

Quality Assurance 2.0 -> Lessons 12.White-box Testing

White-box – це метод тестування, який перевіряє внутрішнє функціонування системи. У цьому методі тестування перевіряються внутрішня структура, будова і код програмного забезпечення. У тестуванні білого ящика код є доступний тестувальникам, тому його також називають тестуванням чистого ящика, тестуванням відкритого ящика, тестуванням прозорого ящика, тестуванням на основі коду та тестуванням скляного ящика.

Типи покриття (types of coverage):

1. **Statement coverage**, або **тестування операторів**, використовує потенційні виконувані оператори в коді. Покриття вимірюється як кількість операторів, виконаних за допомогою тестів, поділена на загальну кількість виконуваних операторів у об'єкті тестування, зазвичай виражається у відсотках.
2. **Decision/Branch coverage**, або **тестування рішень**, реалізує рішення в коді та перевіряє код, який виконується на основі результатів рішення. Для цього тестові випадки слідують за потоками керування, які відбуваються з точки прийняття рішення. Покриття рішень вимірюється як кількість результатів рішення, виконаних за допомогою тестів, поділена на загальну кількість результатів рішення в об'єкті тестування, зазвичай виражається у відсотках.
3. **Path coverage**, або **тестування шляху**, полягає у визначенні шляхів у коді та створенні тестів для їх покриття. Концептуально було б корисно перевірити кожен унікальний шлях через систему. Однак у будь-якій не тривіальній системі кількість тестових випадків може стати надмірно великою через природу циклічних структур. Для застосування цієї техніки *Борис Бейзер* (американський програміст, автор книг з тестування та програмування) рекомендує створювати тести, які *проходять багато шляхів через модуль, від входу до виходу*. Покриття вимірюється як кількість протестованих шляхів в програмному коді поділена на загальну кількість шляхів і помножена на 100%.

Завдання

Для кожного завдання потрібно надати рішення у вигляді блок-схеми, яка буде відображати псевдокод кожного завдання.

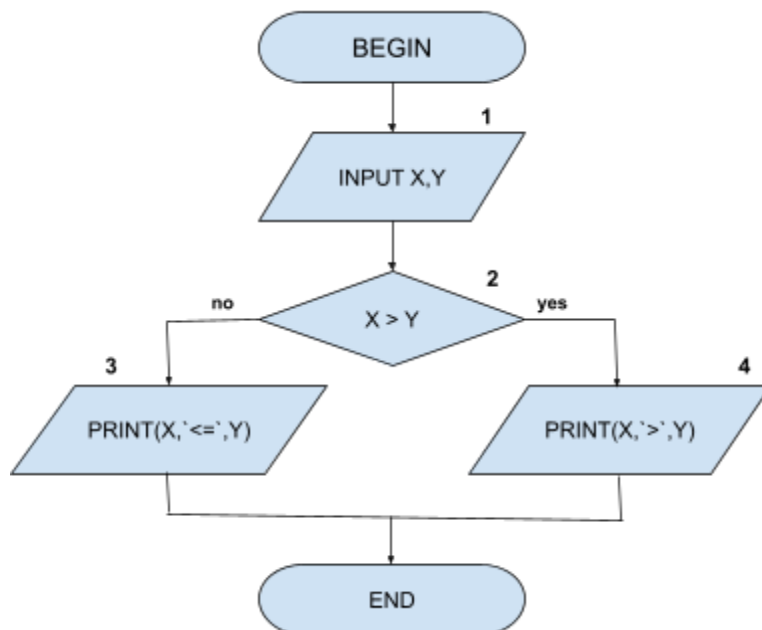
1. Зобразити псевдокод у вигляді блок-схеми

```
Begin
Input X, Y
If X > Y
  __Print (X, 'is greater than', Y)
Else
  __Print (Y, 'is greater than or equal to', X)
End If
End
```

Яка мінімальна кількість тест-кейсів потрібна, щоб гарантувати **100% statement** і **100% decision** покриття?

- A. Statement coverage = 3, Decision coverage = 3
- B. Statement coverage = 2, Decision coverage = 2
- C. Statement coverage = 1, Decision coverage = 2
- D. Statement coverage = 2, Decision coverage = 1

Блок-Схема:



Рішення:

Тест 1: *Data: $X \leq Y$, Statements: 1-2-3, Decisions: 2-no*

Тест 2: *Data: $X > Y$, Statements: 1-2-4, Decisions: 2-yes*

Відповідь: B. Statement coverage = 2, Decision coverage = 2

2. Зобразити псевдокод у вигляді блок-схеми.

```
if (Condition 1)
then statement 1
else statement 2
fi

if (Condition 2)
then statement 3
fi
```

*Яка мінімальна кількість тест-кейсів потрібна, щоб гарантувати **100% path coverage** (покриття шляху)?*

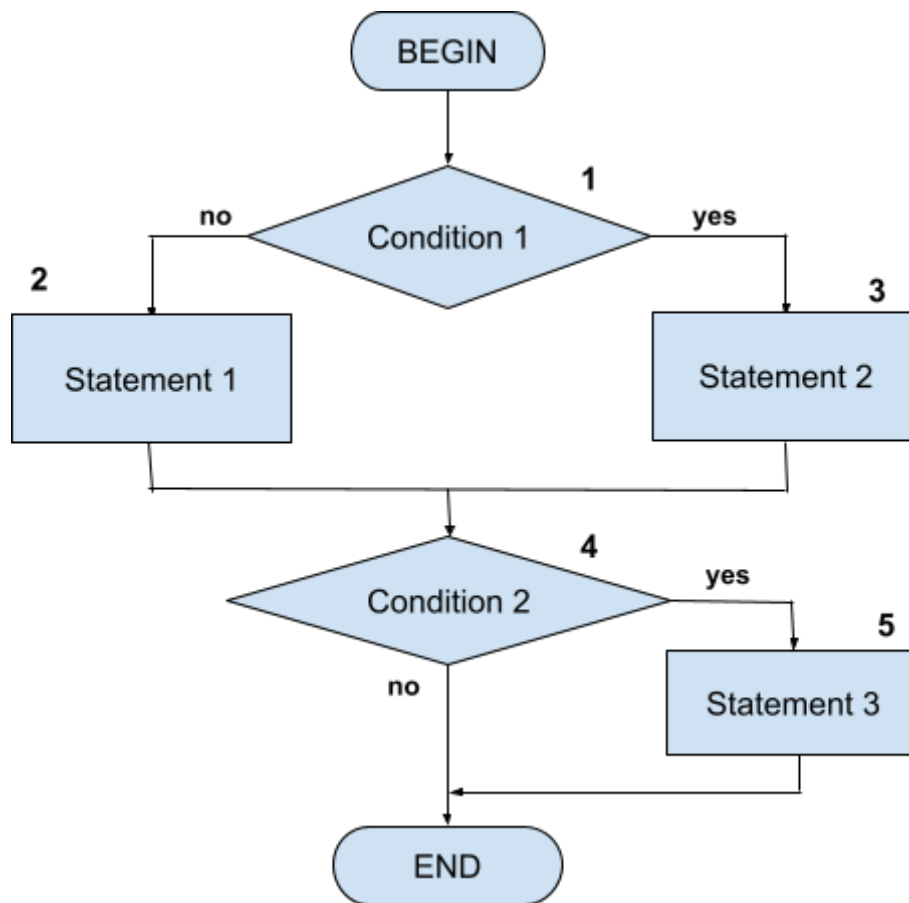
A. 1

B. 2

C. 3

D. Жодна відповідь невірна

Блок-Схема:



Рішення:

Тест 1: *Data: Condition 1 = yes, Condition 2 = yes, Path: 1-3-4-5*

Тест 2: *Data: Condition 1 = no, Condition 2 = no, Path: 1-2-4*

Тест 3: *Data: Condition 1 = yes, Condition 2 = no, Path: 1-3-4*

Тест 4: *Data: Condition 1 = no, Condition 2 = yes, Path: 1-2-4-5*

Відповідь: D. Жодна відповідь невірна, тому що щоб гарантувати 100% path coverage, треба мінімум 4 тест-кейса.

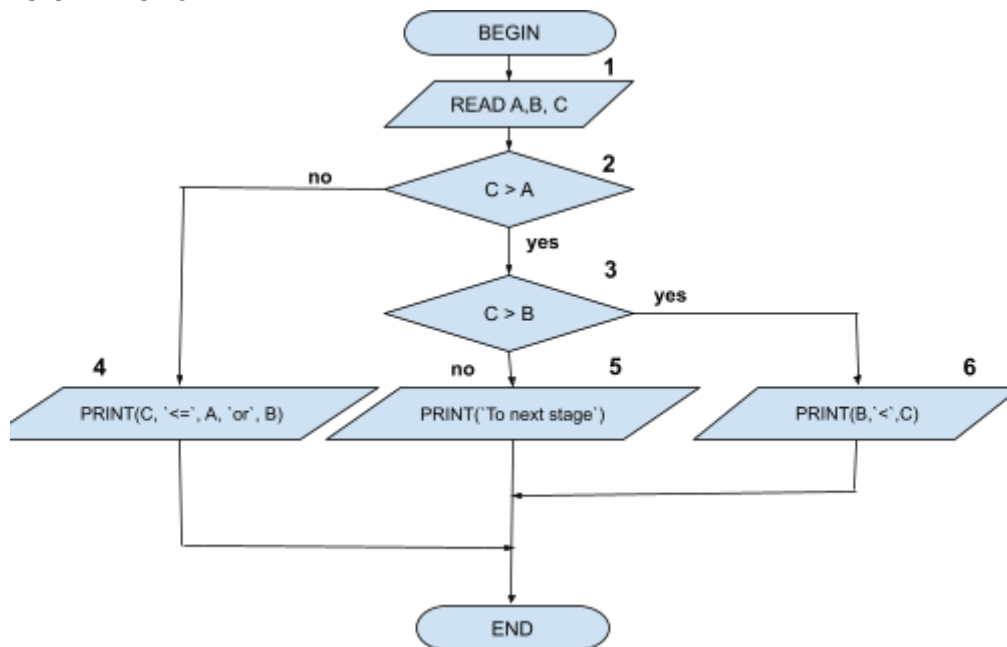
3. Зобразити псевдокод у вигляді блок-схеми.

```
READ A READ B READ C
IF C>A THEN
IF C>B THEN
PRINT 'B can be smaller than C'
ELSE PRINT 'Proceed to next stage'
END IF
ELSE PRINT 'C must be smaller than at least one number'
END IF
```

Яка мінімальна кількість тест-кейсів потрібна, щоб гарантувати **100% statement** та **100% decision** покриття?

- A. 2, 4
- B. 3, 2
- C. 3, 3
- D. 2, 3

Блок-Схема:



Рішення:

Тест 1: Data: **C > A > B**, Statements: **1-2-3-6**, Decisions: **2-yes, 3-yes**

Тест 2: Data: **B > C > A**, Statements: **1-2-3-5**, Decisions: **2-yes, 3-no**

Тест 3: Data: **A > C, B=будь яке**, Statements: **1-2-4**, Decisions: **2-no**

Відповідь: C. 3, 3

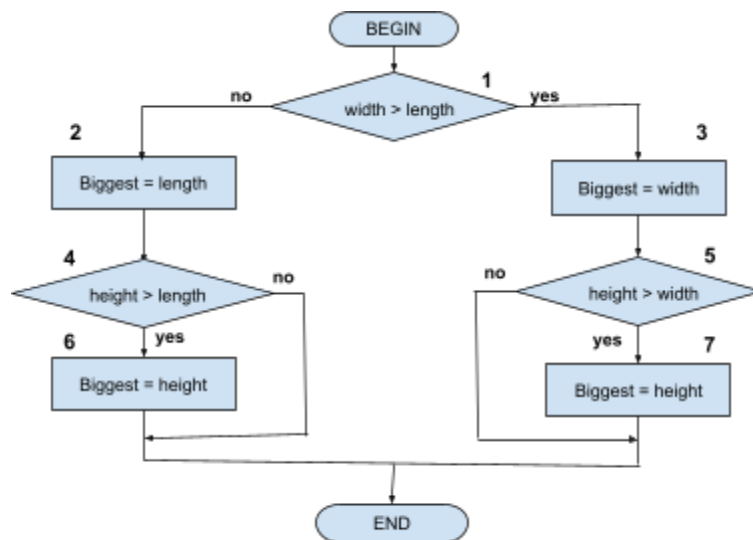
4. Зобразити псевдокод у вигляді блок-схеми.

```
if width > length
then biggest_dimension = width
if height > width then biggest_dimension = height
end_if
else biggest_dimension = length
if height > length then biggest_dimension = height
end_if
end_if
```

Яка мінімальна кількість тест-кейсів потрібна, щоб **гарантувати 100% decision coverage** (покриття рішень)?

- A. 3
- B. 4
- C. 1
- D. 2

Блок-Схема:



Рішення:

Тест 1: Data: **height > width > length**, Decisions: **1-yes, 5-yes**

Тест 2: Data: **height > length > width**, Decisions: **1-no, 4-yes**

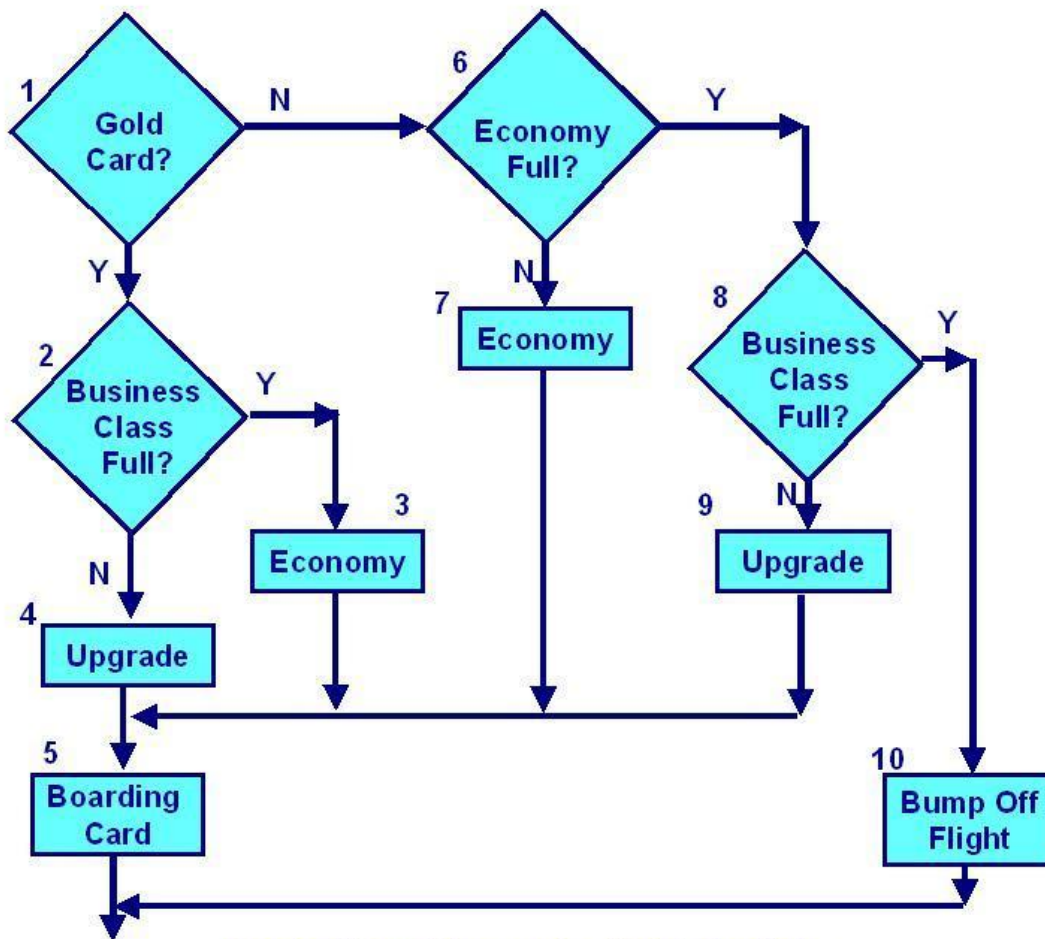
Тест 3: Data: **length > height > width**, Decisions: **1-no, 4-no**

Тест 4: Data: **width > length > height**, Decisions: **1-yes, 5-no**

Відповідь: B. 4

5. Обслуговування пасажирів літака

Якщо ви летите в економ класі, то вас можуть підвищити до бізнес класу, особливо, якщо у Вас є золота картка авіалінії для приватних перельотів. Якщо у вас немає золотої картки, вас можуть “скинути” з рейсу, якщо літак переповнений, або ви спізнились на реєстрацію. Всі ці умови показані на схемі нижче. Зверніть увагу, що всі statements (оператори) пронумеровані.



Control Flow Diagram for Flight Check-in

Ви запускаєте 3 тести:

Тест 1 - Власник золотої картки підвищений до бізнес класу.

Тест 2 - Пасажир без золотої картки залишається в економ класі.

Тест 3 - Пасажир, якого “скинули” з рейсу.

Яке **statement coverage** (покриття операторів) даних трьох тестів?

- A. 60%
- B. 70%
- C. 80%
- D. 90%

Новели:

Тест 1: Пасажир Economy Class є Володарем Gold Card (**1Y**), а в Business Class є вільні місця (**2N**), тож його направляють туди (**4**), й заносять цей факт у Boarding Card (**5**).

Тест 2: Пасажир Economy Class не є Володарем Gold Card (**1N**), а в Economy Class є вільні місця (**6N**), тож його направляють туди (**7**), й заносять цей факт у Boarding Card (**5**).

Тест 3: Пасажир Economy Class не є Володарем Gold Card (**1N**), причому в Economy Class немає вільних місць (**6Y**), і в Business Class немає вільних місць (**8Y**), тож його залишають в аеропорту (**10**).

Рішення:

Всього: Statements $s = 10$, Decisions $d = 8$

Тест 1: Statements: **1-2-4-5** $\Rightarrow x = 4$, Decisions: **1Y, 2N** $\Rightarrow a = 2$

Тест 2: Statements: **1-6-7-5** $\Rightarrow y = 2$, Decisions: **1N, 6N** $\Rightarrow b = 2$

Тест 3: Statements: **1-6-8-10** $\Rightarrow z = 2$, Decisions: **1N, 6Y, 8Y** $\Rightarrow c = 2$

Statements, не перевірені у тестах: **3 та 9**.

Decisions, не перевірені у тестах: **2Y, 8N**

Statement Coverage = $100\% * (x+y+z)/s = 100\% * (4+2+2)/10 = 80\%$

Decision Coverage = $100\% * (a+b+c)/d = 100\% * (2+2+2)/8 = 75\%$

Відповідь: C. 80%