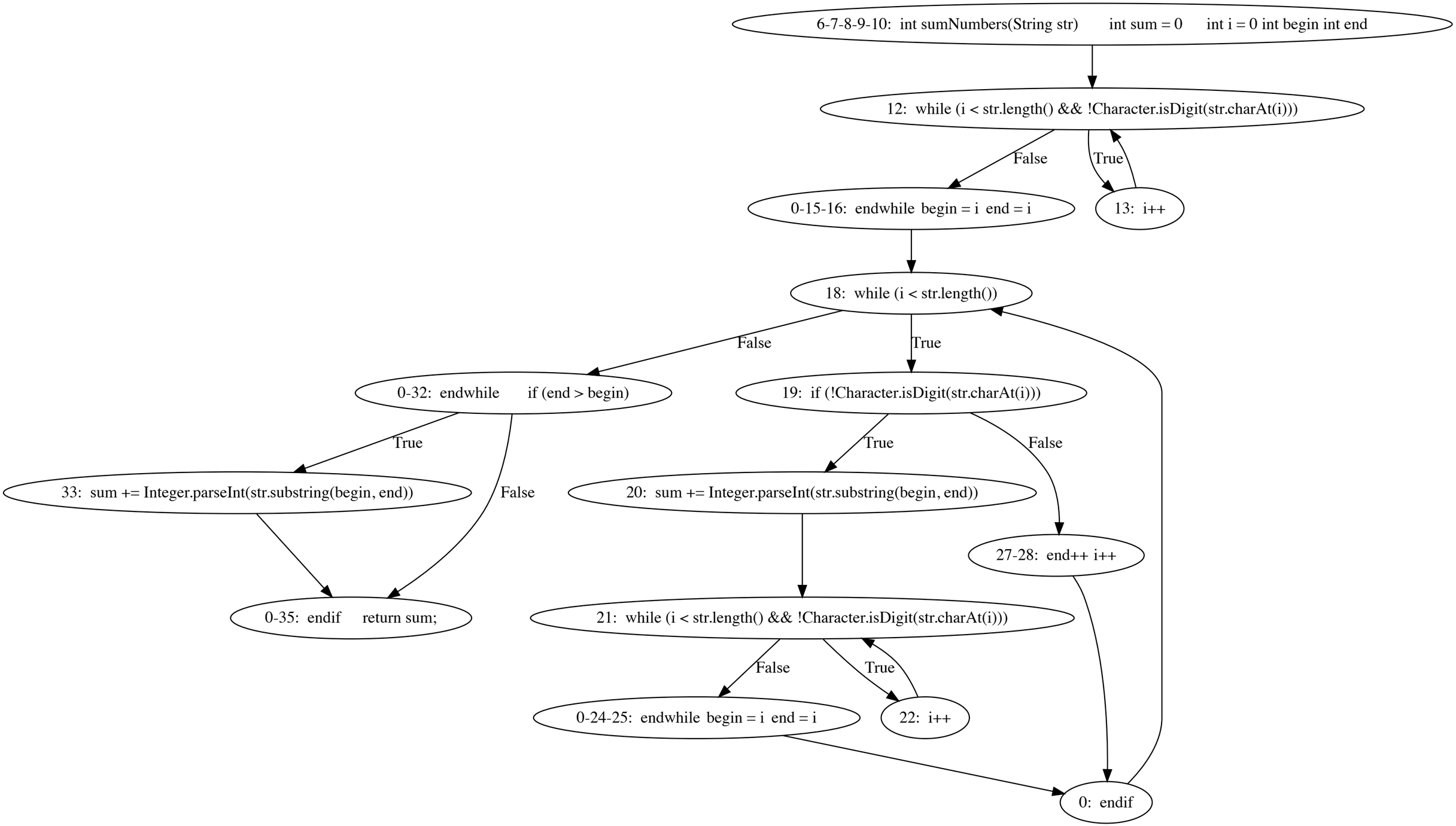

14-15-17: int round10(int num) int rd = num % 10 if (rd >= 5)
True
False
18: return num + 10 - rd;
0-20: endif return num - rd;

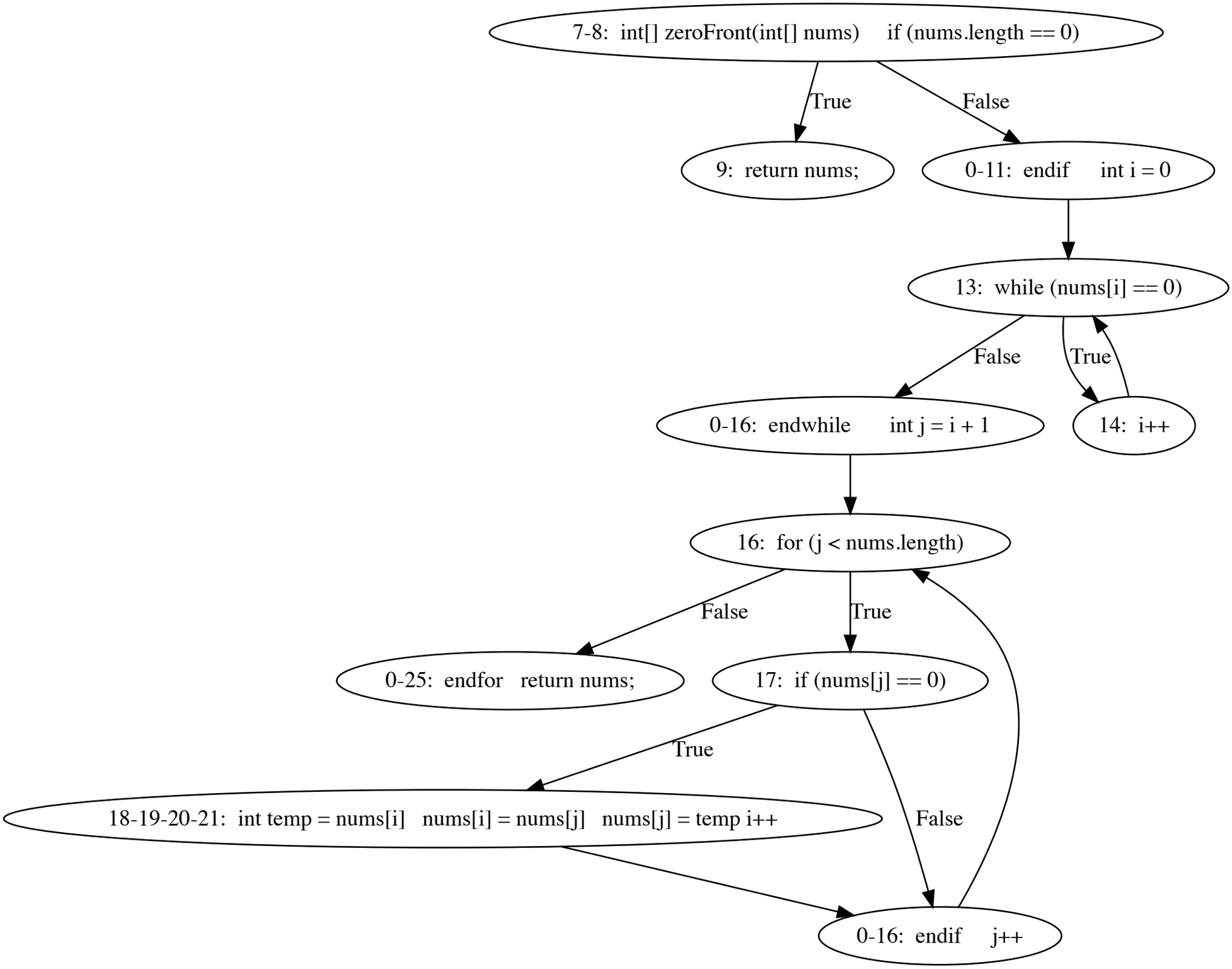

8-9-10-11-13: int commonTwo(String[] a, String[] b) int count = 0 int aIndex = 0 int bIndex = 0 if (a[0].equals(b[0]))
True
False
17: if (a[0].compareTo(b[0]) < 0)
14-15-16: count++ aIndex++ bIndex++
True
False
18: aIndex++
20: bIndex++
0: endif
0: endif
23: while (aIndex < a.length && bIndex < b.length)
False
True
0-37: endwhile return count;
24: if (aIndex > 0 && a[aIndex-1].equals(a[aIndex]))
True
False
26: if (a[aIndex].equals(b[bIndex]))
25: aIndex++
True
False
30: if (a[aIndex].compareTo(b[bIndex]) < 0)
27-28-29: count++ aIndex++ bIndex++
True
False
31: aIndex++
33: bIndex++
0: endif
0: endif
0: endif

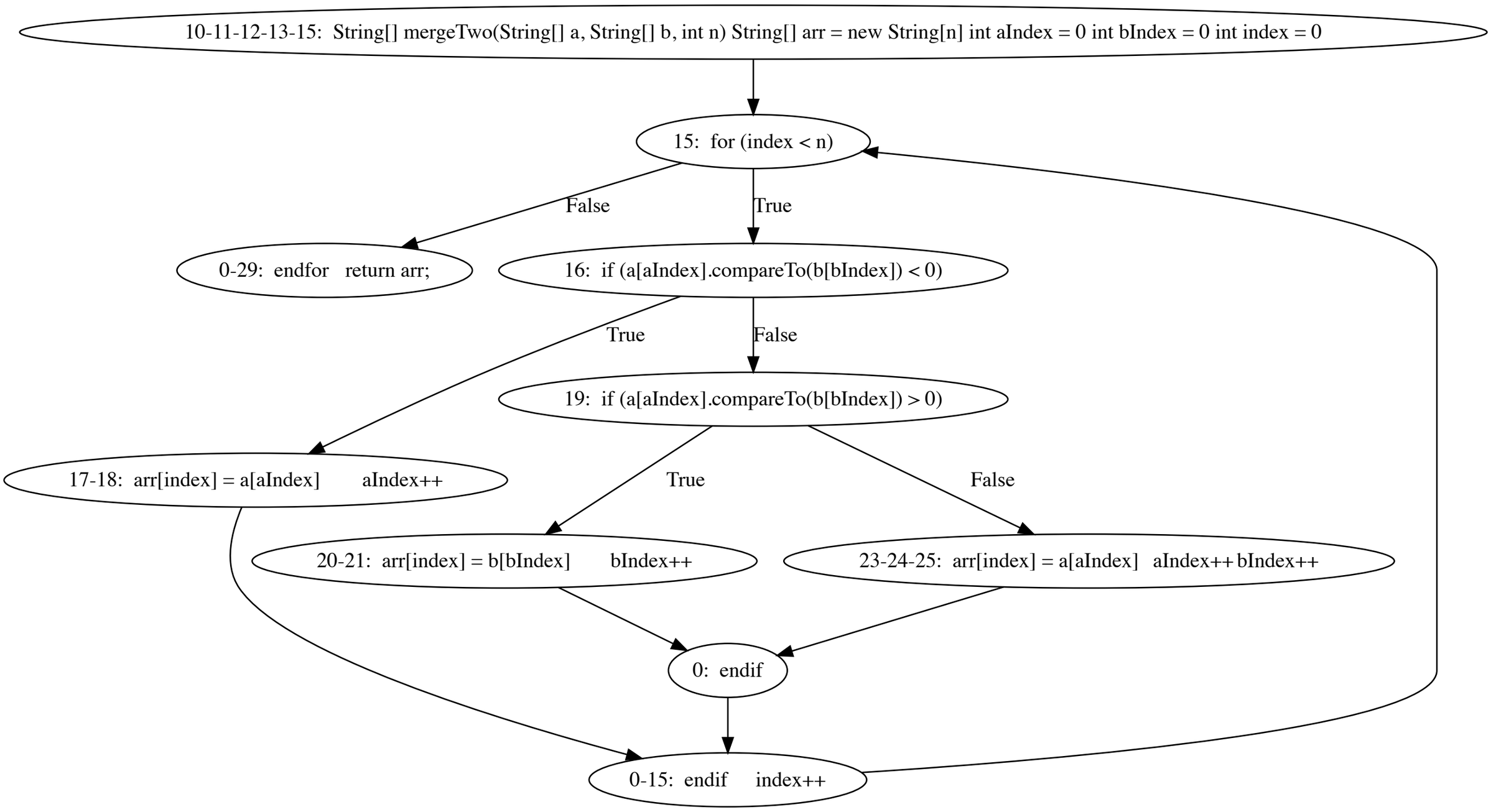

6-7-9-11: int[] seriesUp(int n) int[] arr = new int[n*(n+1)/2] int index = 0 int i = 1
11: for (i <= n)
False
True
0-18: endfor return arr;
12: int j = 0
12: for (j < i)
True
False
13: arr[index + j] = j + 1
0-15-11: endfor index += i i++
12: j++

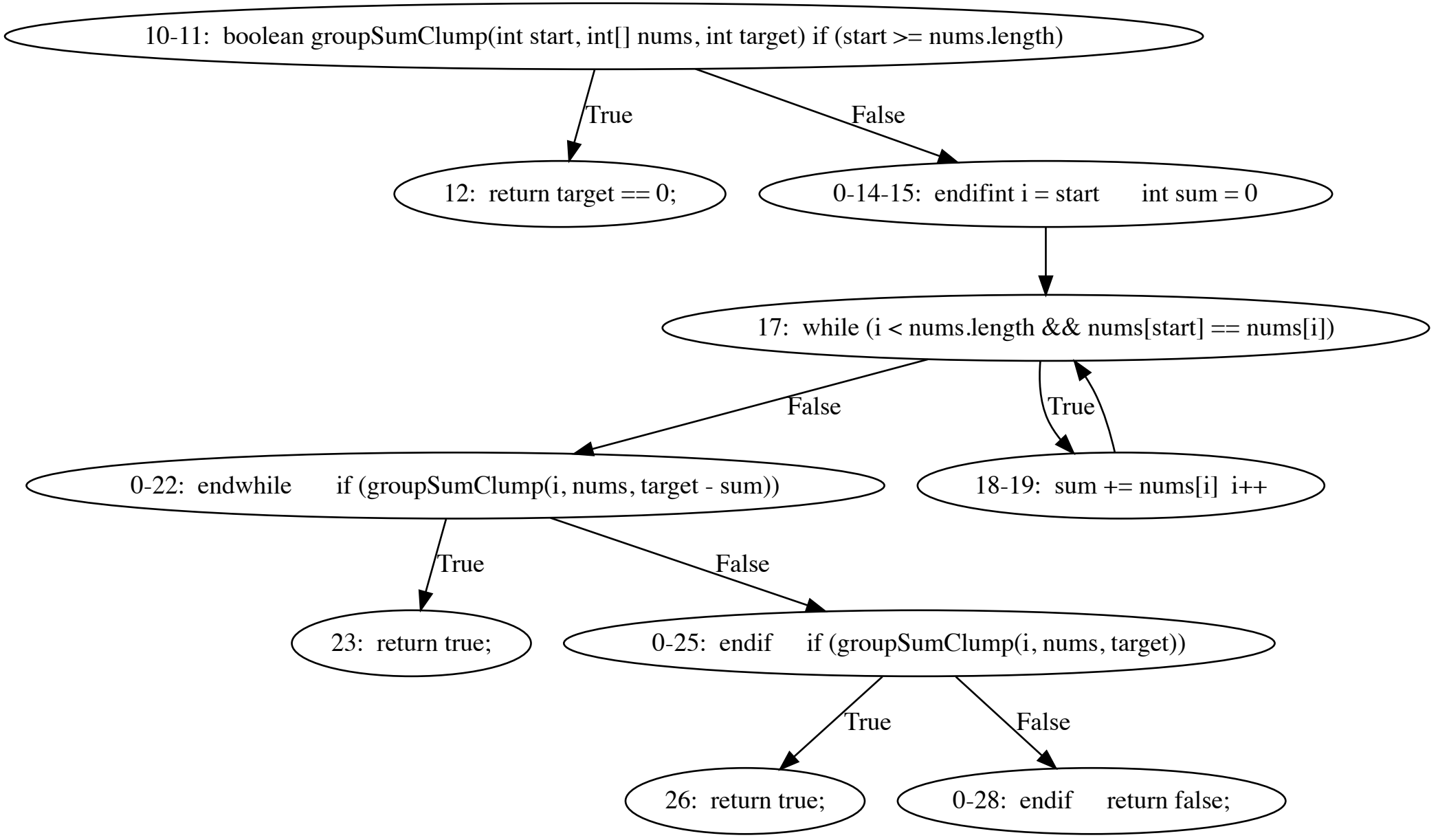

7-8: int[] pre4(int[] nums) int i = 0
10: while (nums[i] != 4)
False
True
0-13-15: endwhile int[] arr = new int[i] int j = 0
11: i++
15: for (j < i)
False
True
0-18: endfor return arr;
16-15: arr[j] = nums[j] j++

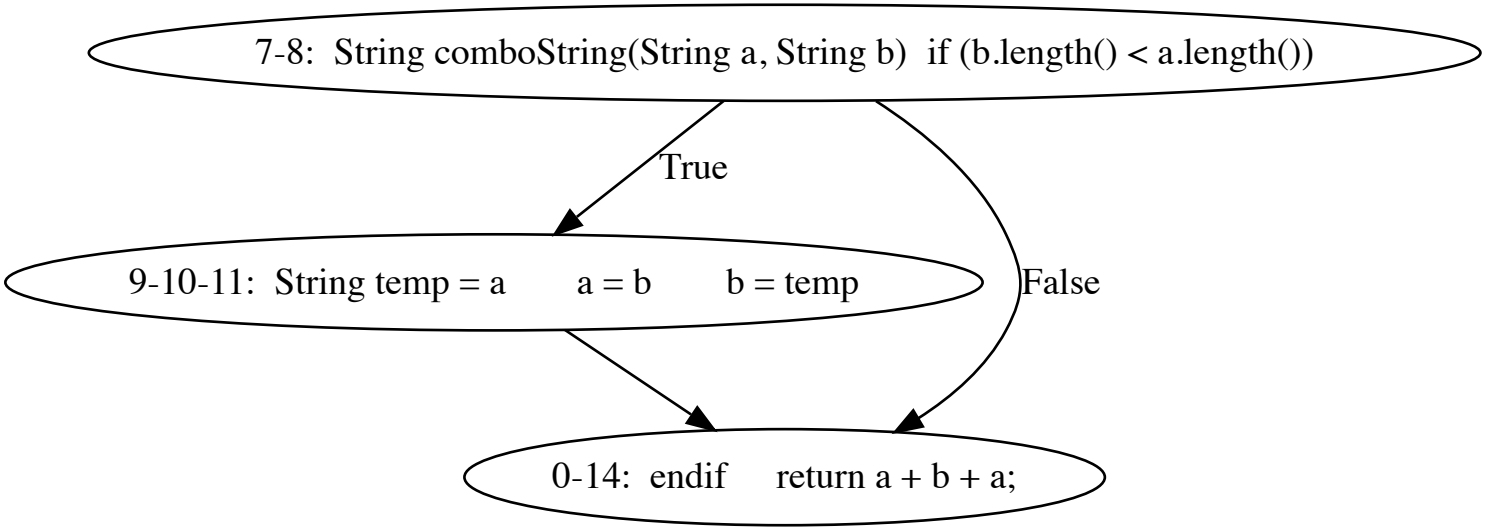

10-11-12-13: int scoresAverage(int[] scores)int first = average(scores, 0, scores.length / 2) int second = average(scores, scores.length / 2, scores.length) return Math.max(first, second);
16-17-19: int average(int[] scores, int start, int end) int sum = 0 int i = start
19: for (i < end)
False
True
0-22: endfor return sum / (end - start);
20-19: sum += scores[i] i++

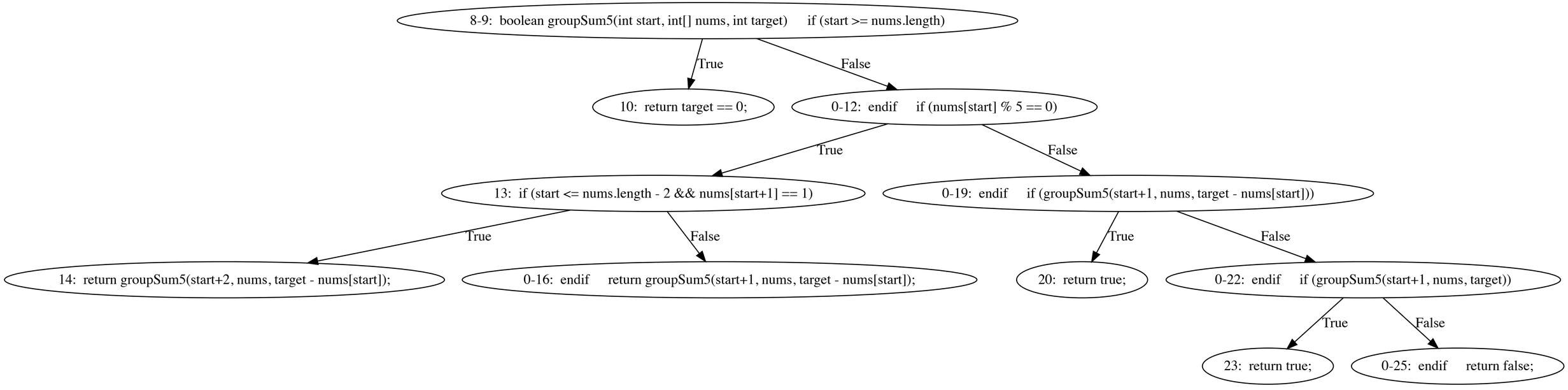

6-7-8-9-10: int sumNumbers(String str) int sum = 0 int i = 0 int begin int end
12: while (i < str.length() && !Character.isDigit(str.charAt(i)))
False
True
0-15-16: endwhile begin = i end = i
13: i++
18: while (i < str.length())
False
True
0-32: endwhile if (end > begin)
19: if (!Character.isDigit(str.charAt(i)))
True
False
True
False
33: sum += Integer.parseInt(str.substring(begin, end))
20: sum += Integer.parseInt(str.substring(begin, end))
27-28: end++ i++
0-35: endif return sum;
21: while (i < str.length() && !Character.isDigit(str.charAt(i)))
False
True
0-24-25: endwhile begin = i end = i
22: i++
0: endif

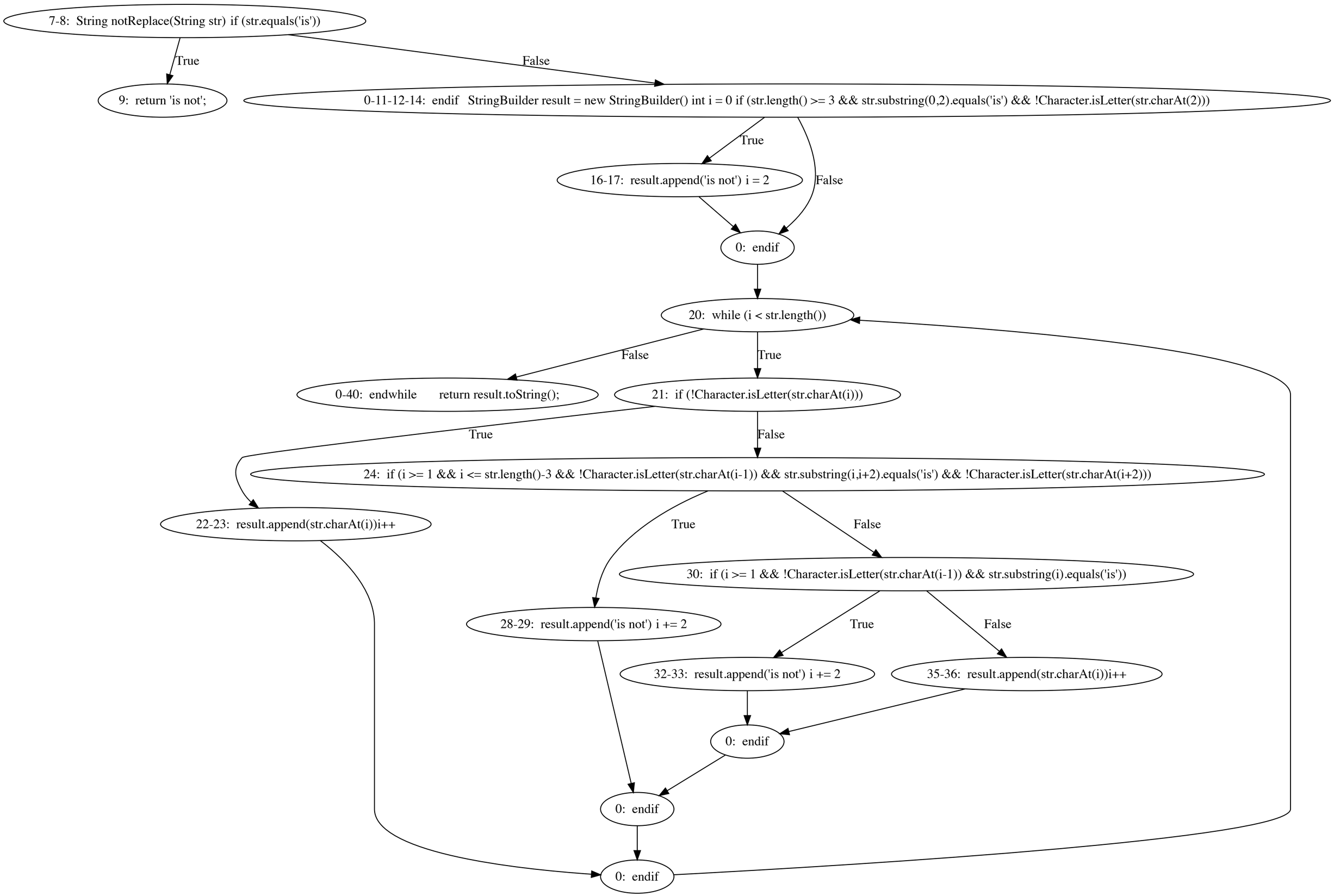

7-8: int[] zeroFront(int[] nums) if (nums.length == 0)
True
False
9: return nums;
0-11: endif int i = 0
13: while (nums[i] == 0)
False
True
0-16: endwhile int j = i + 1
14: i++
16: for (j < nums.length)
False
True
0-25: endfor return nums;
17: if (nums[j] == 0)
True
False
18-19-20-21: int temp = nums[i] nums[i] = nums[j] nums[j] = temp i++
0-16: endif j++

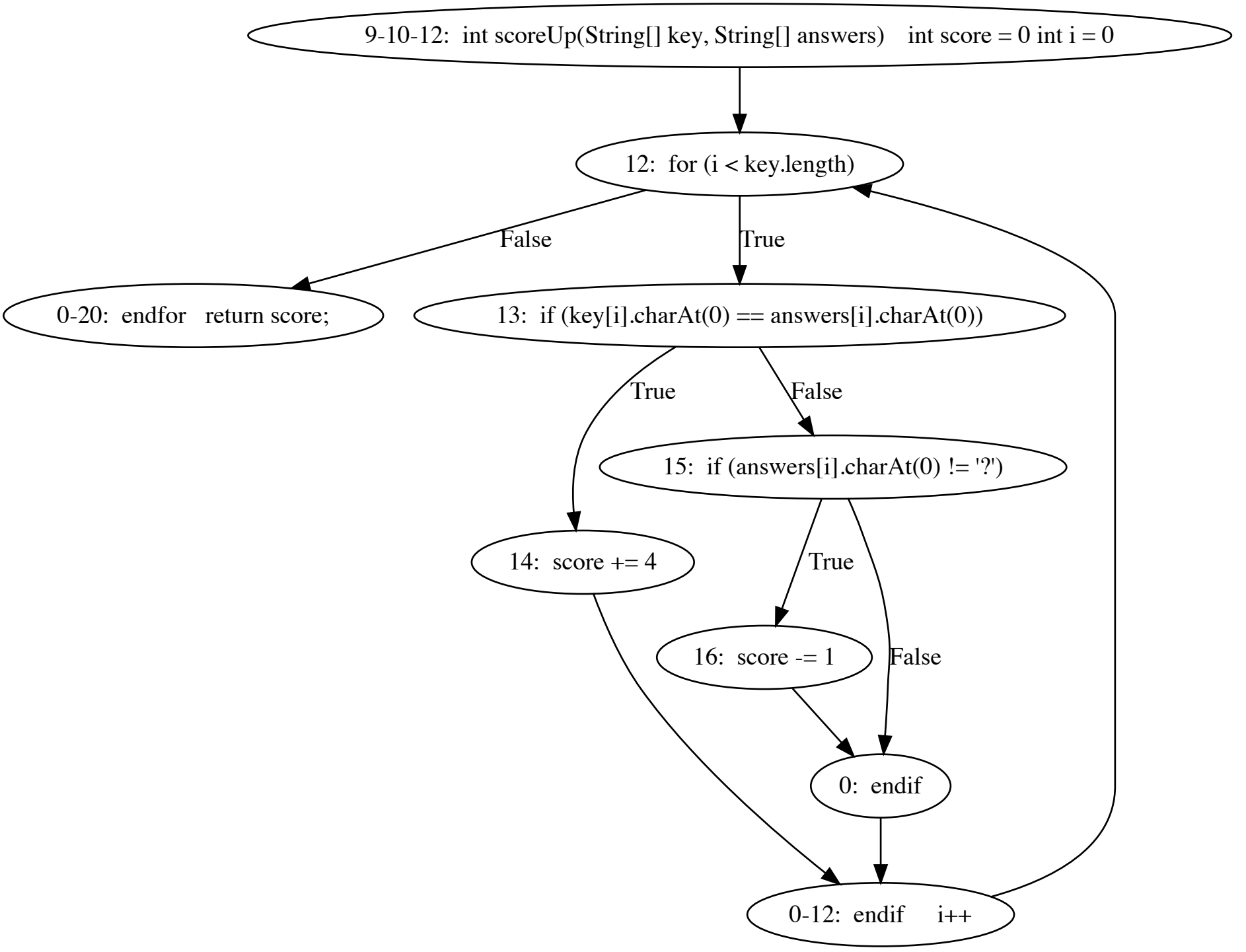

10-11-12-13-15: String[] mergeTwo(String[] a, String[] b, int n) String[] arr = new String[n] int aIndex = 0 int bIndex = 0 int index = 0
15: for (index < n)
False
True
0-29: endfor return arr;
16: if (a[aIndex].compareTo(b[bIndex]) < 0)
True
False
19: if (a[aIndex].compareTo(b[bIndex]) > 0)
17-18: arr[index] = a[aIndex] aIndex++
True
False
20-21: arr[index] = b[bIndex] bIndex++
23-24-25: arr[index] = a[aIndex] aIndex++ bIndex++
0: endif
0-15: endif index++

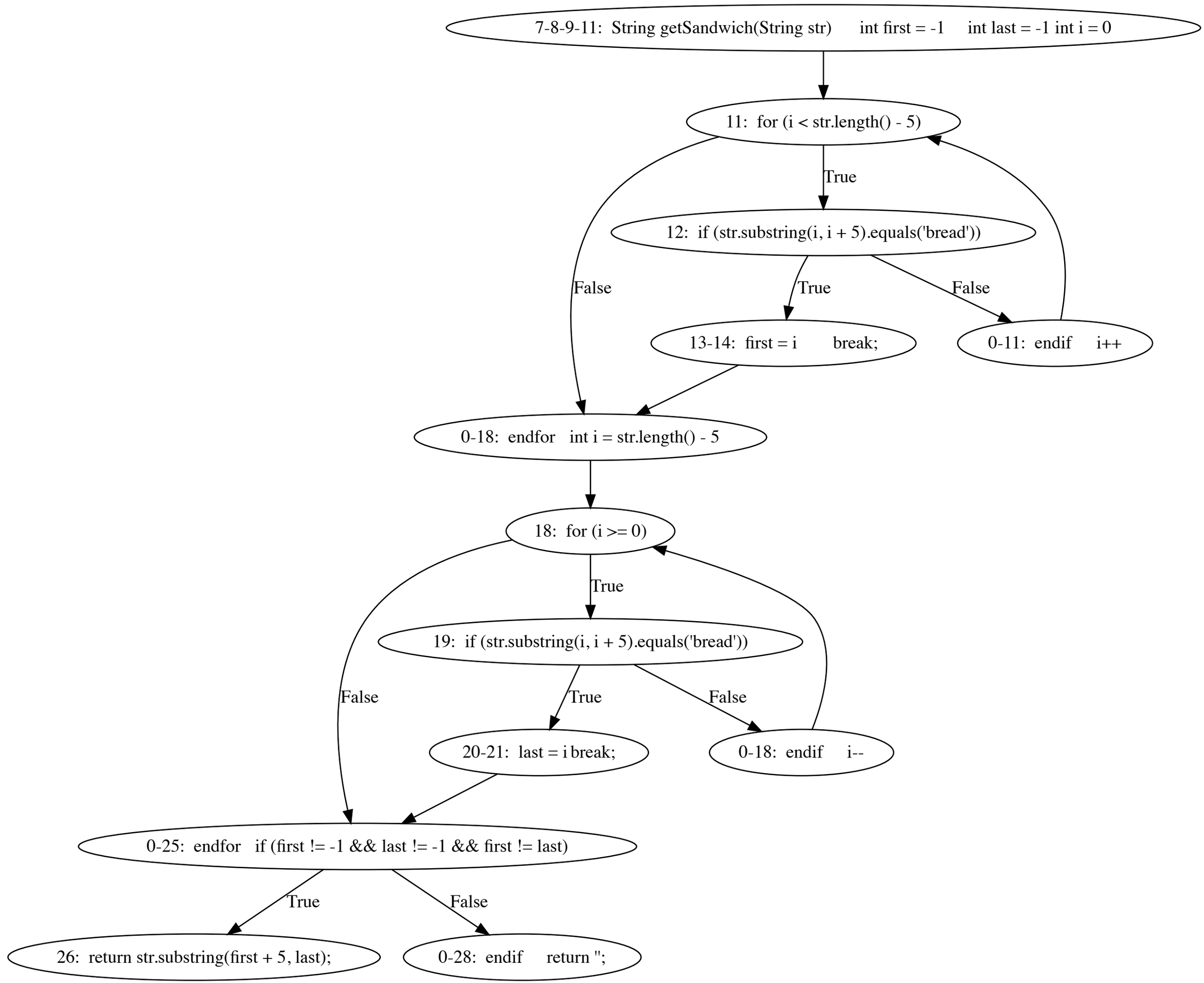

10-11: boolean groupSumClump(int start, int[] nums, int target) if (start >= nums.length)
True
False
12: return target == 0;
0-14-15: endifint i = start int sum = 0
17: while (i < nums.length && nums[start] == nums[i])
False
True
0-22: endwhile if (groupSumClump(i, nums, target - sum))
18-19: sum += nums[i] i++
True
False
23: return true;
0-25: endif if (groupSumClump(i, nums, target))
True
False
26: return true;
0-28: endif return false;

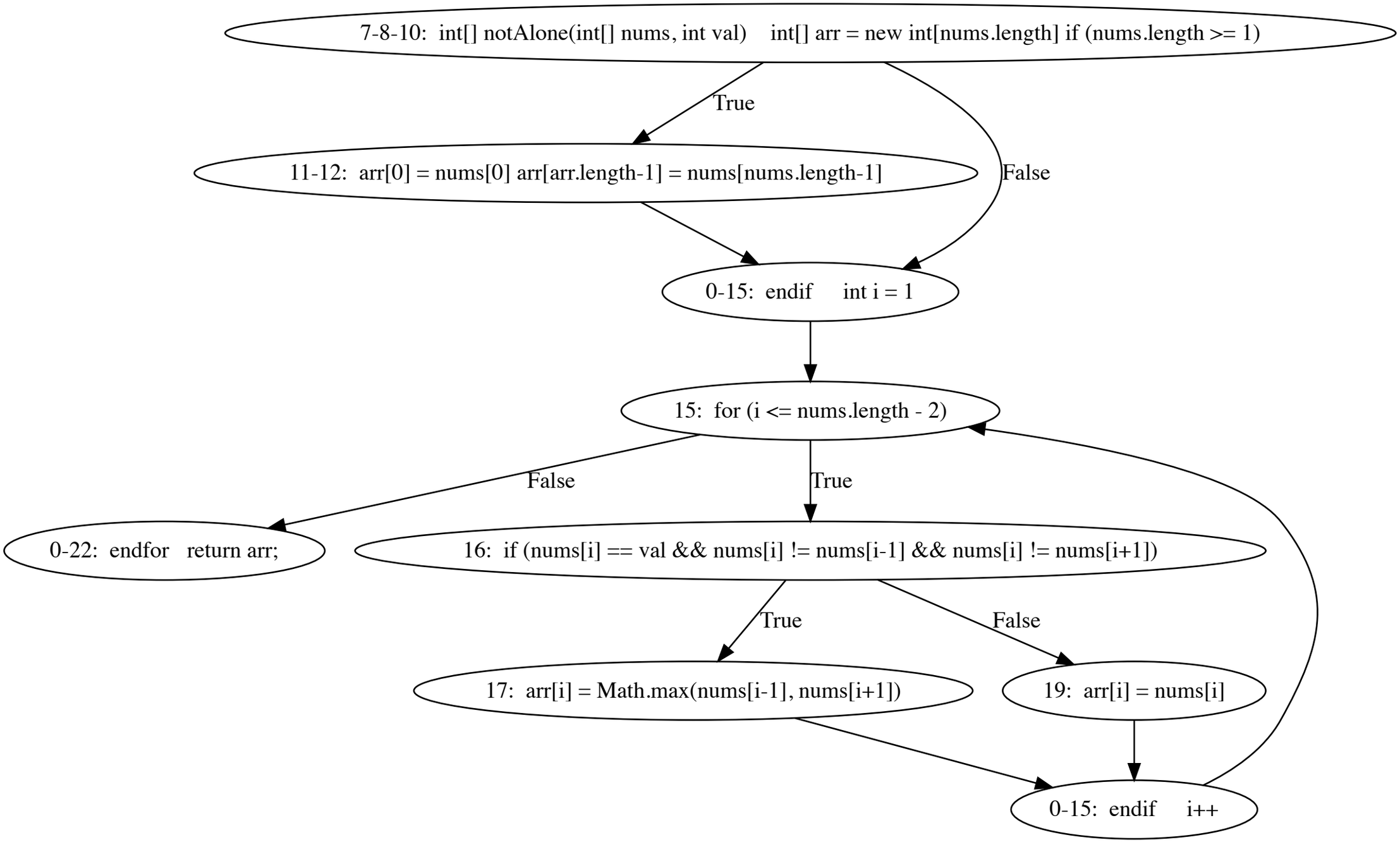

7-8: String comboString(String a, String b) if (b.length() < a.length())
True
False
9-10-11: String temp = a a = b b = temp
0-14: endif return a + b + a;

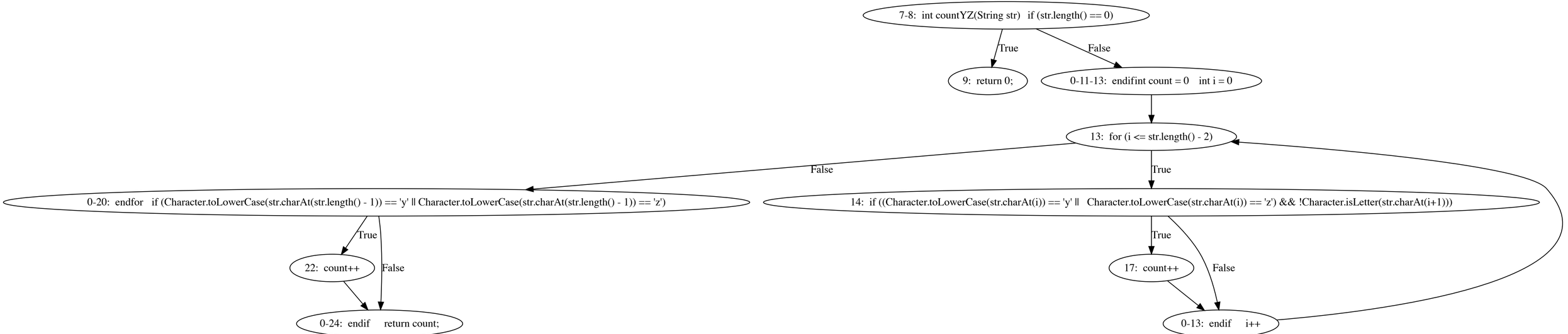

8-9: boolean groupSum5(int start, int[] nums, int target) if (start >= nums.length)
True
False
10: return target == 0;
0-12: endif if (nums[start] % 5 == 0)
True
False
13: if (start <= nums.length - 2 && nums[start+1] == 1)
0-19: endif if (groupSum5(start+1, nums, target - nums[start]))
True
False
True
False
14: return groupSum5(start+2, nums, target - nums[start]);
0-16: endif return groupSum5(start+1, nums, target - nums[start]);
20: return true;
0-22: endif if (groupSum5(start+1, nums, target))
True
False
23: return true;
0-25: endif return false;

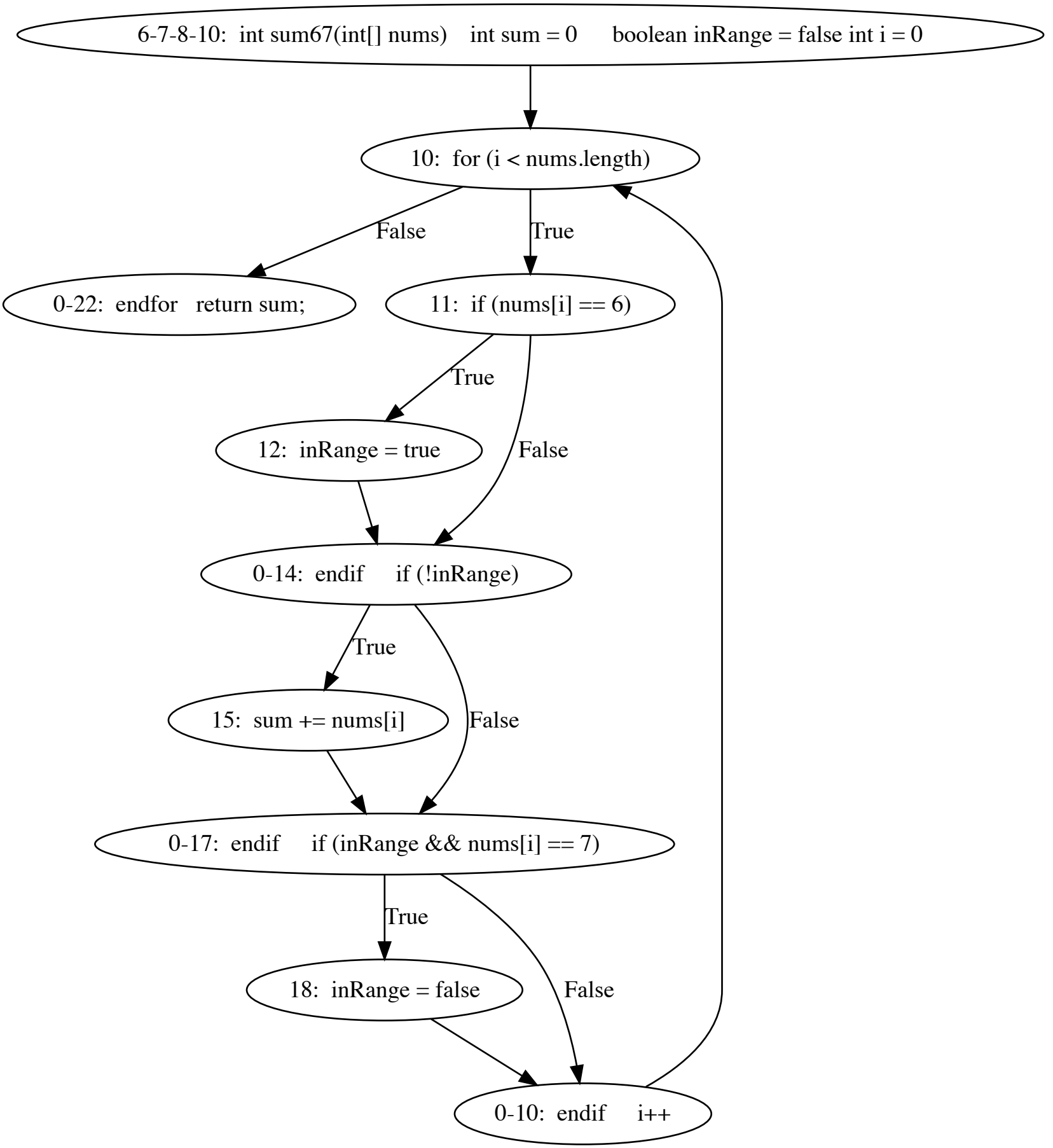

7-8: String notReplace(String str) if (str.equals('is'))
True
False
9: return 'is not';
0-11-12-14: endif StringBuilder result = new StringBuilder() int i = 0 if (str.length() >= 3 && str.substring(0,2).equals('is') && !Character.isLetter(str.charAt(2)))
True
False
16-17: result.append('is not') i = 2
0: endif
20: while (i < str.length())
False
True
0-40: endwhile return result.toString();
21: if (!Character.isLetter(str.charAt(i)))
True
False
24: if (i >= 1 && i <= str.length()-3 && !Character.isLetter(str.charAt(i-1)) && str.substring(i,i+2).equals('is') && !Character.isLetter(str.charAt(i+2)))
22-23: result.append(str.charAt(i))i++
True
False
30: if (i >= 1 && !Character.isLetter(str.charAt(i-1)) && str.substring(i).equals('is'))
28-29: result.append('is not') i += 2
True
False
32-33: result.append('is not') i += 2
35-36: result.append(str.charAt(i))i++
0: endif
0: endif
0: endif

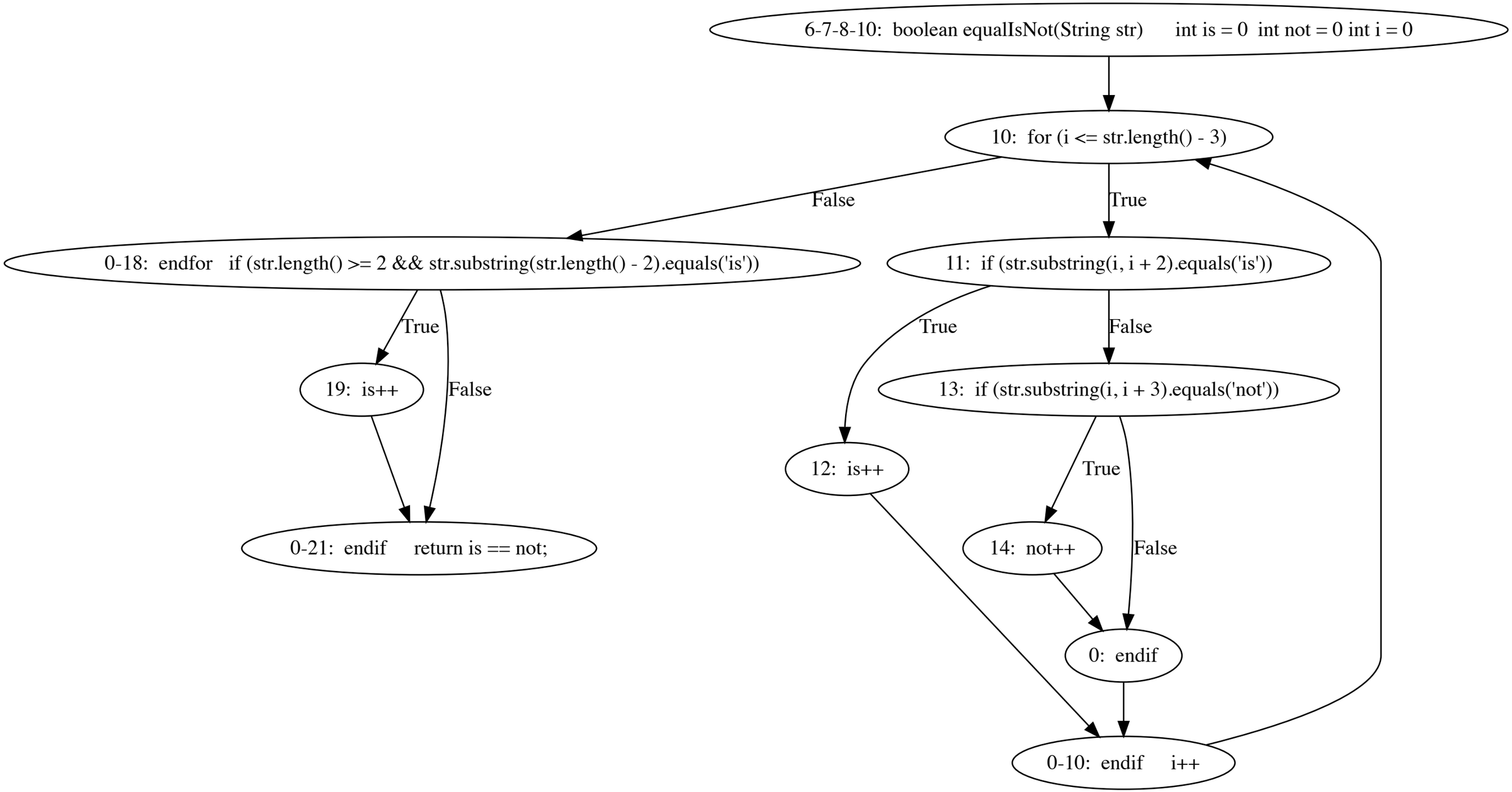

9-10-12: int scoreUp(String[] key, String[] answers) int score = 0 int i = 0
12: for (i < key.length)
False
True
0-20: endfor return score;
13: if (key[i].charAt(0) == answers[i].charAt(0))
True
False
15: if (answers[i].charAt(0) != '?')
14: score += 4
True
16: score -= 1
False
0: endif
0-12: endif i++

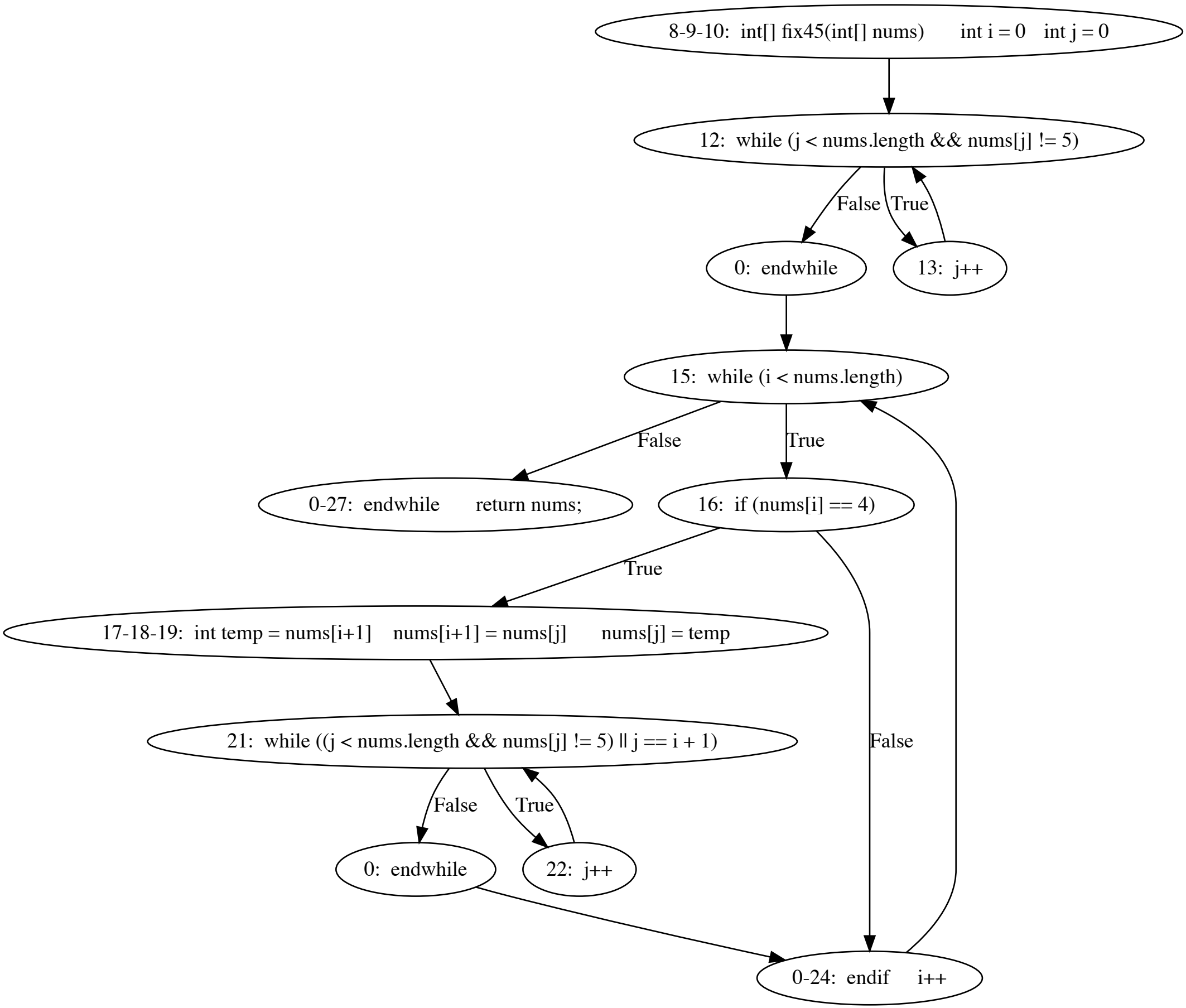

7-8-9-11: String getSandwich(String str) int first = -1 int last = -1 int i = 0
11: for (i < str.length() - 5)
True
12: if (str.substring(i, i + 5).equals('bread'))
False
True
False
13-14: first = i break;
0-11: endif i++
0-18: endfor int i = str.length() - 5
18: for (i >= 0)
True
19: if (str.substring(i, i + 5).equals('bread'))
False
True
False
20-21: last = i break;
0-18: endif i--
0-25: endfor if (first != -1 && last != -1 && first != last)
True
False
26: return str.substring(first + 5, last);
0-28: endif return '';

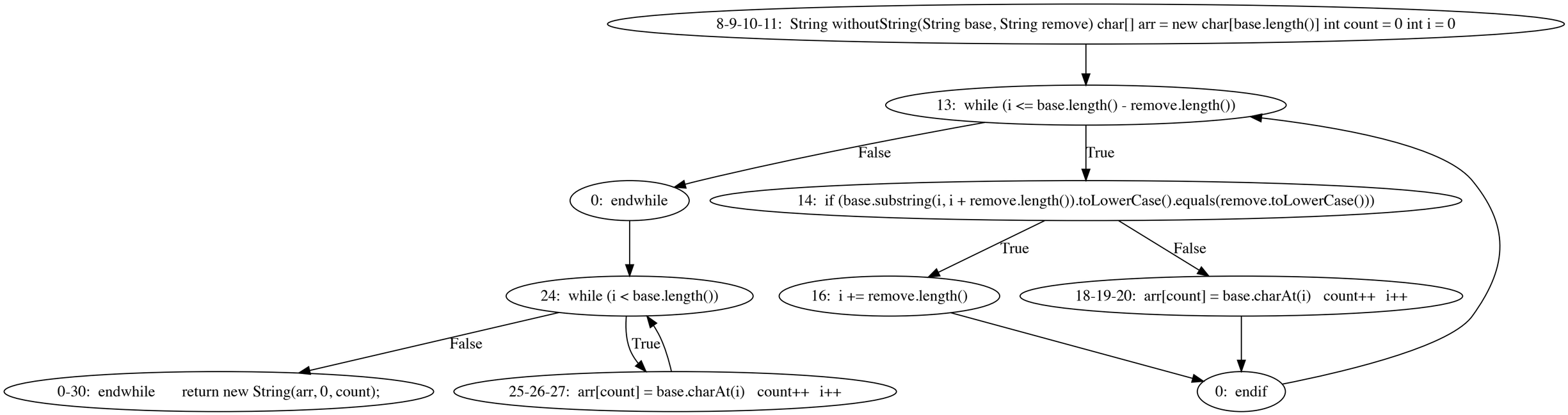

7-8-10: int[] notAlone(int[] nums, int val) int[] arr = new int[nums.length] if (nums.length >= 1)
True
False
11-12: arr[0] = nums[0] arr[arr.length-1] = nums[nums.length-1]
0-15: endif int i = 1
15: for (i <= nums.length - 2)
False
True
0-22: endfor return arr;
16: if (nums[i] == val && nums[i] != nums[i-1] && nums[i] != nums[i+1])
True
False
17: arr[i] = Math.max(nums[i-1], nums[i+1])
19: arr[i] = nums[i]
0-15: endif i++

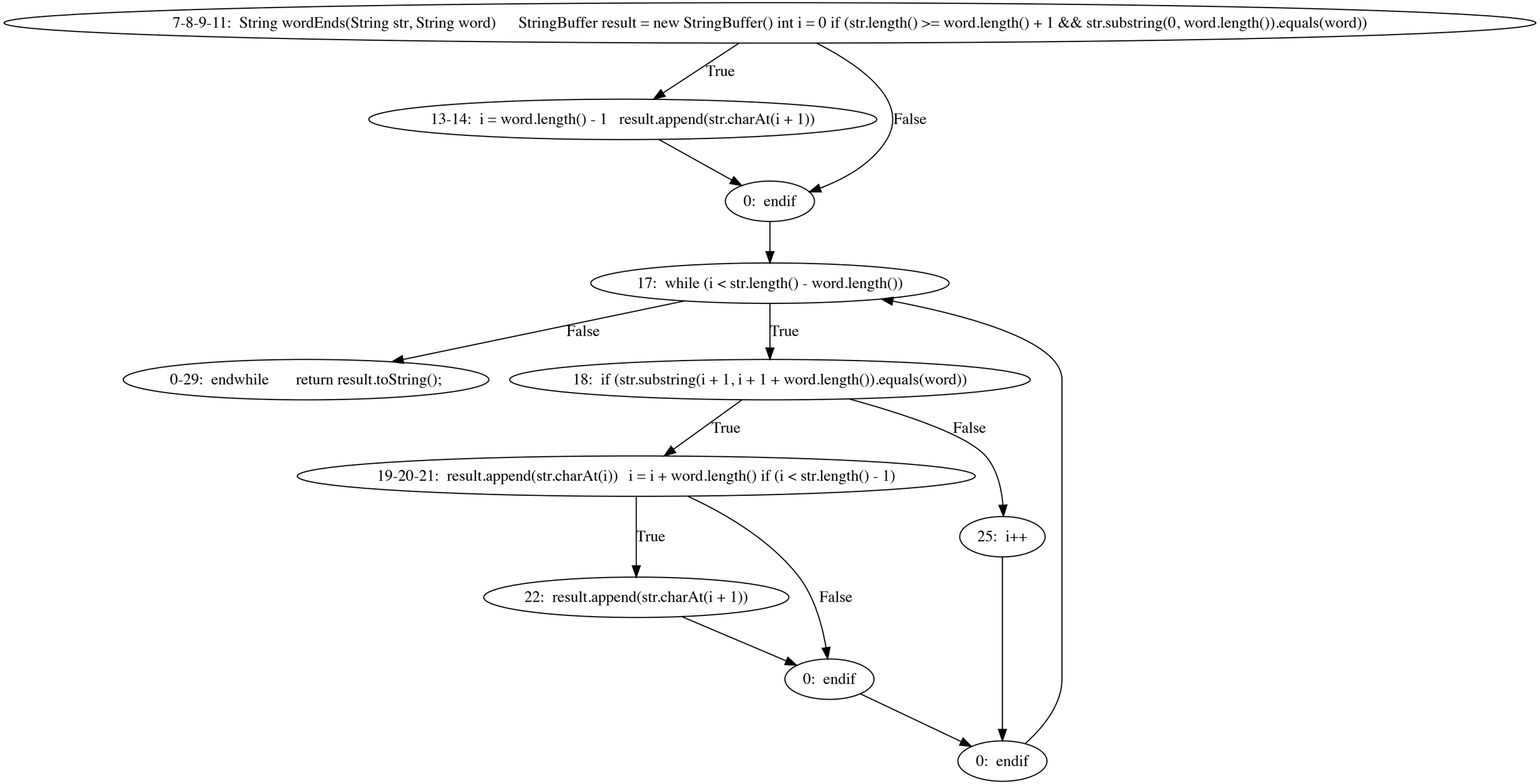

7-8: int countYZ(String str) if (str.length() == 0)
True
False
9: return 0;
0-11-13: endifint count = 0 int i = 0
13: for (i <= str.length() - 2)
False
True
0-20: endfor if (Character.toLowerCase(str.charAt(str.length() - 1)) == 'y' || Character.toLowerCase(str.charAt(str.length() - 1)) == 'z')
14: if ((Character.toLowerCase(str.charAt(i)) == 'y' || Character.toLowerCase(str.charAt(i)) == 'z') && !Character.isLetter(str.charAt(i+1)))
True
True
False
22: count++
17: count++
False
0-24: endif return count;
0-13: endif i++

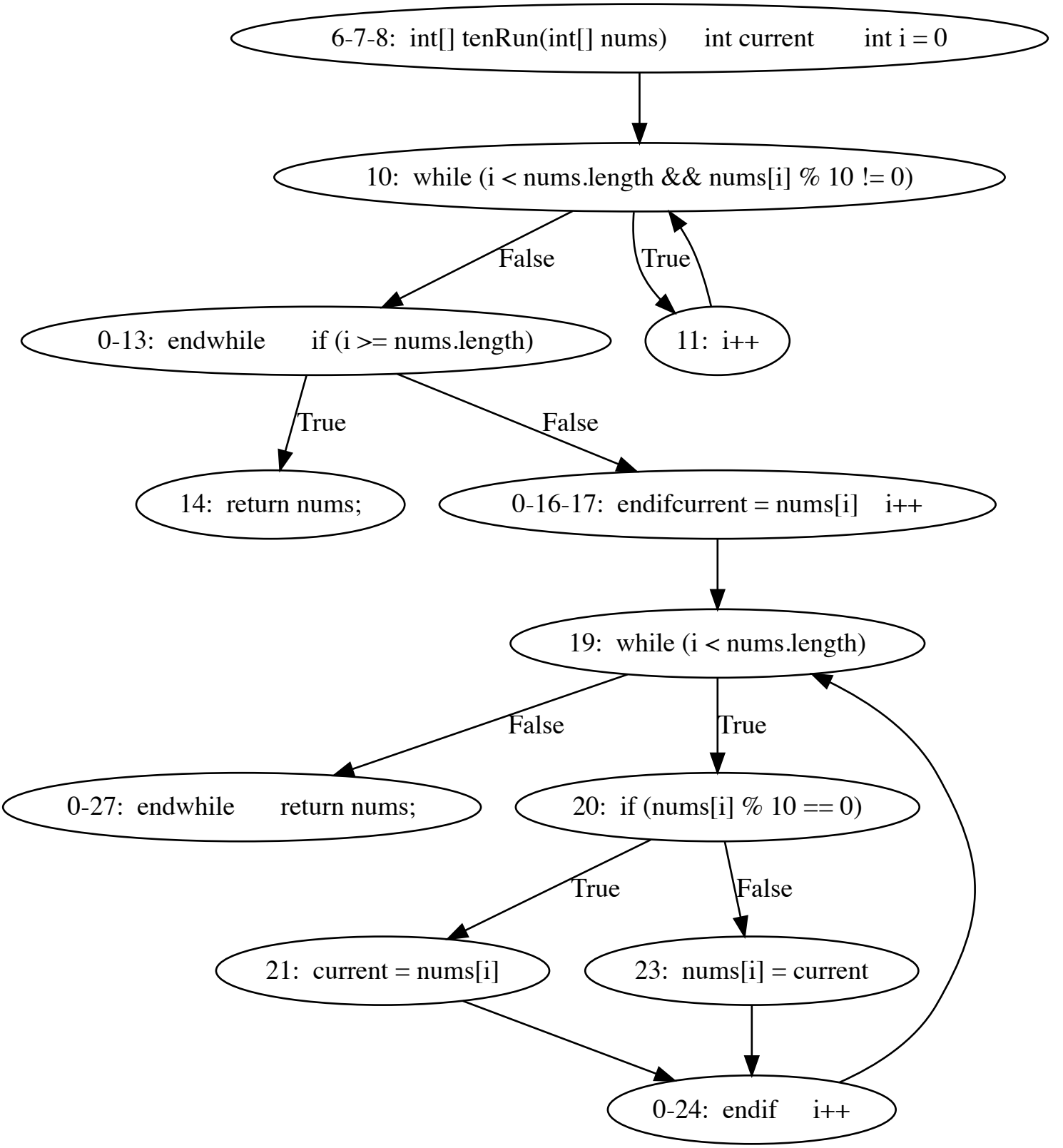

6-7-8-10: int sum67(int[] nums) int sum = 0 boolean inRange = false int i = 0
10: for (i < nums.length)
False
True
0-22: endfor return sum;
11: if (nums[i] == 6)
True
False
12: inRange = true
0-14: endif if (!inRange)
True
False
15: sum += nums[i]
0-17: endif if (inRange && nums[i] == 7)
True
False
18: inRange = false
0-10: endif i++

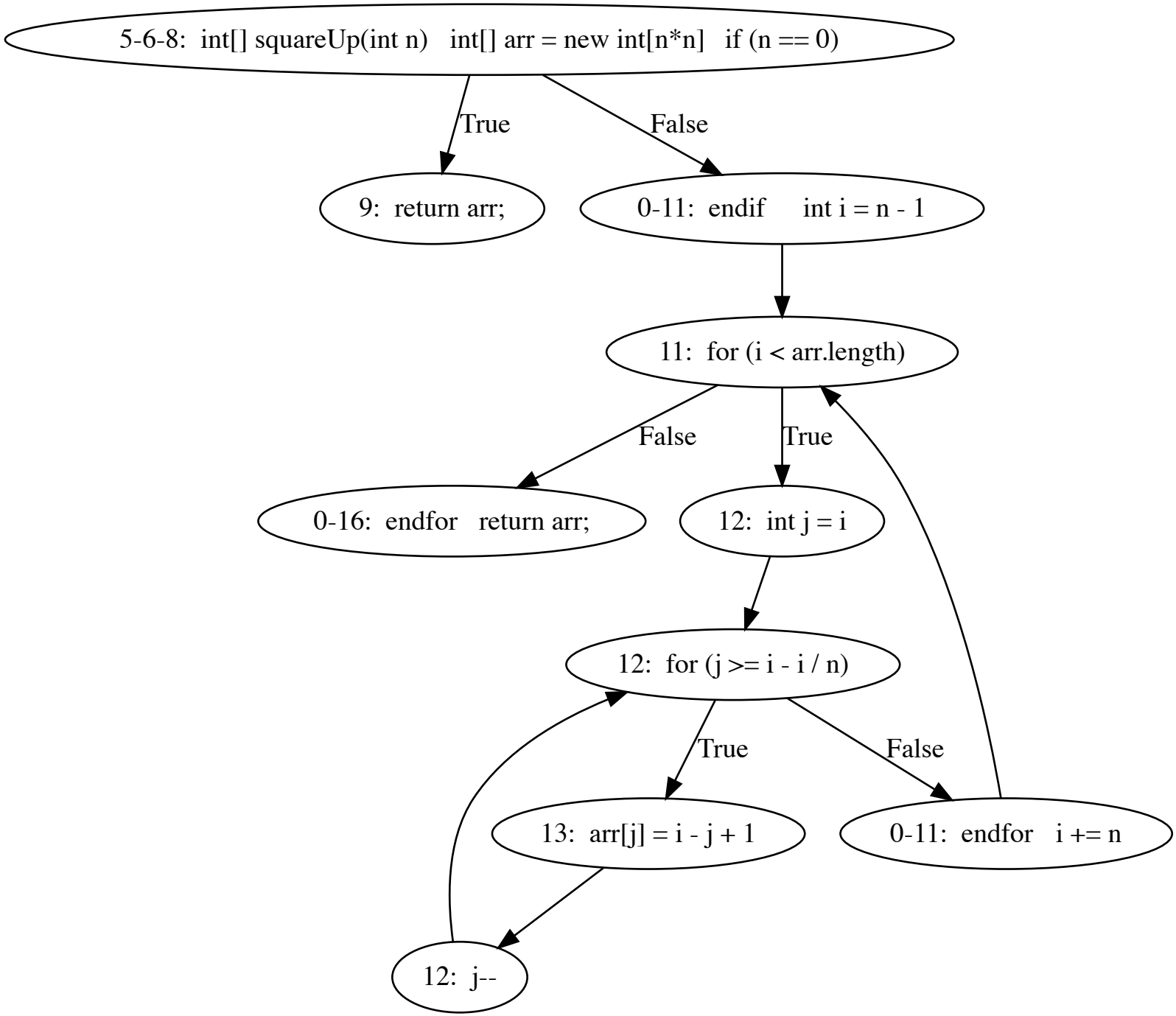

6-7-8-10: boolean equalIsNot(String str) int is = 0 int not = 0 int i = 0
10: for (i <= str.length() - 3)
False
True
0-18: endfor if (str.length() >= 2 && str.substring(str.length() - 2).equals('is'))
11: if (str.substring(i, i + 2).equals('is'))
True
False
True
False
19: is++
13: if (str.substring(i, i + 3).equals('not'))
12: is++
True
False
14: not++
0: endif
0-21: endif return is == not;
0-10: endif i++

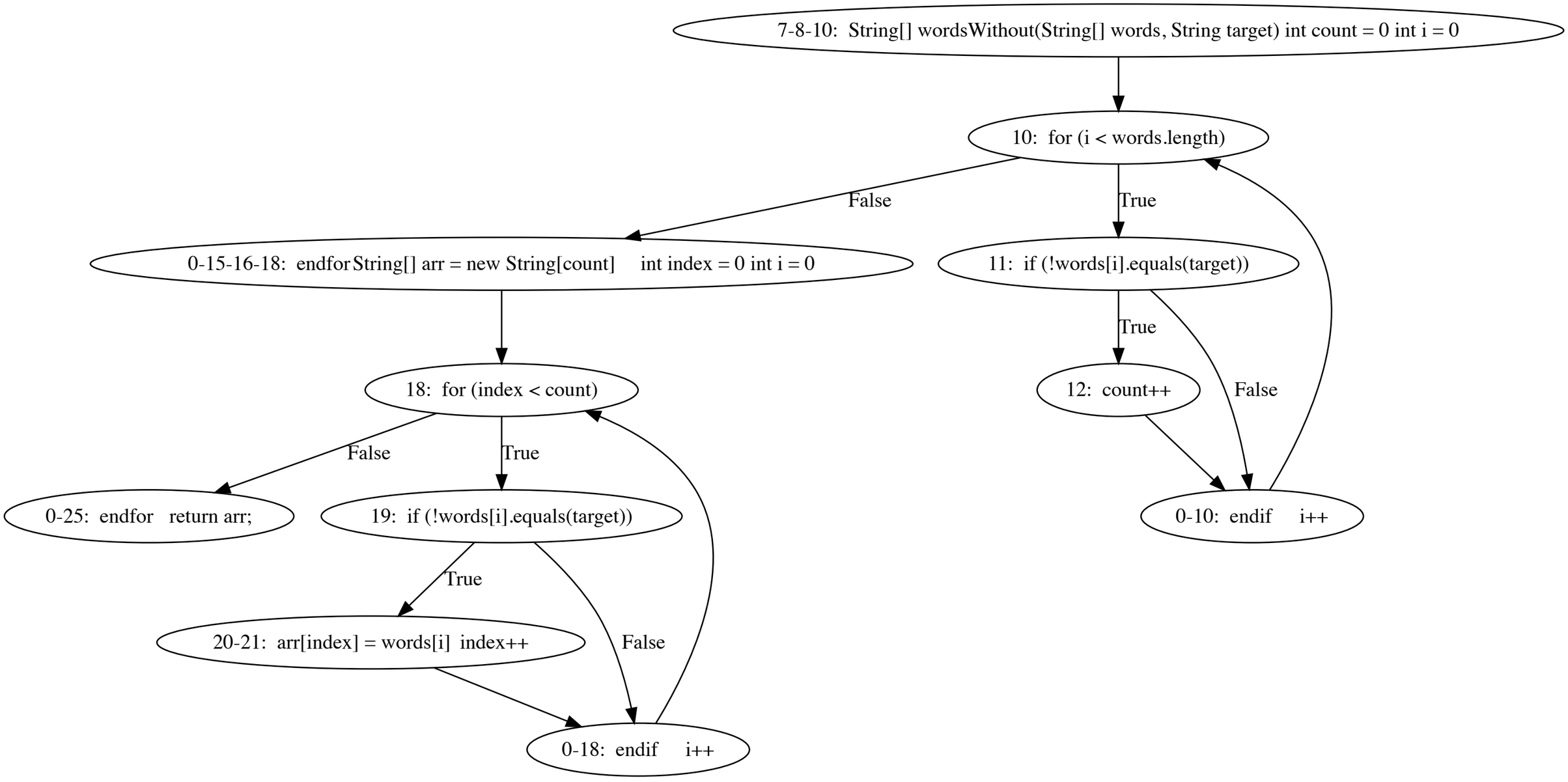

8-9-10: int[] fix45(int[] nums) int i = 0 int j = 0
12: while (j < nums.length && nums[j] != 5)
False
True
0: endwhile
13: j++
15: while (i < nums.length)
False
True
0-27: endwhile return nums;
16: if (nums[i] == 4)
True
False
17-18-19: int temp = nums[i+1] nums[i+1] = nums[j] nums[j] = temp
21: while ((j < nums.length && nums[j] != 5) || j == i + 1)
False
True
0: endwhile
22: j++
0-24: endif i++

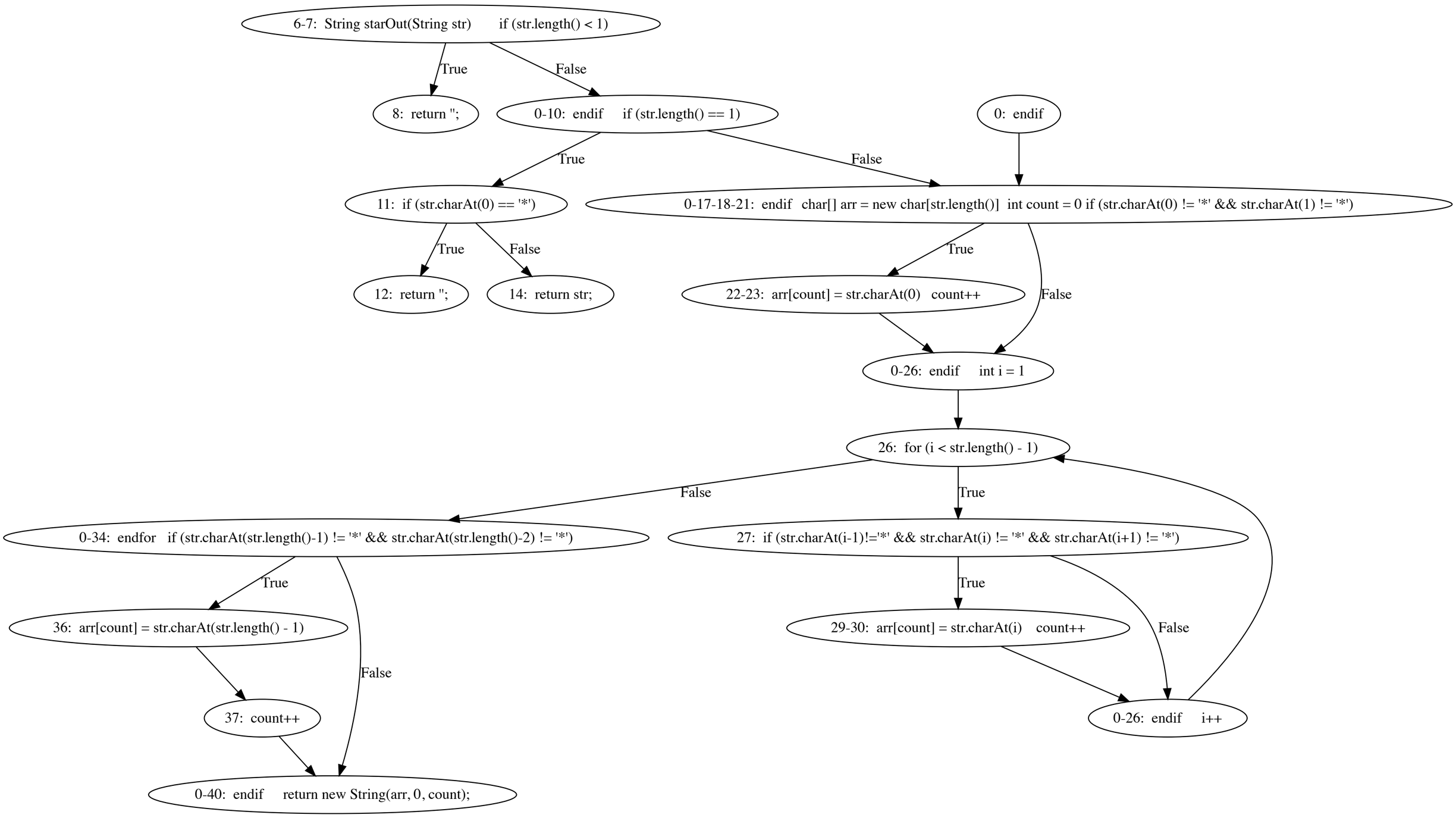

8-9-10-11: String withoutString(String base, String remove) char[] arr = new char[base.length()] int count = 0 int i = 0
13: while (i <= base.length() - remove.length())
False
True
0: endwhile
14: if (base.substring(i, i + remove.length()).toLowerCase().equals(remove.toLowerCase()))
True
False
24: while (i < base.length())
16: i += remove.length()
18-19-20: arr[count] = base.charAt(i) count++ i++
False
True
0-30: endwhile return new String(arr, 0, count);
25-26-27: arr[count] = base.charAt(i) count++ i++
0: endif

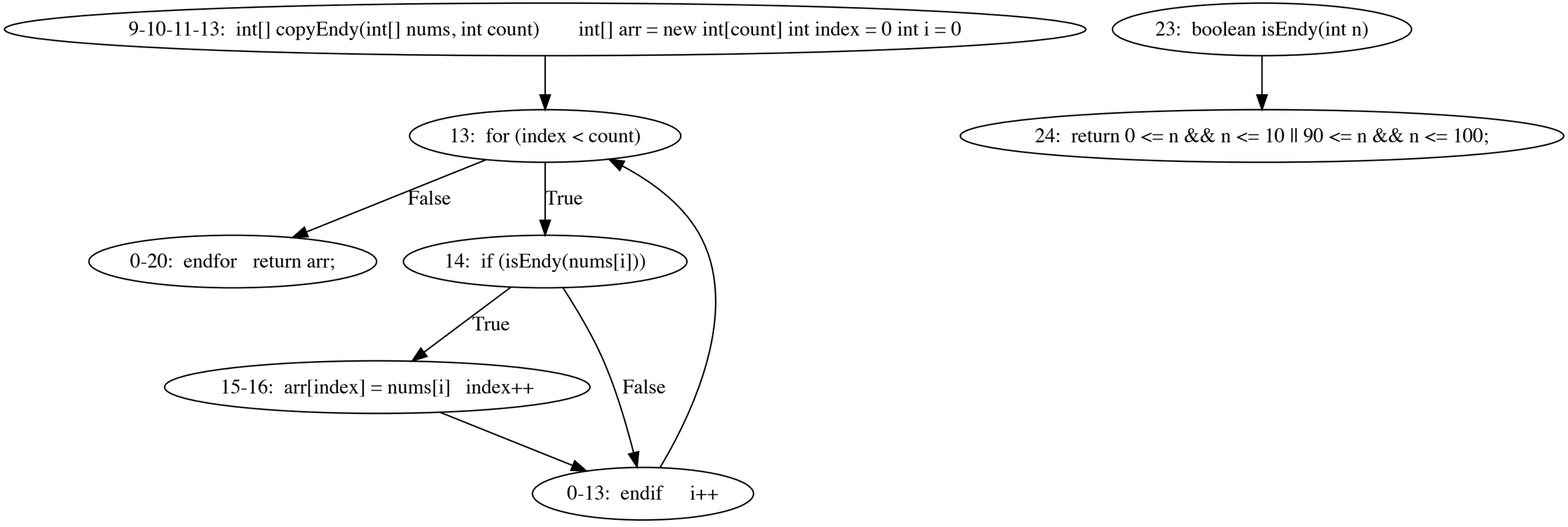

7-8-9-11: String wordEnds(String str, String word) StringBuffer result = new StringBuffer() int i = 0 if (str.length() >= word.length() + 1 && str.substring(0, word.length()).equals(word))
True
False
13-14: i = word.length() - 1 result.append(str.charAt(i + 1))
0: endif
17: while (i < str.length() - word.length())
False
True
0-29: endwhile return result.toString();
18: if (str.substring(i + 1, i + 1 + word.length()).equals(word))
True
False
19-20-21: result.append(str.charAt(i)) i = i + word.length() if (i < str.length() - 1)
True
25: i++
22: result.append(str.charAt(i + 1))
False
0: endif
0: endif

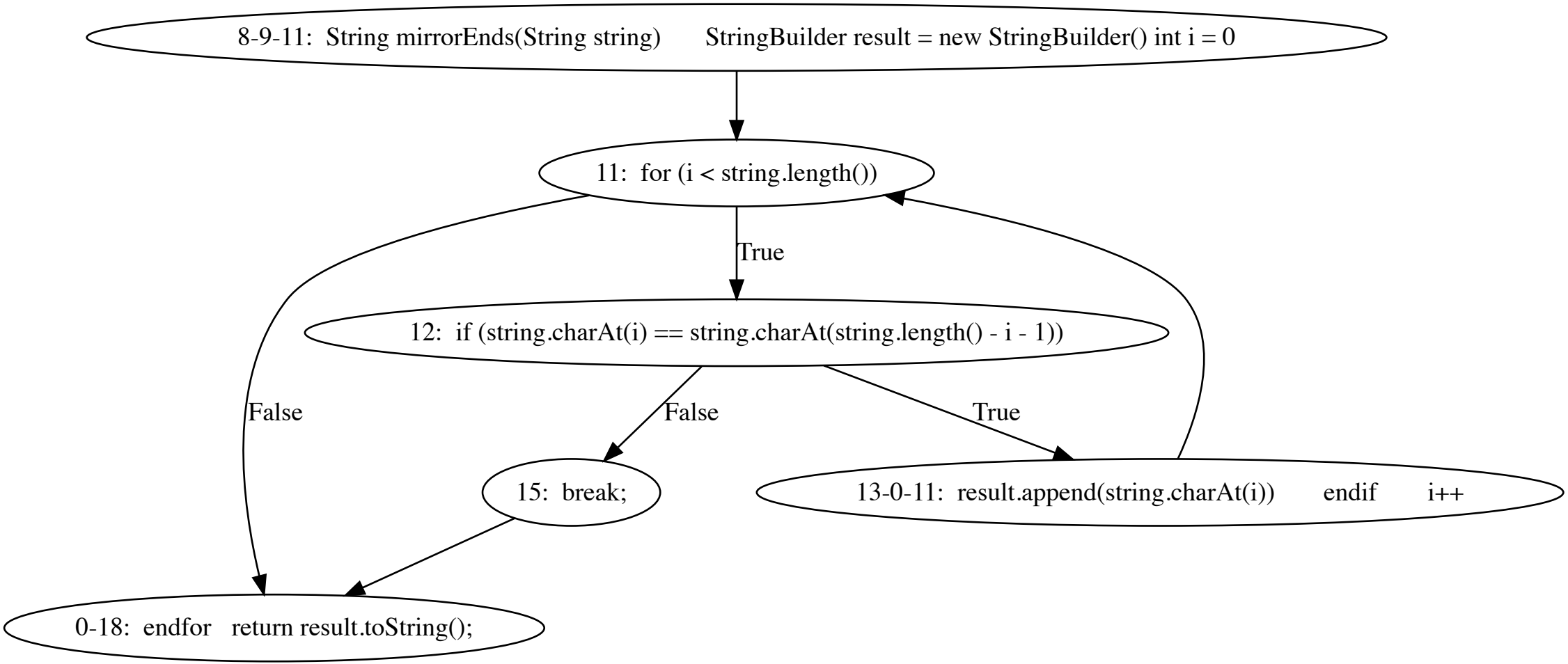

6-7-8: int[] tenRun(int[] nums) int current int i = 0
10: while (i < nums.length && nums[i] % 10 != 0)
False
True
0-13: endwhile if (i >= nums.length)
11: i++
True
False
14: return nums;
0-16-17: endifcurrent = nums[i] i++
19: while (i < nums.length)
False
True
0-27: endwhile return nums;
20: if (nums[i] % 10 == 0)
True
False
21: current = nums[i]
23: nums[i] = current
0-24: endif i++


5-6-8: int[] squareUp(int n) int[] arr = new int[n*n] if (n == 0)
True
False
9: return arr;
0-11: endif int i = n - 1
11: for (i < arr.length)
False
True
0-16: endfor return arr;
12: int j = i
12: for (j >= i - i / n)
True
False
13: arr[j] = i - j + 1
0-11: endfor i += n
12: j--

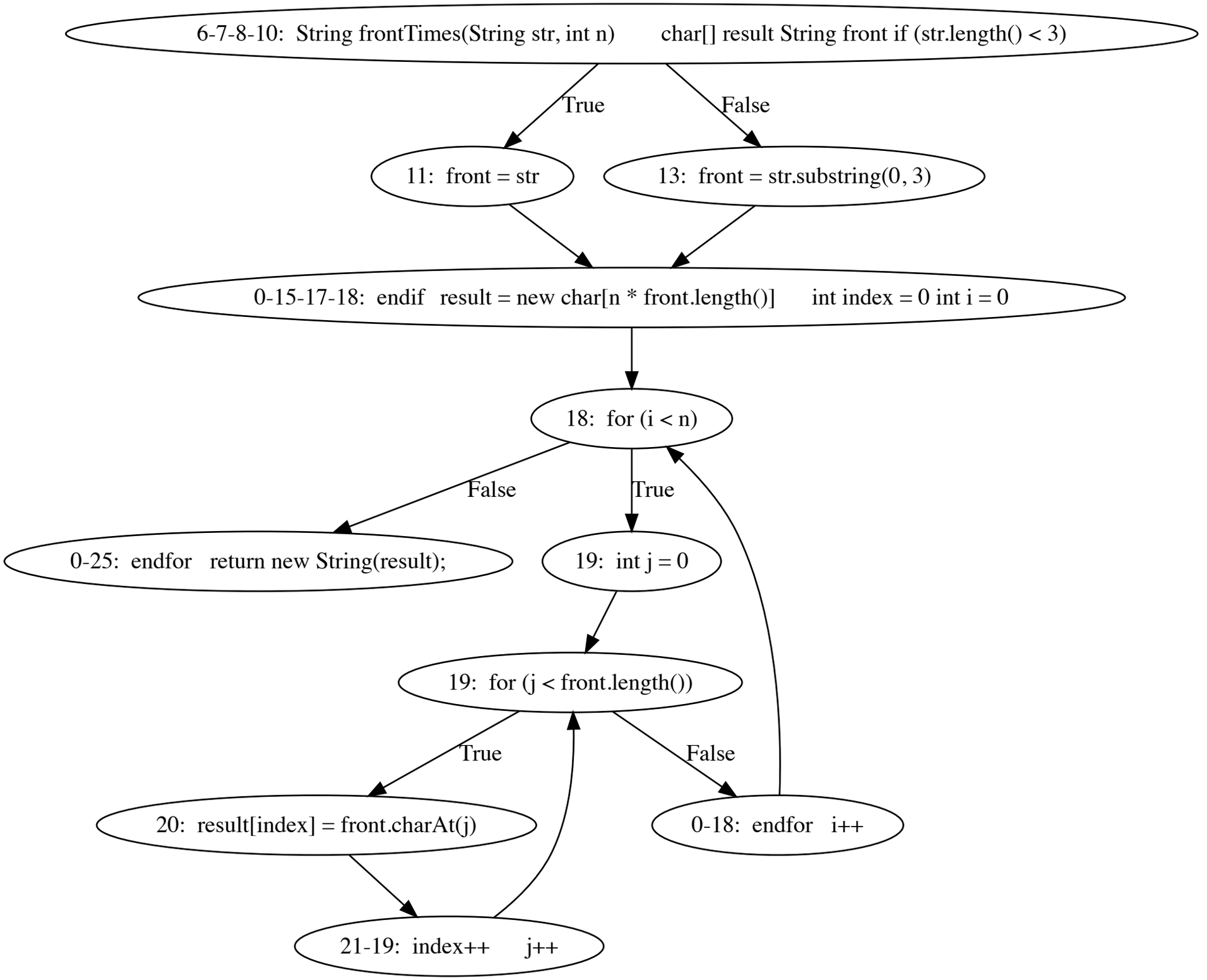

7-8-10: String[] wordsWithout(String[] words, String target) int count = 0 int i = 0
10: for (i < words.length)
False
True
0-15-16-18: endfor String[] arr = new String[count] int index = 0 int i = 0
11: if (!words[i].equals(target))
True
18: for (index < count)
12: count++
False
False
True
0-25: endfor return arr;
19: if (!words[i].equals(target))
0-10: endif i++
True
20-21: arr[index] = words[i] index++
False
0-18: endif i++

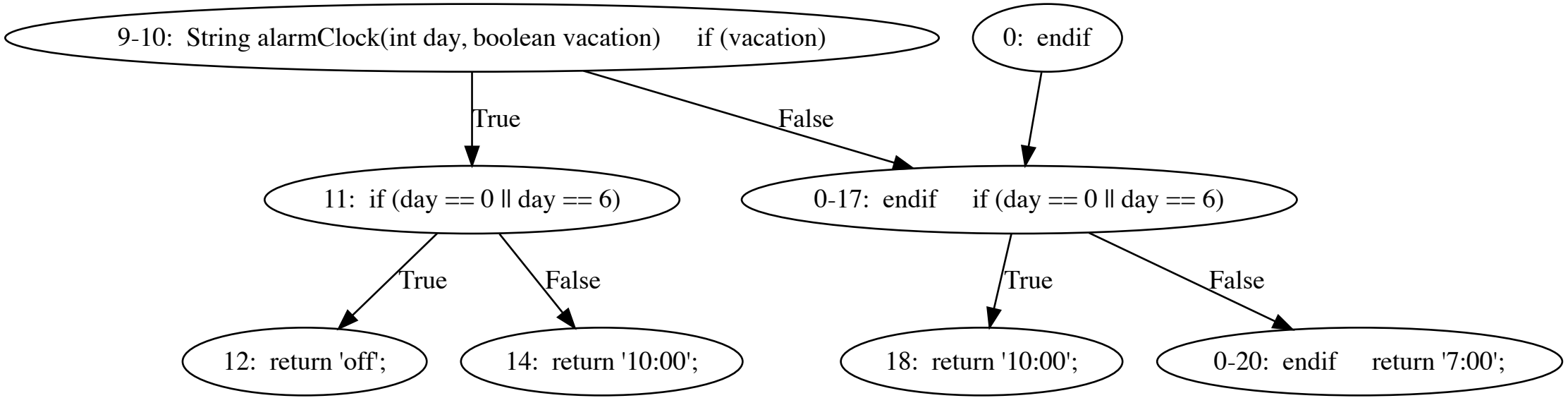

6-7: String starOut(String str) if (str.length() < 1)
True
False
8: return '';
0-10: endif if (str.length() == 1)
0: endif
True
False
11: if (str.charAt(0) == '*')
0-17-18-21: endif char[] arr = new char[str.length()] int count = 0 if (str.charAt(0) != '*' && str.charAt(1) != '*')
True
False
12: return '';
14: return str;
True
22-23: arr[count] = str.charAt(0) count++
False
0-26: endif int i = 1
26: for (i < str.length() - 1)
False
True
0-34: endfor if (str.charAt(str.length()-1) != '*' && str.charAt(str.length()-2) != '*')
27: if (str.charAt(i-1)!='*' && str.charAt(i) != '*' && str.charAt(i+1) != '*')
True
True
36: arr[count] = str.charAt(str.length() - 1)
29-30: arr[count] = str.charAt(i) count++
False
False
37: count++
0-26: endif i++
0-40: endif return new String(arr, 0, count);

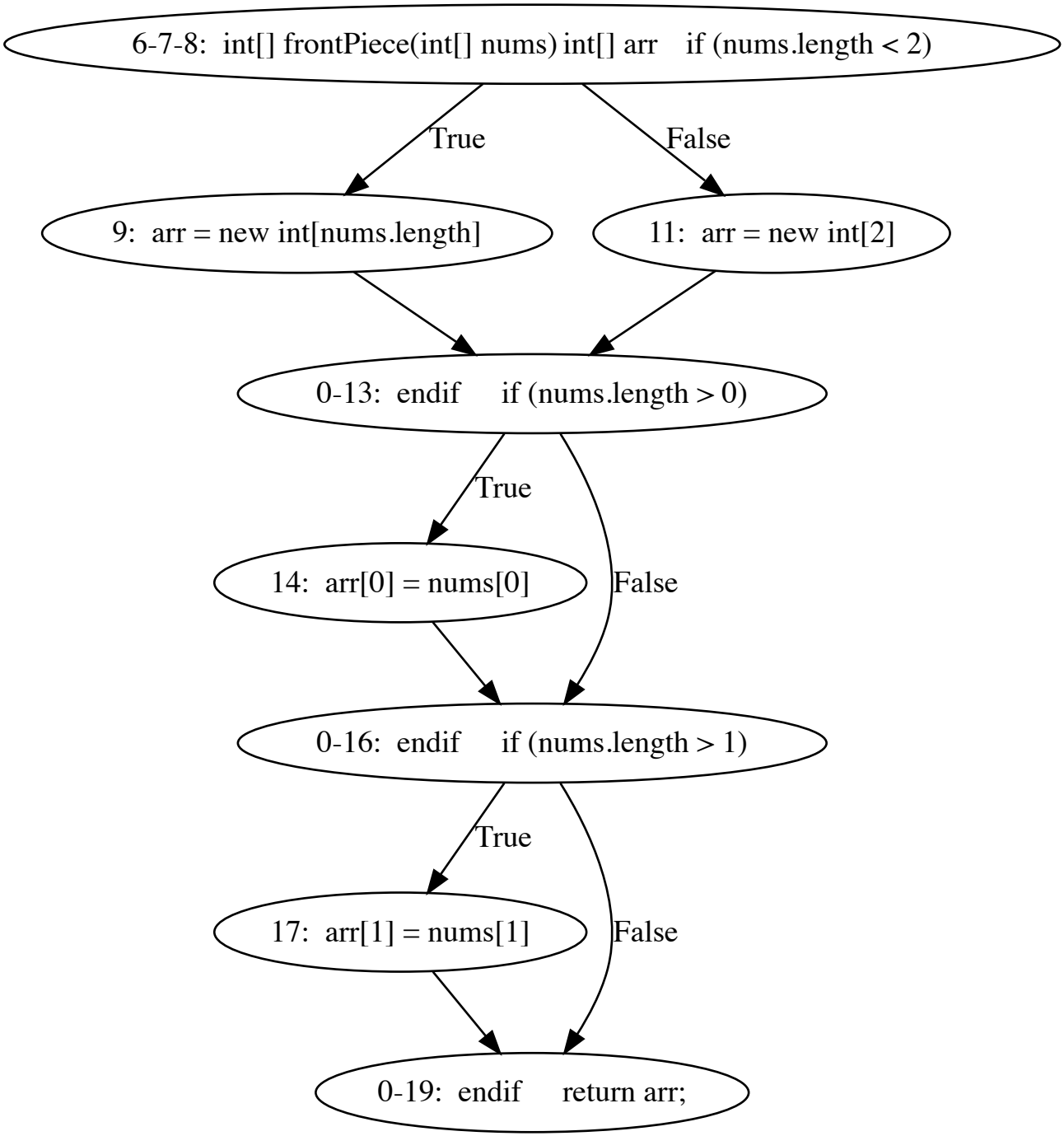

9-10-11-13: int[] copyEndy(int[] nums, int count) int[] arr = new int[count] int index = 0 int i = 0
23: boolean isEndy(int n)
13: for (index < count)
24: return 0 <= n && n <= 10 || 90 <= n && n <= 100;
False
True
0-20: endfor return arr;
14: if (isEndy(nums[i]))
True
15-16: arr[index] = nums[i] index++
False
0-13: endif i++

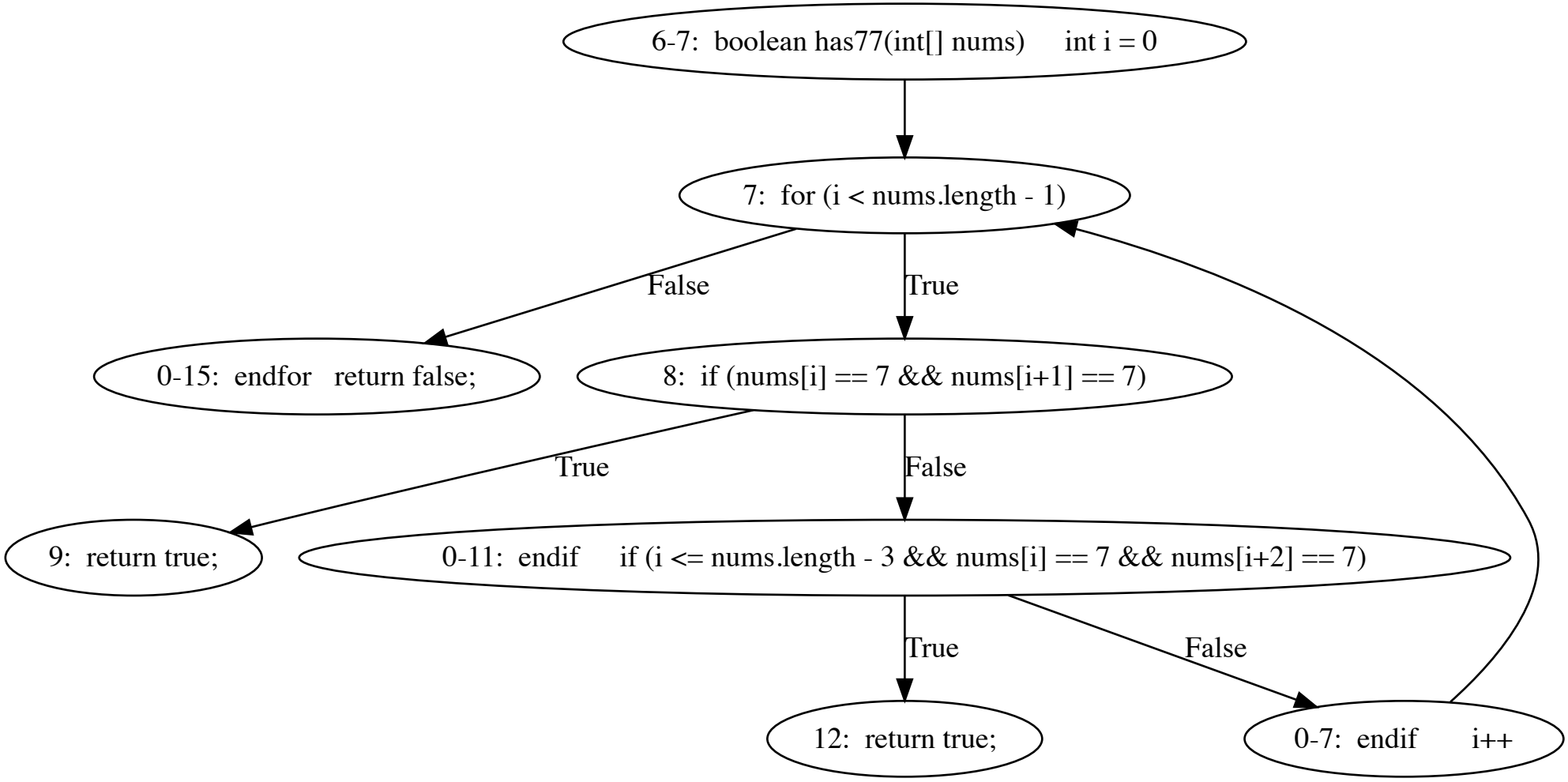

8-9-11: String mirrorEnds(String string) StringBuilder result = new StringBuilder() int i = 0
11: for (i < string.length())
True
12: if (string.charAt(i) == string.charAt(string.length() - i - 1))
False
False
True
15: break;
13-0-11: result.append(string.charAt(i)) endif i++
0-18: endfor return result.toString();

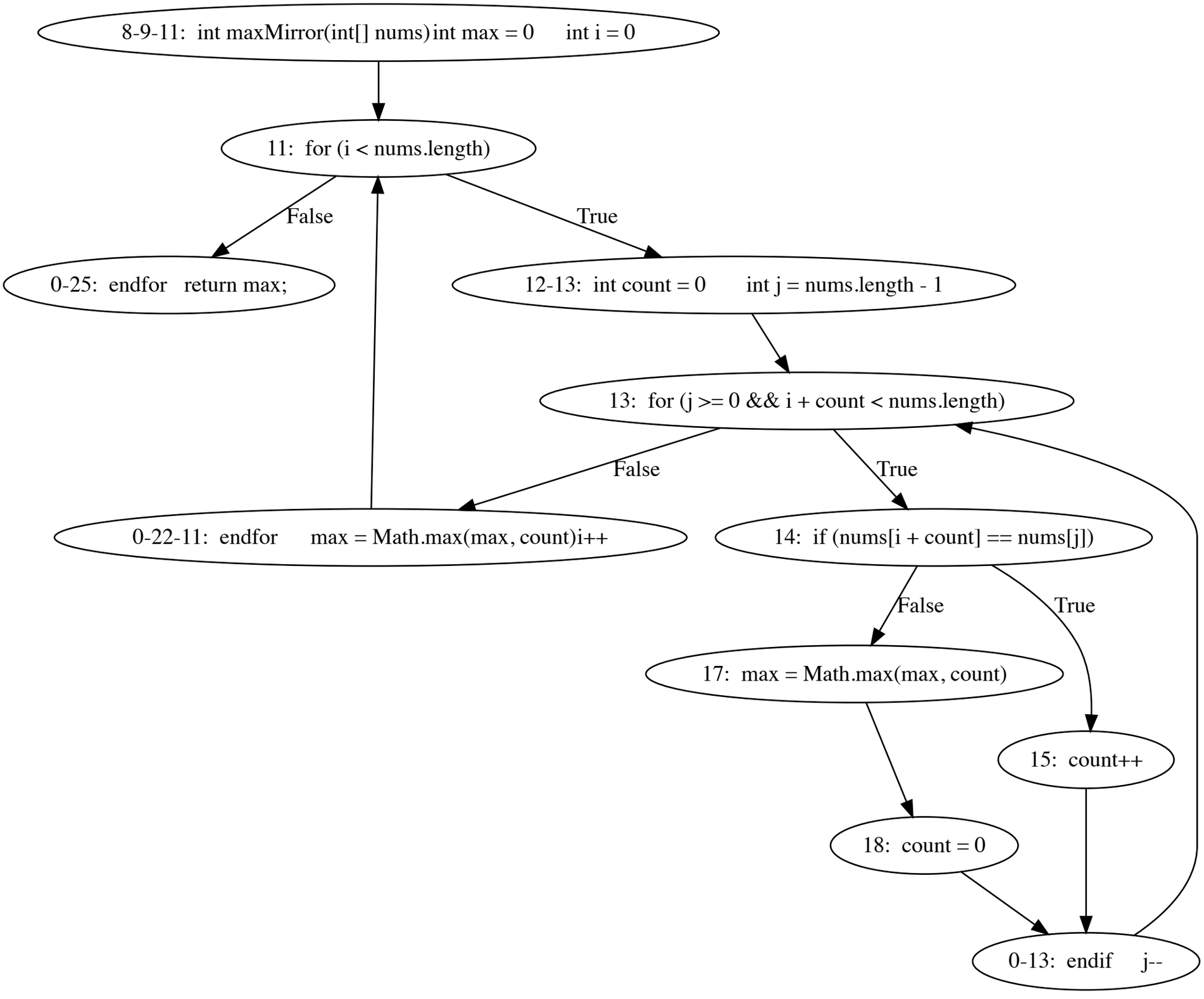
7-8: boolean closeFar(int a, int b, int c)return (isClose(a, b) && isFar(a, b, c)) || (isClose(a, c) && isFar(a, c, b));
12-13: boolean isClose(int a, int b) return Math.abs(a - b) <= 1;
16-17: boolean isFar(int a, int b, int c) return Math.abs(a - c) >= 2 && Math.abs(b - c) >= 2;

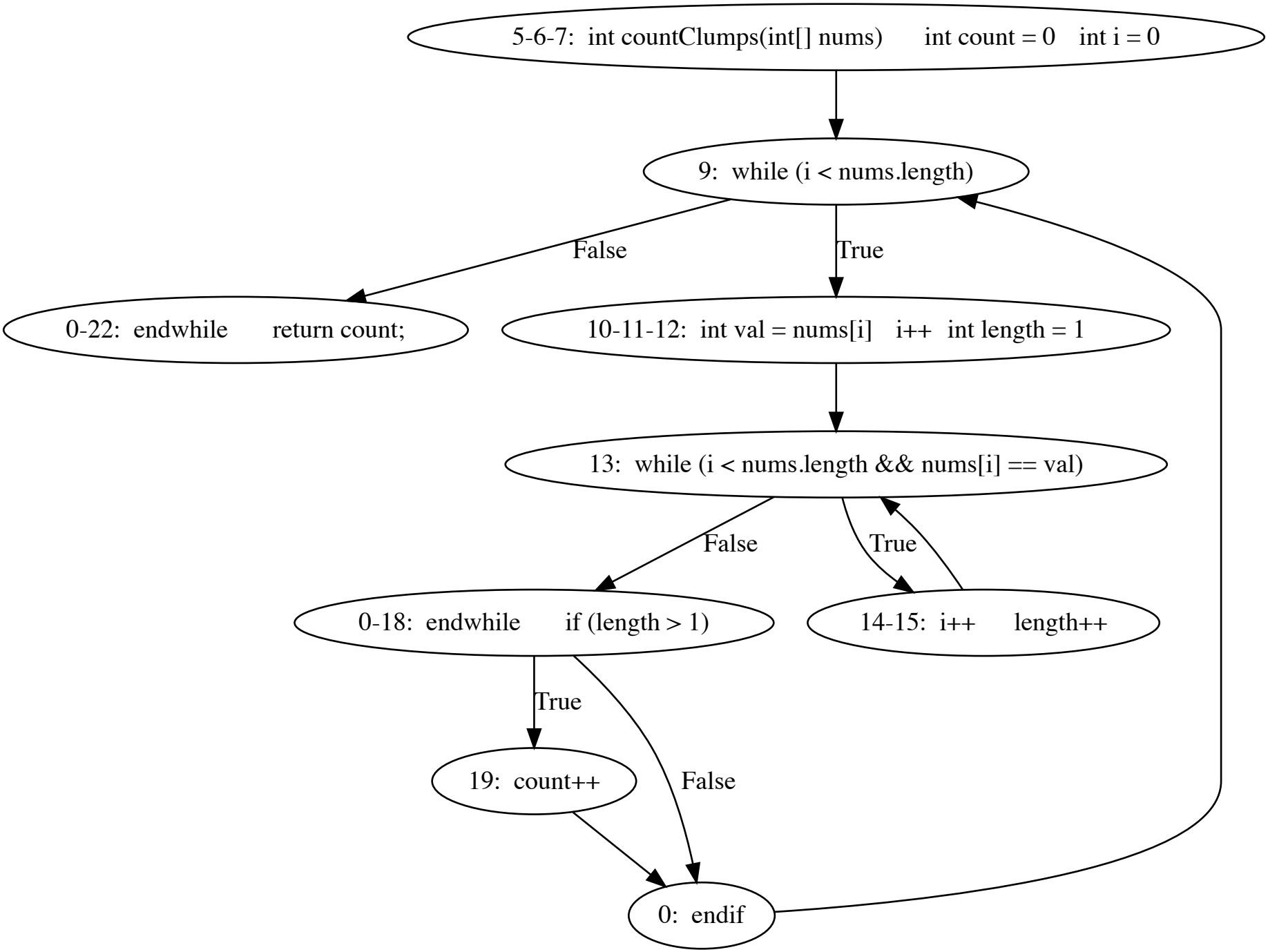

6-7-8-10: String frontTimes(String str, int n) char[] result String front if (str.length() < 3)
True
False
11: front = str
13: front = str.substring(0, 3)
0-15-17-18: endif result = new char[n * front.length()] int index = 0 int i = 0
18: for (i < n)
False
True
0-25: endfor return new String(result);
19: int j = 0
19: for (j < front.length())
True
False
20: result[index] = front.charAt(j)
0-18: endfor i++
21-19: index++ j++

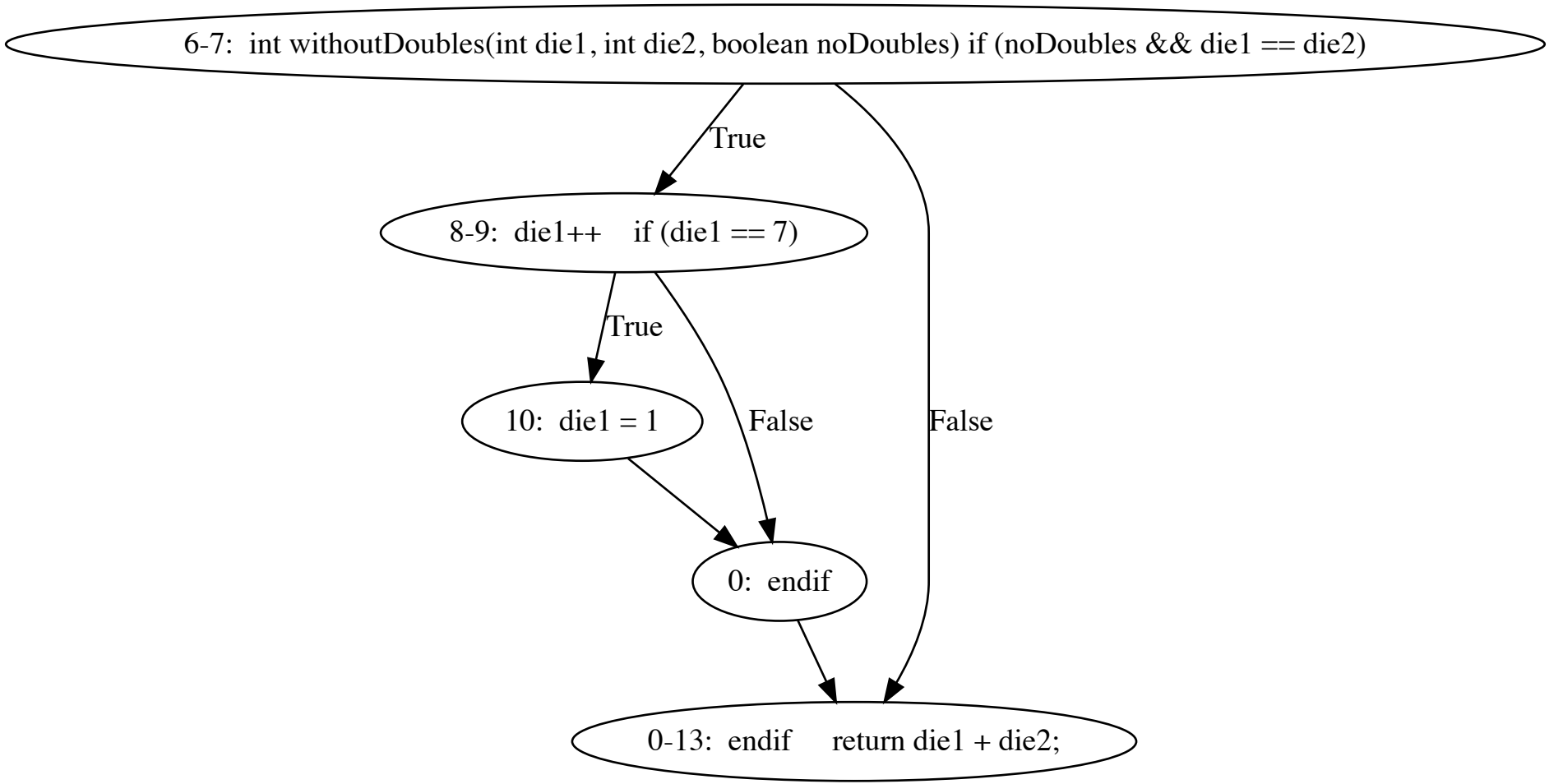

9-10: String alarmClock(int day, boolean vacation) if (vacation)
0: endif
True
False
11: if (day == 0 || day == 6)
0-17: endif if (day == 0 || day == 6)
True
False
True
False
12: return 'off';
14: return '10:00';
18: return '10:00';
0-20: endif return '7:00';

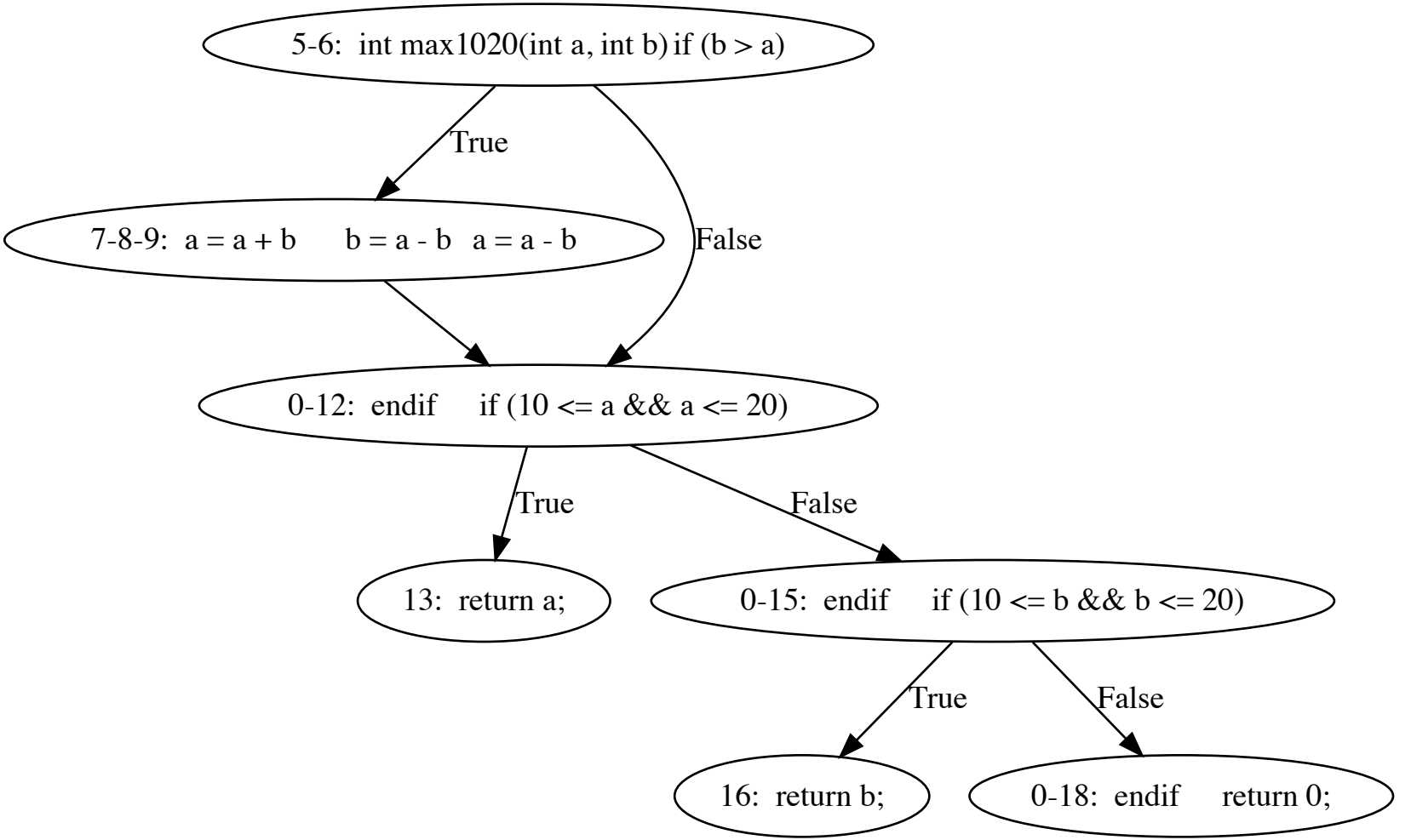

6-7-8: int[] frontPiece(int[] nums) int[] arr if (nums.length < 2)
True
False
9: arr = new int[nums.length]
11: arr = new int[2]
0-13: endif if (nums.length > 0)
True
False
14: arr[0] = nums[0]
0-16: endif if (nums.length > 1)
True
False
17: arr[1] = nums[1]
0-19: endif return arr;

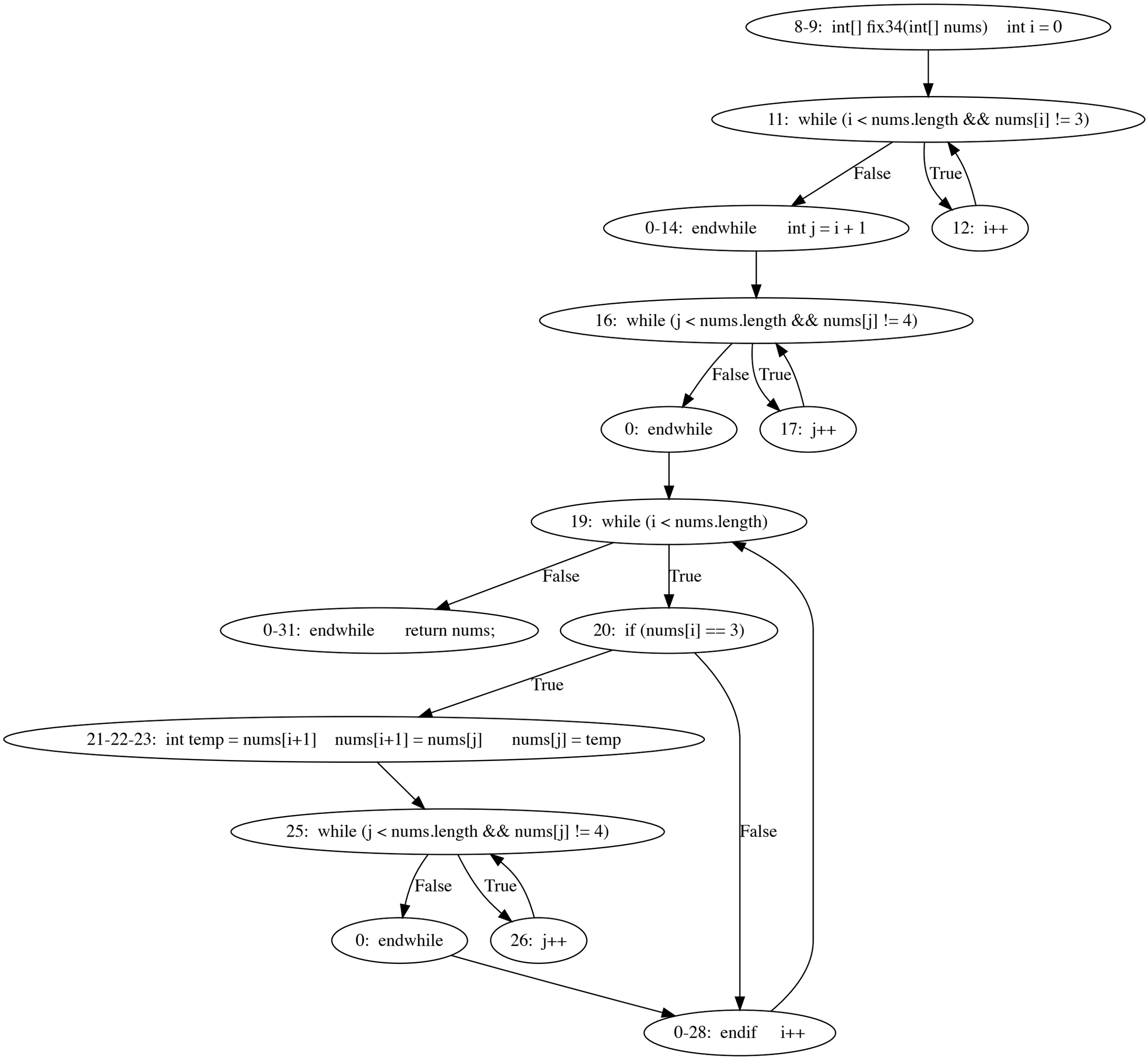

6-7: boolean has77(int[] nums) int i = 0
7: for (i < nums.length - 1)
False
True
0-15: endfor return false;
8: if (nums[i] == 7 && nums[i+1] == 7)
True
False
9: return true;
0-11: endif if (i <= nums.length - 3 && nums[i] == 7 && nums[i+2] == 7)
True
False
12: return true;
0-7: endif i++

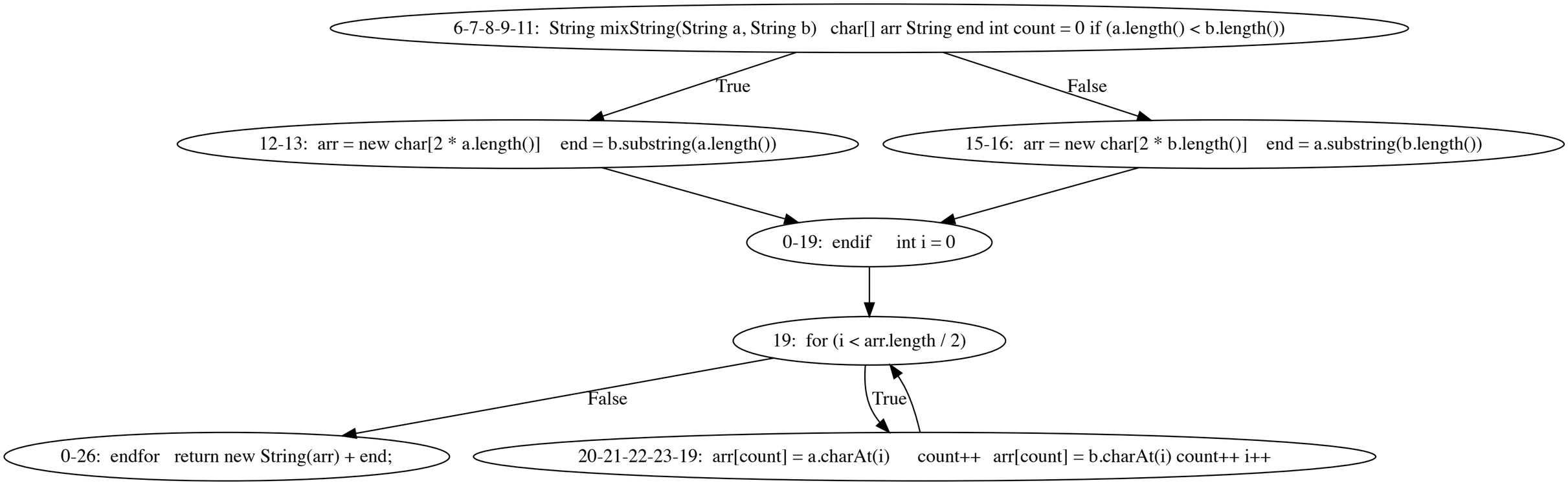

8-9-11: int maxMirror(int[] nums)int max = 0 int i = 0
11: for (i < nums.length)
False
True
0-25: endfor return max;
12-13: int count = 0 int j = nums.length - 1
13: for (j >= 0 && i + count < nums.length)
False
True
0-22-11: endfor max = Math.max(max, count)i++
14: if (nums[i + count] == nums[j])
False
True
17: max = Math.max(max, count)
15: count++
18: count = 0
0-13: endif j--


5-6-7: int countClumps(int[] nums) int count = 0 int i = 0
9: while (i < nums.length)
False
True
0-22: endwhile return count;
10-11-12: int val = nums[i] i++ int length = 1
13: while (i < nums.length && nums[i] == val)
False
True
0-18: endwhile if (length > 1)
14-15: i++ length++
True
19: count++
False
0: endif


6-7: int withoutDoubles(int die1, int die2, boolean noDoubles) if (noDoubles && die1 == die2)
True
8-9: die1++ if (die1 == 7)
True
10: die1 = 1
False
False
0: endif
0-13: endif return die1 + die2;


5-6: int max1020(int a, int b) if (b > a)
True
False
7-8-9: a = a + b b = a - b a = a - b
0-12: endif if (10 <= a && a <= 20)
True
False
13: return a;
0-15: endif if (10 <= b && b <= 20)
True
False
16: return b;
0-18: endif return 0;


8-9: int[] fix34(int[] nums) int i = 0
11: while (i < nums.length && nums[i] != 3)
False
True
0-14: endwhile int j = i + 1
12: i++
16: while (j < nums.length && nums[j] != 4)
False
True
0: endwhile
17: j++
19: while (i < nums.length)
False
True
0-31: endwhile return nums;
20: if (nums[i] == 3)
True
False
21-22-23: int temp = nums[i+1] nums[i+1] = nums[j] nums[j] = temp
25: while (j < nums.length && nums[j] != 4)
False
True
0: endwhile
26: j++
0-28: endif i++


6-7-8-9-11: String mixString(String a, String b) char[] arr String end int count = 0 if (a.length() < b.length())
True
False
12-13: arr = new char[2 * a.length()] end = b.substring(a.length())
15-16: arr = new char[2 * b.length()] end = a.substring(b.length())
0-19: endif int i = 0
19: for (i < arr.length / 2)
False
True
0-26: endfor return new String(arr) + end;
20-21-22-23-19: arr[count] = a.charAt(i) count++ arr[count] = b.charAt(i) count++ i++