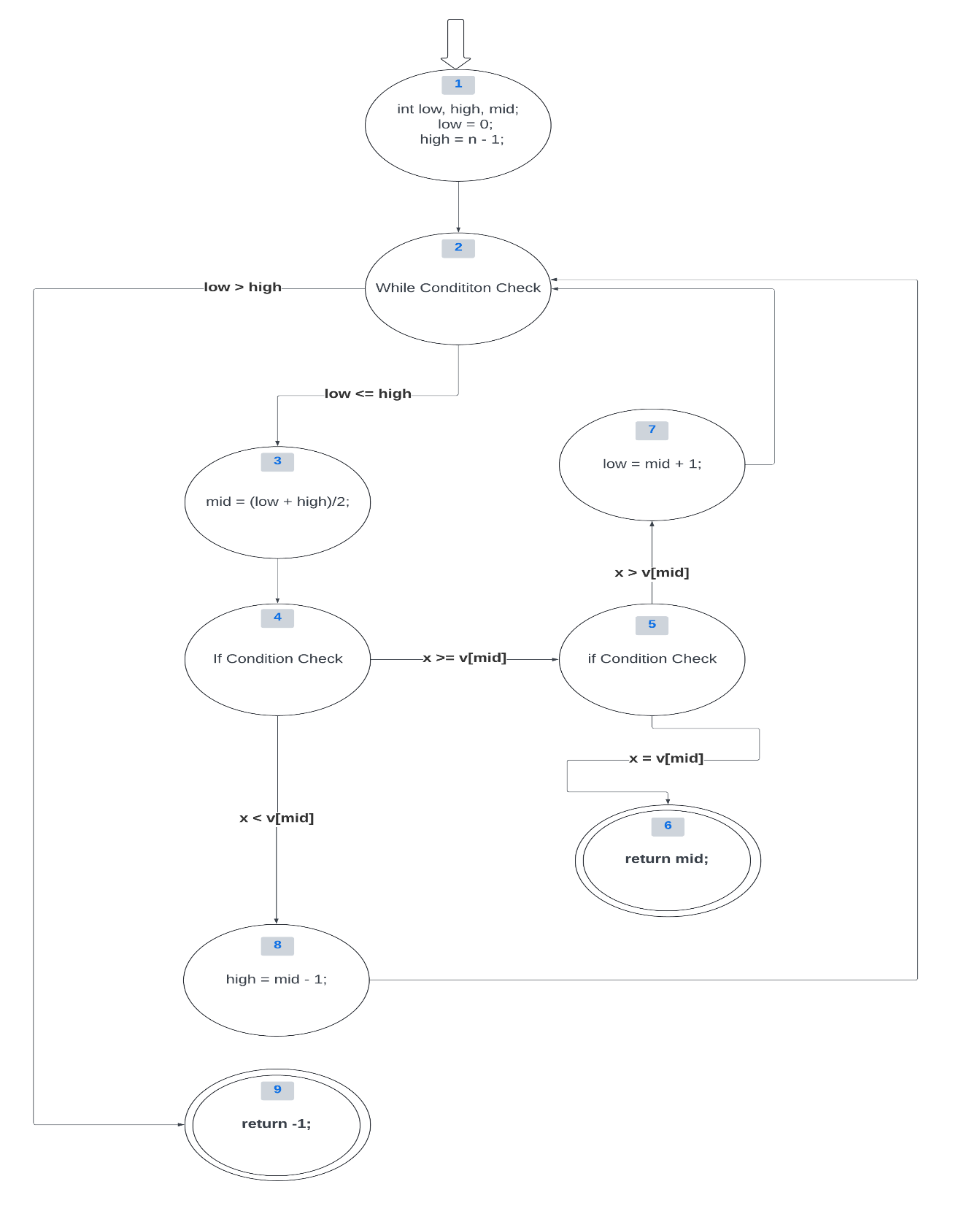
محمد مهدی نظری 9931061 – ابراهیم صدیقی 9931098

1)

الف)



ب)

پوشش گره: برای پوشش گره باید تست کیس هایی ارایه دهیم که همه گره ها پیمایش شده باشند. با فرض اینکه آرایه v به صورت صعودی مرتب شده باشد و متغیر x در آن حضور داشته باشد. برای اجرای همه حالات داخل حلقه باید تست کیس هایی داشته باشیم که x > v[mid] و x < v[mid] باشد ( در هر کدام از این دوحالت گره x == v[mid] اجرا خواهد شد). برای پیمایش گره آخر هم باید مقدار n صفر یا کمتر از آن باشد تا وارد حلقه نشود. سه تست کیس زیر باهم پوشش گره را ضمانت میکنند.

Test Case = { int[] v, int x, int n}

Test Case 1 = { [1, 2, 3], 1, 3} => nodes visited = {1, 2, 3, 4, 5, 6, 8}

Test Case 2 = { [1, 2, 3], 3, 3} => nodes visited = {1, 2, 3, 4, 5, 6, 7}

Test Case 3 = { [], 1, 0} => nodes visited = {1, 2, 9}

پوشش یال: برای پوشش یال باید همه ی یال ها پیمایش شوند. تست کیس های این مورد هم مانند قسمت پوشش گره است.

Test Case = { int[] v, int x, int n}

Test Case 1 = { [1, 2, 3], 1, 3} => edges visited = {[1,2], [2,3], [3,4], [4,8], [8,2], [4,5], [5,6]}

Test Case 2 = { [1, 2, 3], 3, 3} => edges visited = {[1,2], [2,3], [3,4], [4,5], [5,7], [7,2], [5,6]}

Test Case 3 = { [], 1, 0} => edges visited = {[1,2], [2,9]}

مینیمم تعداد تست کیس برای هر دو قسمت پوشش یال و گره 3 تا میباشد که میتوان بیشتر هم تعریف کرد از جمله حالت x == v[mid]

2)

الف)



ب)

مسیر های اصلی:

1. 1. Start -> 2 -> 6 -> 7 -> 8
2. 1. Start -> 2 -> 3 -> 5 -> 2 -> 3 -> 4 -> 6 -> 7 -> 8 -> End

تست کیس های ما:

Test Case = { int n}

Test Case 1 = { 2 } => path 1

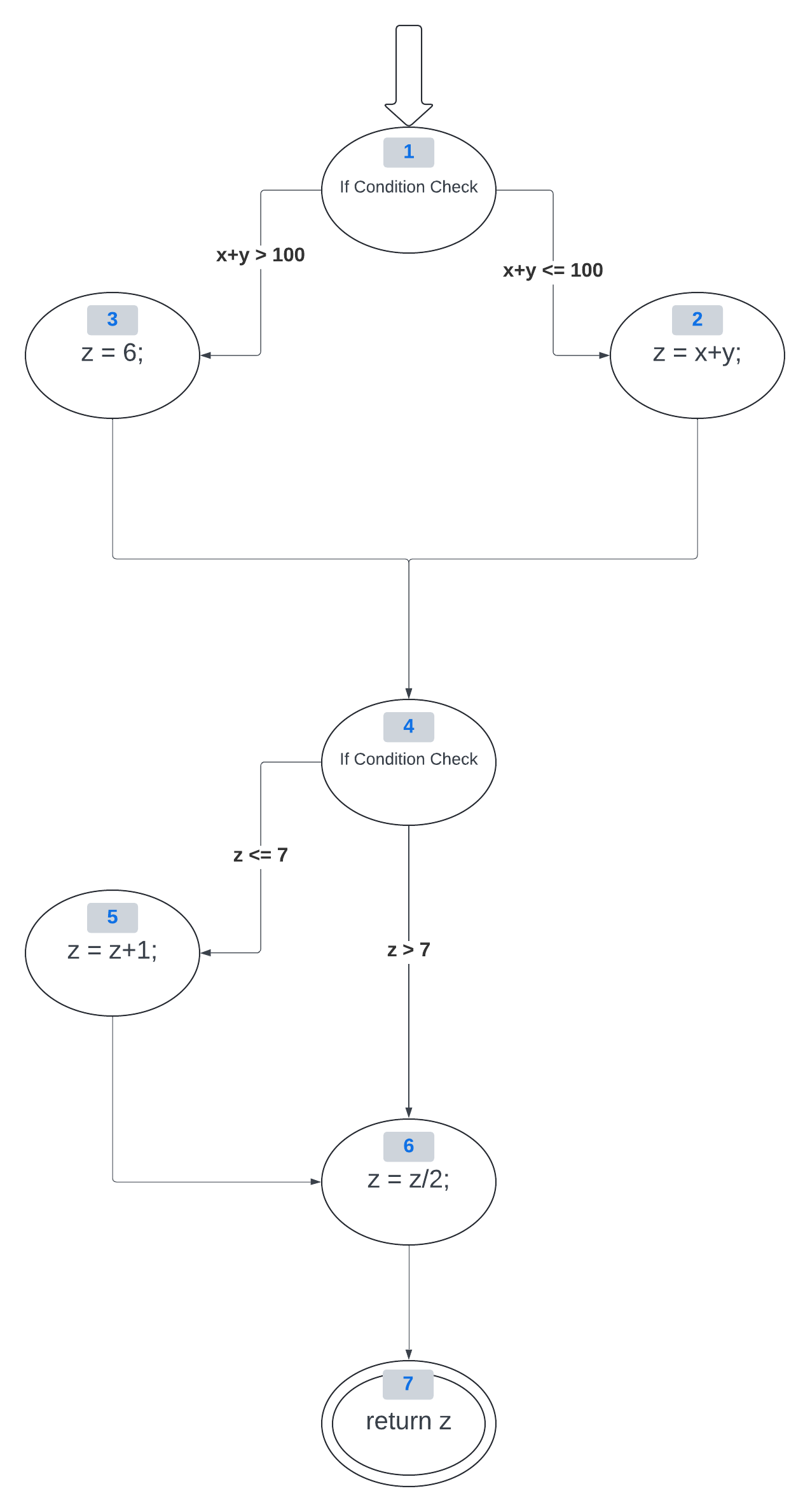
Test Case 1 = { 4 } => path 2

Test Case 1 = { 7 } => path 2

Test Case 1 = { 10 } => path 2

3)

الف)



ب)

Node Coverage:

TR = {1, 2, 3, 4, 5, 6, 7}

Edge Coverage:

TR = {[1,2], [1,3], [2,4], [3,4], [4,5], [4,6], [5,6], [6,7]}

پ)

2 مورد

Test Path = [1,3,4,5,6,7][1,2,4,5,6,7]

ت)

2 مورد

Test Path = [1,3,4,5,6,7][1,2,4,6,7]

این مورد پوشش گره را هم انجام میدهد.

ث)

اگر Statement Coverage را بر مبنای تعداد خط در نظر بگیریم طبق کد برنامه:

1 Public int computeIt (int x, int y) {

2 if (x+y > 100)

3 z = 6;

4 else

5 z = x+y;

6 if (z <= 7)

7 z = z+1;

8 z = z/2;

9 return z;

10 }

برای تست t1 شماره خط های {1,2,3,6,7,8,9,10} اجرا میشوند و برای تست t2 شماره خط های {1,2,4,5,8,9,10} اجرا میشوند که در مجموع برای S1 همه ی خط های برنامه اجرا شده است پس پوشش خط یا Statement Coverage برابر 100 درصد است.

ج)

طبق جواب قسمت ت با حداقل 2 مسیر میتوان به پوشش یال 100 درصد رسید. در این کد هر مجموعه تستی که پوشش یال 100 درصد داشته باشد قطعا پوشش گره 100 درصد هم دارد اما برعکس آن صادق نیست. طبق گراف دو مجموعه مسیر [1,3,4,5,6,7][1,2,4,5,6,7] پوشش گره 100 درصد دارند اما یال [4,6] را طی نکرده اند پس پوشش یال آن 100 درصد نیست.

مجموعه تست برای هم پوشش یال 100 درصدی هم پوشش گره 100 درصدی :

S = {t1:{x1,y1}, t2:{x2,y2}}

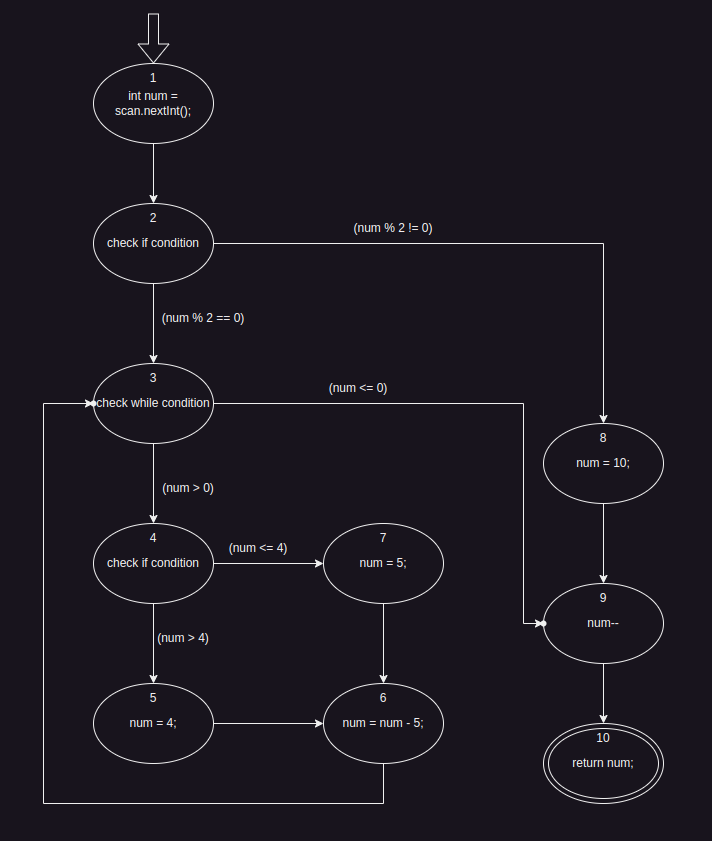
S1 = {t1:{100,1}, t2:{7,1}}

مجموعه تست فقط برای پوشش گره 100 درصدی:

S2 = {t1:{100,2}, t2:{6,1}}

4)

الف)

****

ب)

Node Coverage:

TR = { 1, 2, 3, 4, 5, 6, 3, 4, 7, 6, 3, 9, 10 }

TR = { 1, 2, 8, 9, 10}

Edge Coverage:

TR = {[1,2], [2,3], [2,8], [3,4], [3,9], [4,5], [4,7], [5,6], [6,3], [7,6], [8,9], [9,10]}

پ)

TR = { 1, 2, 8, 9, 10}

ت)

TR = {[1,2], [2,3], [2,8], [3,4], [3,9], [4,5], [4,7], [5,6], [6,3], [7,6], [8,9], [9,10]}

ث)

تست 1: عدد ورودی زوج و بزرگتر از 4 است.

ورودی: 6

خروجی مورد انتظار: 0

تست 2: عدد ورودی زوج و کوچکتر یا مساوی 4 است.

ورودی: 4

خروجی مورد انتظار: 1-

تست 3: عدد ورودی فرد است.

ورودی: 7

خروجی مورد انتظار: 9

تست 4: عدد ورودی 0 است.

ورودی: 0

خروجی مورد انتظار: 1-

تست 5: عدد ورودی منفی است.

ورودی: 3-

خروجی مورد انتظار: 1-