**חלק 2 (50 נק') – Control and dataflow testing, symbolic execution**

נתונות פונקציות הבאות:

A)

**public** **static** **double** calculatePrice(**double** distance, **int** age, **int** type) {

**double** cost = 1000;

**boolean** valid = **true;**

**if** (age < 1 || age >120 || distance < 0)

valid = **false;**

**if** (valid && age < 18 && distance > 500)

valid = **false;**

**if** (valid)

**switch** (type) {

**case** 2:

**double** temp = age \* distance / 100;

cost = Math.min(temp, cost);

**if** (age < 18)

cost = 100;

**break**;

**case** 1:

cost = distance \* 7 / age;

**break**;

**default**:

valid = **false;**

}

**if** (!valid)

**return** -1;

**return** cost;

}

B)

**public** **static** **int** findMostCommonPositiveNumber(**int**[] array){

**int** commNum = -1;

**int** commNumCount=0;

**for**(**int** i=0; i< array.length; i++){

**if** (array[i]>=0){

**int** count = 1;

**for**(**int** j=i+1; j< array.length; j++){

**if** (array[i] == array[j])

count++;

}

**if** (count>commNumCount){

commNumCount = count;

commNum = array[i];

}

}

**else**

System.***out***.println("Negative number!");

}

**if** (commNum!=-1)

System.***out***.println("The common number is: "+commNum);

**else**

System.***out***.println("No positive number in the array");

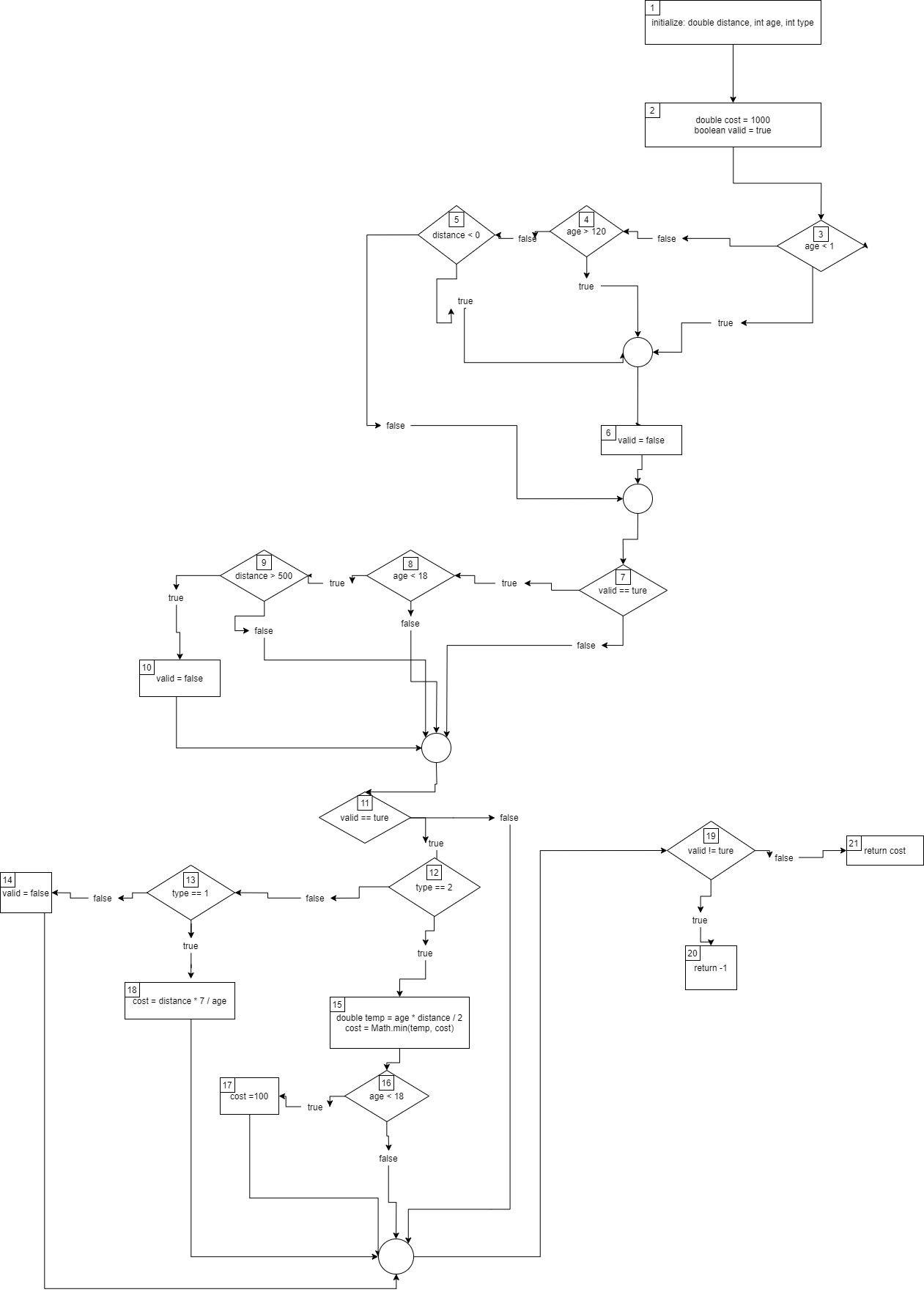
**return** commNum;

}

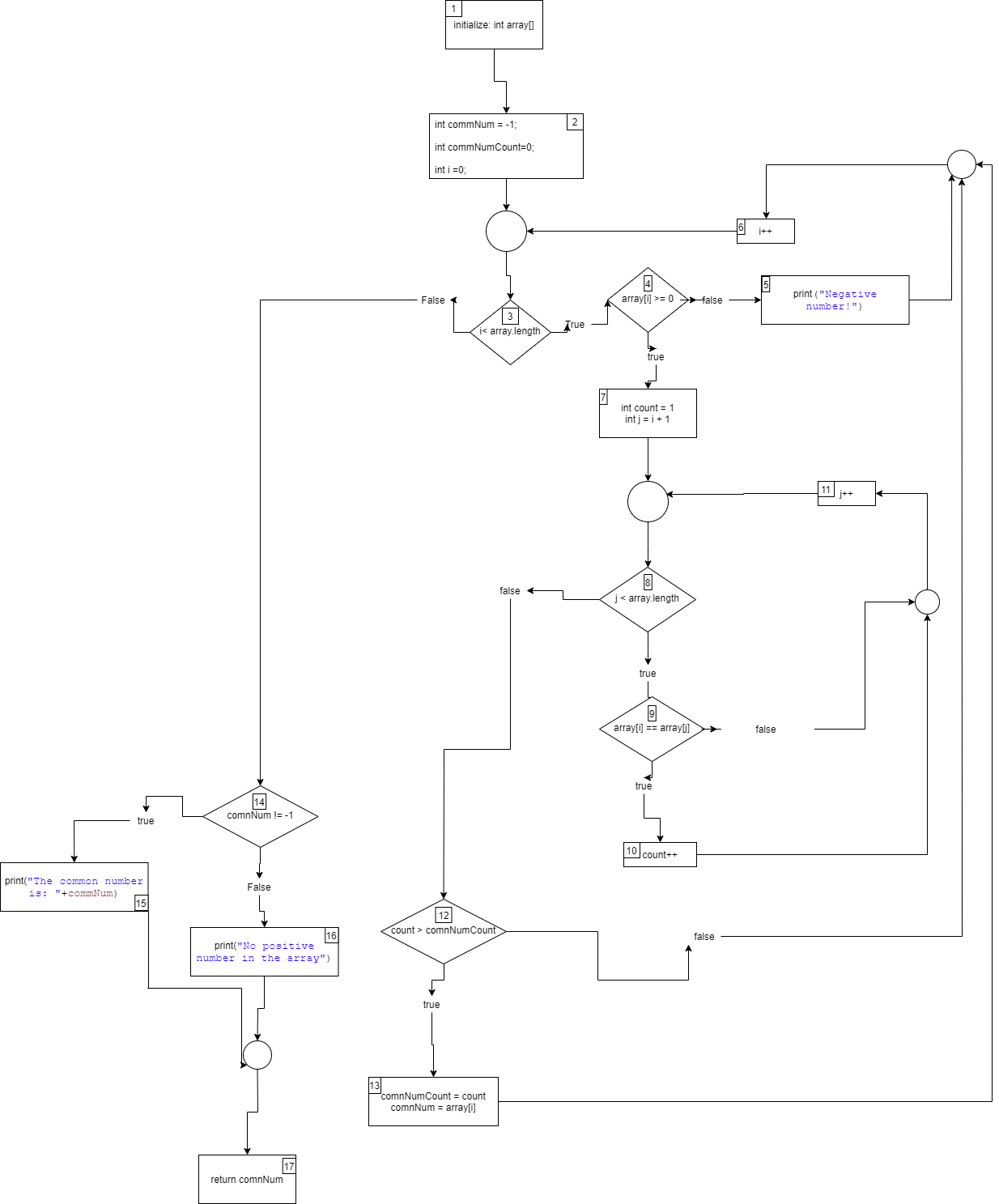
עבור הפונקציות A ו-B ענו על הסעיפים הבאים:

1. בנו את הגרף CFG עבור הפונקציות.

עבור A:



עבור B:



1. האם קיימים מסלולים שלמים (complete path) לא פיזיביליים בגרף? אם כן, רשמו לפחות מסלול אחד.

עבור A:

כאשר אנחנו עוברים בקודקוד 6 הפרמטר valid הופך ל-false ולא משתנה יותר לאורך המסלול ולכן, לאחר קודקוד תנאי 19 לעולם לא יגיע ל-21 כי valid != true תמיד יהיה true.

עבור B:

כאשר אנחנו עוברים בקודקוד 2 הפרמטר comnNum מקבל את הערך

ואם אנחנו עוברים מקודקוד 3 ישירות לקודקוד 14 הערך הזה לעולם לא ישתנה ולכן בקודקוד 14 התנאי comnNum =! -1 תמיד יהיה false ולא נצליח להגיע לקודקוד 15.

1. מצאו מסלולים שלמים העונים על הקריטריון של statement coverage מלא. רשמו את הקלט המתאים (test case) עבור כל אחד מהמסלולים. אם לא קיימים מסלולים פיזיביליים הנותנים כיסוי מלא, הסבירו מדוע.

עבור A:

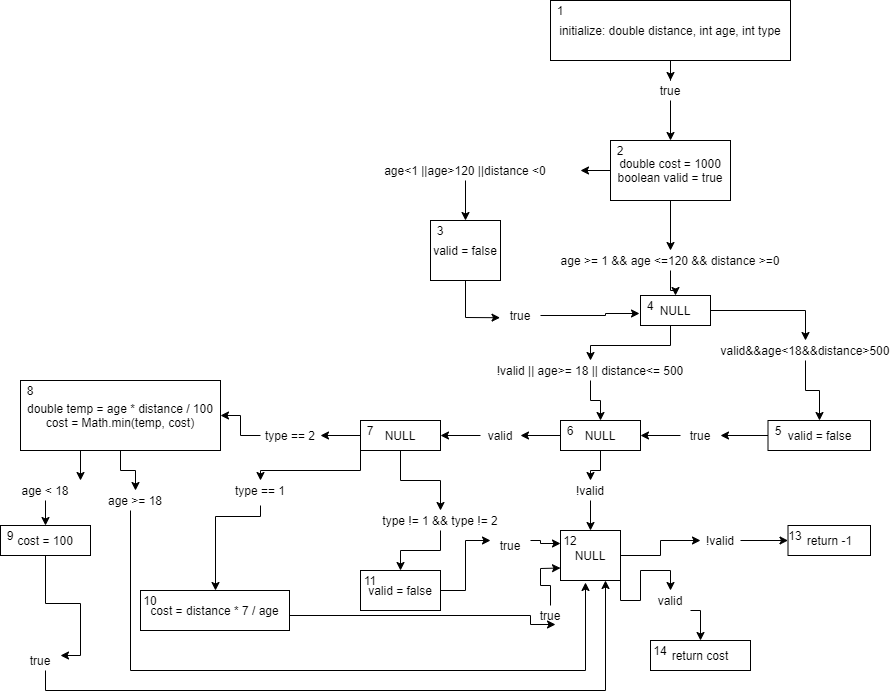
עבור B:

1. מצאו מסלולים שלמים העונים על הקריטריון של branch coverage מלא. רשמו את הקלט המתאים (test case) עבור כל אחד מהמסלולים. אם לא קיימים מסלולים פיזיביליים הנותנים כיסוי מלא, הסבירו מדוע.

עבור A:

עבור B:

1. בנו את הגרף DFG עבור הפונקציה.



1. עבור כל definition של כל אחד מהמשתנים ציינו האם הוא global definition. הסבירו.

|  |  |  |
| --- | --- | --- |
| Explain | Is global definition | Definition |
| יש clear path בין קודוקד זה לבין p-use של משתנה זה לאחר קודקוד 2 | כן | distance- node 1 |
| יש clear path בין קודוקד זה לבין p-use של משתנה זה לאחר קודקוד 2 | כן | age-node 1 |
| יש clear path בין קודוקד זה לבין p-use של משתנה זה לאחר קודקוד 7 | כן | type-node 1 |
| יש clear path בין קודקוד זה לבין c-use בקודוק 8 כאשר משתמשים בcost בתוך פונקציית הmath- מסלול: | כן | cost-node 2 |
| יש clear path מהגדרה זו ל-p-use של משתנה זה לאחר קודקוד 4 כאשר המסלול הוא: | כן | valid-node 2 |
| יש clear path מהגדרה זו ל-p-use של משתנה זה לאחר קודקוד 4 כאשר המסלול הוא: | כן | valid – node 3 |
| יש clear path מהגדרה זו ל-p-use של משתנה זה לאחר קודקוד 6 כאשר המסלול הוא: | כן | valid – node 5 |
| השימוש במשתנה זה נעשה רק בתוך אותו קודקוד | לא | temp- node 8 |
| יש clear path מהגדרה זו ל- c-use של משתנה זה לאחר המסלול הוא: | כן | cost – node 8 |
| יש clear path מהגדרה זו ל- c-use של משתנה זה לאחר המסלול הוא: | כן | cost – node 9 |
| יש clear path מהגדרה זו ל- c-use של משתנה זה לאחר המסלול הוא: | כן | cost – node 10 |
| יש clear path מהגדרה זו ל-p-use של משתנה זה לאחר קודקוד 12 כאשר המסלול הוא: | כן | valid – node 11 |

1. רשמו את כל ה-du pairs של כל אחד מהמשתנים. כלומר, עבור כל משתנה, יש לרשום את כל הזוגות של definition ו- use (p-use או c-use) כך שיש def clear path בין definition ו-use.

|  |  |
| --- | --- |
| Use | Definition |
| Edge(2,3) | distance node 1 |
| Edge(2,4) |
| Edge(4,5) |
| Edge(4,6) |
| Node 8 |
| Node 10 |
| Edge(2,3) | age node 1 |
| Edge(2,4) |
| Edge(4,5) |
| Edge(4,6) |
| Node 8 |
| Node 10 |
| Edge(8,9) |
| Edge(8,12) |
| Edge(7,8) | type node 1 |
| Edge(7,10) |
| Edge(7,11) |
| Node 8 | cost node 2 |
| Node 14 | cost node 8 |
| Node 14 | cost node 9 |
| Node 14 | cost node 10 |
| Edge(4,6) | valid node 2 |
| Edge(4,5) |
| Edge(6,7) |
| Edge(6,12) |
| Edge(12,14) |
| Edge(12,13) |
| Edge(4,6) | valid node 3 |
| Edge(4,5) |
| Edge(6,7) |
| Edge(6,12) |
| Edge(12,14) |
| Edge(12,13) |
| Edge(6,7) | valid node 5 |
| Edge(6,12) |
| Edge(12,14) |
| Edge(12,13) |
| Edge(12,14) | valid node 11 |
| Edge(12,13) |

1. מצאו קבוצת מסלולים וקלט מתאים לכל מסלול, אשר תהוונה כיסוי מלא לפי קריטריונים הבאים:

אם לא ניתן לעמוד באחד או יותר מהכיסויים הסבירו למה.

All-defs

* 1. All-C-uses
  2. All-P-uses

לא ניתן למצוא כי לא ניתן להגיע מnode ,5,3,11 ל-קשת (12,14) כי קודם נבדק התנאי

!valid ובגלל שבשלושת הקודקודים שציינתי הופכים את valid לfalse לא מגיעים לקשת עם ה- p-use valid.

* 1. All-uses

לא ניתן למצוא כי לא ניתן להגיע מnode ,5,3,11 ל-קשת (12,14) כי קודם נבדק התנאי

!valid ובגלל שבשלושת הקודקודים שציינתי הופכים את valid לfalse לא מגיעים לקשת עם ה- p-use valid.

* 1. All-P-uses/Some-C-uses

לא ניתן למצוא כי לא ניתן להגיע מnode ,5,3,11 ל-קשת (12,14) כי קודם נבדק התנאי

!valid ובגלל שבשלושת הקודקודים שציינתי הופכים את valid לfalse לא מגיעים לקשת עם ה- p-use valid.

* 1. All-C-uses/Some-P-uses

1. נתון הקוד הבא המתבסס על קוד A עם שינויים:

**public** **static** **double** calculatePriceAdjusted(**double** distance, **int** age) {

**2. double** cost = 1000;

**3. if** (age < 1 || age >120 || distance < 0)

4. cost = age\*(Math.E^)(-1)\*distance();

**5. else if** ( distance > 500)

6. cost = Math.min(age \* distance / 100, cost);

**7.** assert (cost<=1)

}

ציירו תרשים עץ symbolic execution סטאטי המפרט את ה- σו- π עבור כל stmt.

