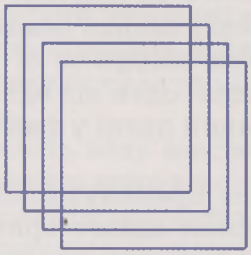
3. Збережіть проект у вашій папці у файлі з іменем **практична 5_1**.
4. Складіть проект для малювання кіл різного кольору (мал. 3.23), поки виконавець не доторкнеться до межі **Сцени**.
5. Збережіть проект у вашій папці у файлі з іменем **практична 5_2**.
6. Закрийте середовище **Scratch**.



Мал. 3.22. Чотири квадрати



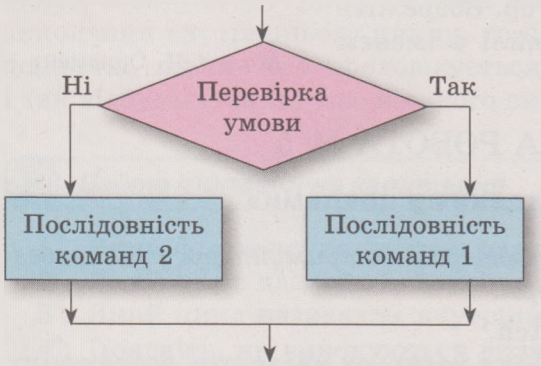
Мал. 3.23. Кола різних кольорів

3.3. ВКЛАДЕНІ РОЗГАЛУЖЕННЯ

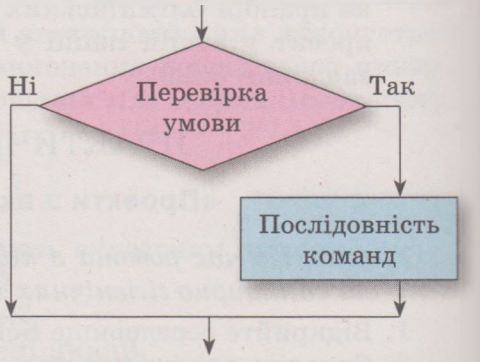
1. Який фрагмент алгоритму називають повним розгалуженням?
2. Який фрагмент алгоритму називають неповним розгалуженням?
3. Під час розв'язування яких завдань ви використовували розгалуження?

ВКЛАДЕНІ РОЗГАЛУЖЕННЯ

У попередніх класах ви вже ознайомилися з розгалуженнями та використовували їх. Нагадаємо, що розгалуження може бути повним (мал. 3.24) і неповним (мал. 3.25).



Мал. 3.24. Повне розгалуження



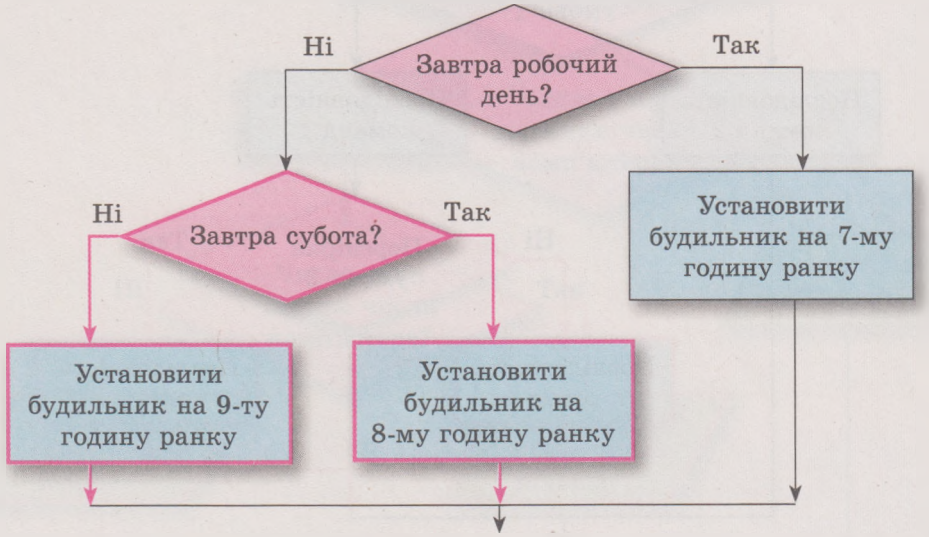
Мал. 3.25. Неповне розгалуження

Ви розглядали алгоритми, у яких було кілька розгалужень, що виконувалися по черзі, одне за одним. У них кожне наступне розгалуження виконувалося тоді, коли виконання попереднього розгалуження вже закінчилося.

Але бувають й інші випадки, інші життєві ситуації. У них наступне розгалуження виконується, коли виконання попереднього розгалуження ще не закінчено.

Наприклад, вам потрібно встановити будильник на завтра. Якщо завтра робочий день, то ви повинні встати о 7-й годині ранку, щоб іти до школи. Якщо завтра субота, то ви повинні встати о 8-й годині ранку, щоб їхати на заняття гуртка. Якщо завтра неділя, то ви встаєте о 9-й годині ранку.

Блок-схема алгоритму встановлення будильника має такий вигляд (мал. 3.26):



Мал. 3.26. Алгоритм установлення будильника

У наведеному на малюнку 3.26 алгоритмі друге розгалуження з умовою *Завтра субота?* міститься всередині першого розгалуження з умовою *Завтра робочий день?*.

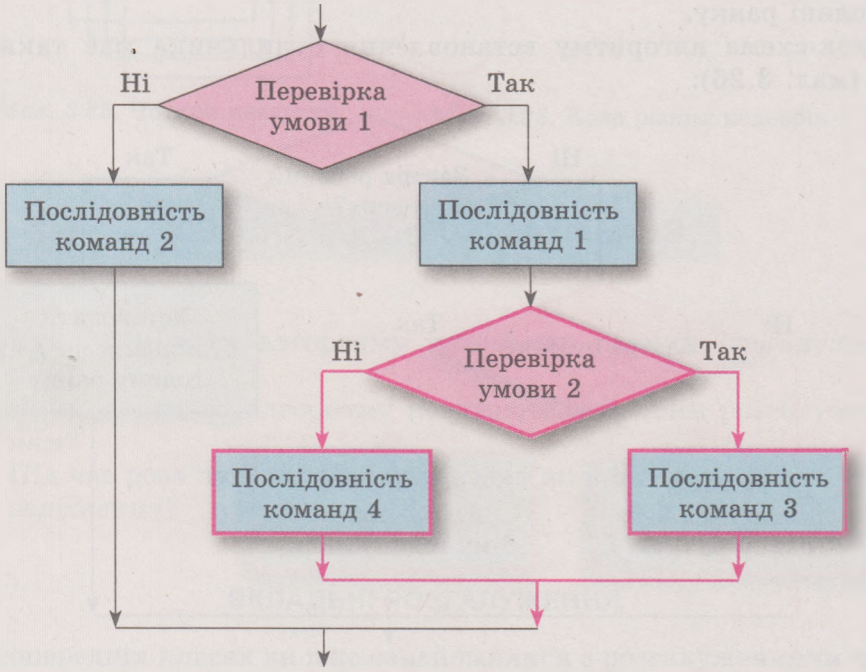
Такий фрагмент алгоритму називають **вкладеним розгалуженням**.

Вкладені розгалуження – це фрагмент алгоритму, у якому одне розгалуження міститься всередині іншого розгалуження.

Розглянемо виконання наведеного на малюнку 3.26 фрагмента алгоритму. Спочатку перевіряється умова *Завтра робочий день?*. Якщо результат перевірки цієї умови **Так**, то виконується команда *Установити будильник на 7-му годину ранку* і на цьому виконання всього цього фрагмента алгоритму закінчується. Якщо результат перевірки умови *Завтра робочий день?* – **Ні**, то перевіряється умова *Завтра субота?*. Якщо результат перевірки цієї умови **Так**, то виконується команда

Установити будильник на 8-му годину ранку і на цьому виконання всього цього фрагмента алгоритму закінчується, а якщо результат перевірки цієї умови **Ні**, то виконується команда *Установити будильник на 9-ту годину ранку* і виконання всього цього фрагмента алгоритму закінчується.

У наведеному на малюнку 3.26 фрагменті алгоритму внутрішнє розгалуження виконується, якщо результат перевірки умови зовнішнього розгалуження **Ні**. Аналогічно можливі такі вкладені розгалуження, у яких внутрішнє розгалуження виконується, якщо результат перевірки умови зовнішнього розгалуження **Так**. Приклад такого фрагмента алгоритму наведено на малюнку 3.27.

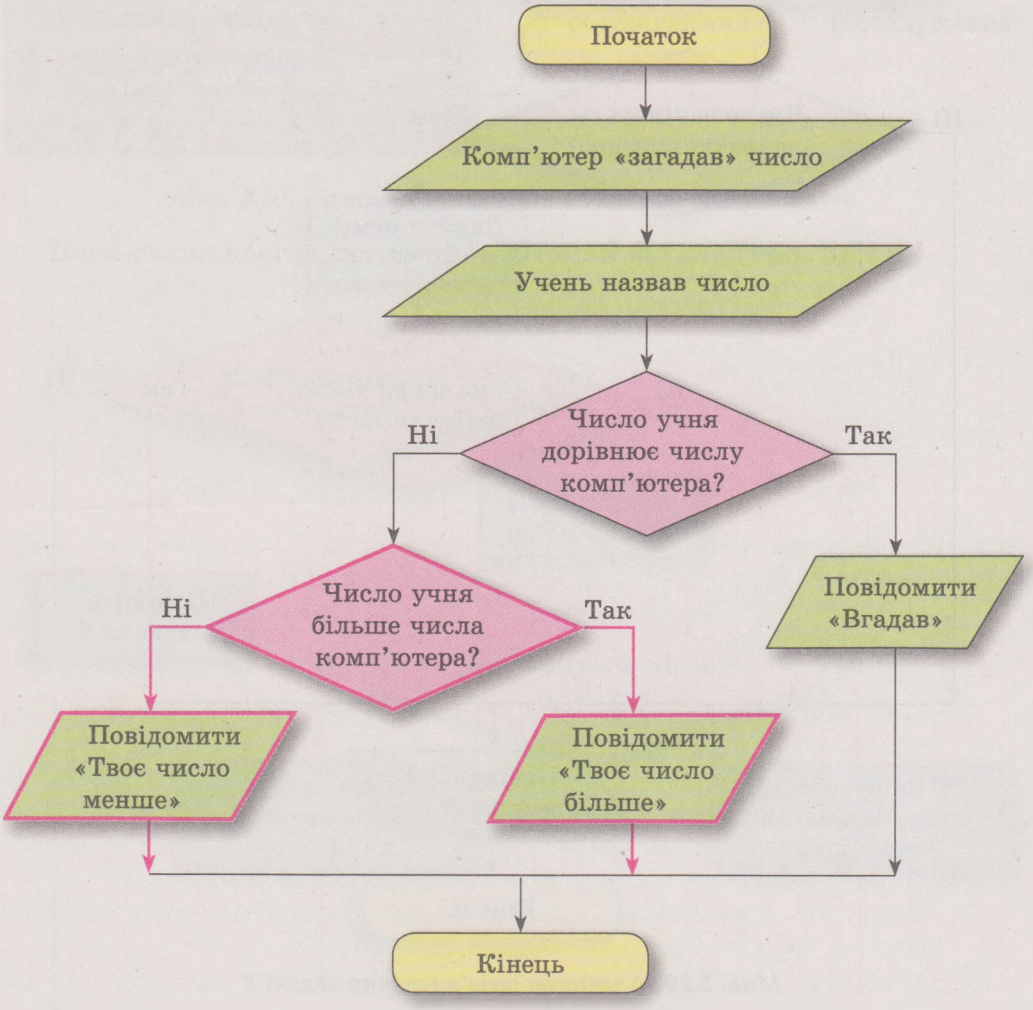


Мал. 3.27. Вкладені розгалуження

Розглянемо виконання наведеного на малюнку 3.27 фрагмента алгоритму. Спочатку перевіряється *умова 1*. Якщо результат перевірки цієї умови **Ні**, то виконується *Послідовність команд 2* і виконання всього цього фрагмента алгоритму закінчується. Якщо результат перевірки *умови 1* – **Так**, то виконується *Послідовність команд 1* і потім перевіряється *умова 2*. Якщо результат перевірки *умови 2* – **Так**, то виконується *Послідовність команд 3* і виконання всього цього фрагмента алгоритму закінчується. А якщо результат перевірки *умови 2* – **Ні**, то виконується *Послідовність команд 4* і виконання всього цього фрагмента алгоритму закінчується.

Наведемо приклад алгоритму з вкладеними розгалуженнями для розв'язування нижченаведеної задачі-гри.

Задача-гра 1. Комп'ютер «загадав» натуральне число від 1 до 20. Учень намагається його вгадати. А комп'ютер відповідає, чи вгадав учень загадане число, чи учень назвав число, більше від загаданого, чи учень назвав число, менше від загаданого (мал. 3.28).

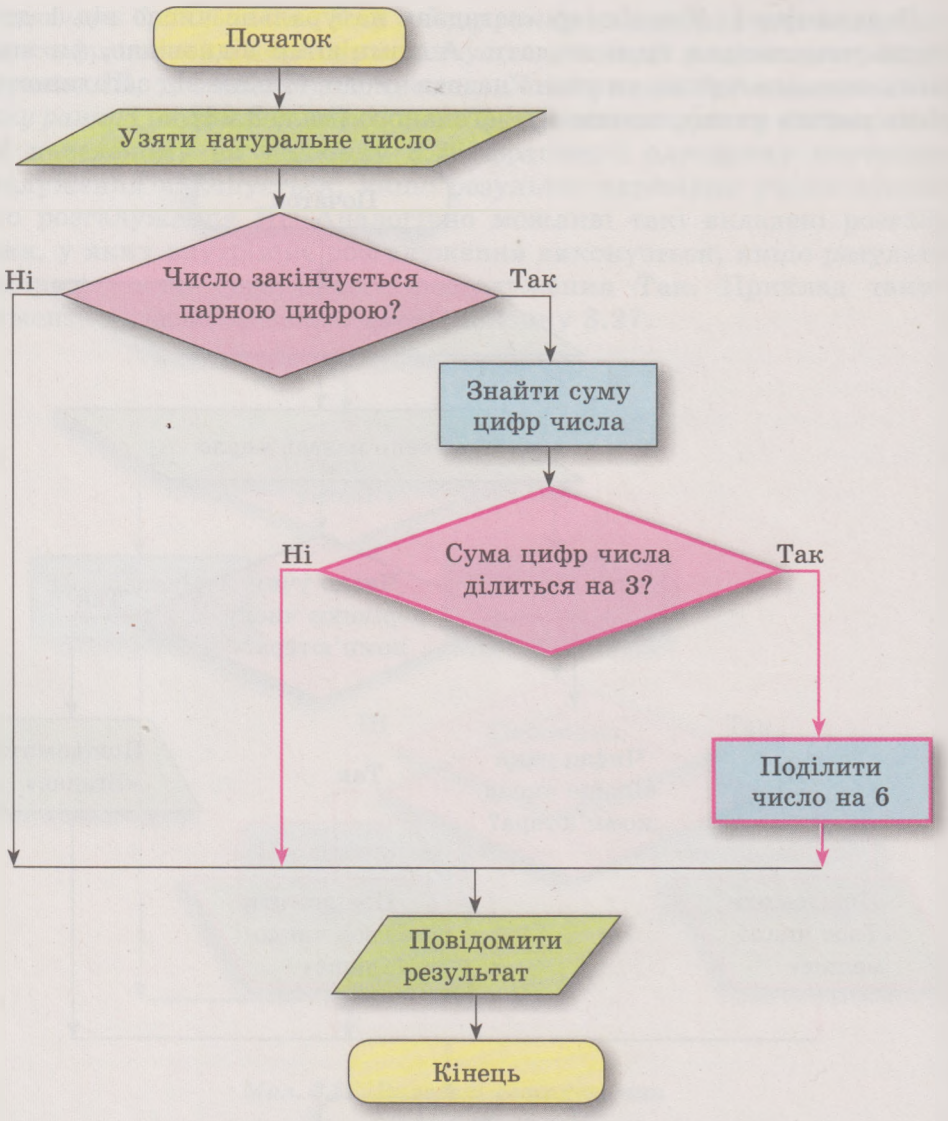


Мал. 3.28. Блок-схема алгоритму задачі-гри 1

У наведених вище прикладах і зовнішнє, і внутрішнє розгалуження є повним. Але серед вкладених розгалужень можуть бути і неповні розгалуження. Наведемо приклад алгоритму з такими вкладеними розгалуженнями.

Задача 2. Нехай задано натуральне число. Якщо воно ділиться на 6, то потрібно його розділити на 6.

Щоб число ділилося на 6, воно повинно ділитися на 2 і на 3. А ознаки подільності на 2 і на 3 ви знаєте. Тоді маємо такий алгоритм (мал. 3.29).



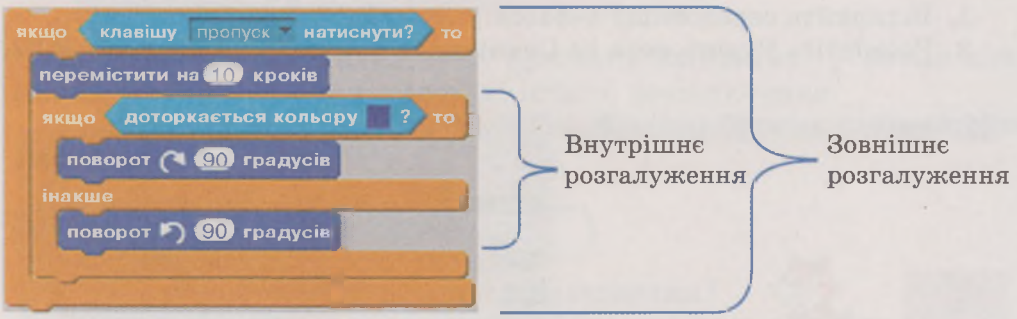
Мал. 3.29. Алгоритм розв'язування задачі 2

ВКЛАДЕНІ РОЗГАЛУЖЕННЯ В SCRATCH

Вкладені розгалуження можна використовувати і в **Scratch.**

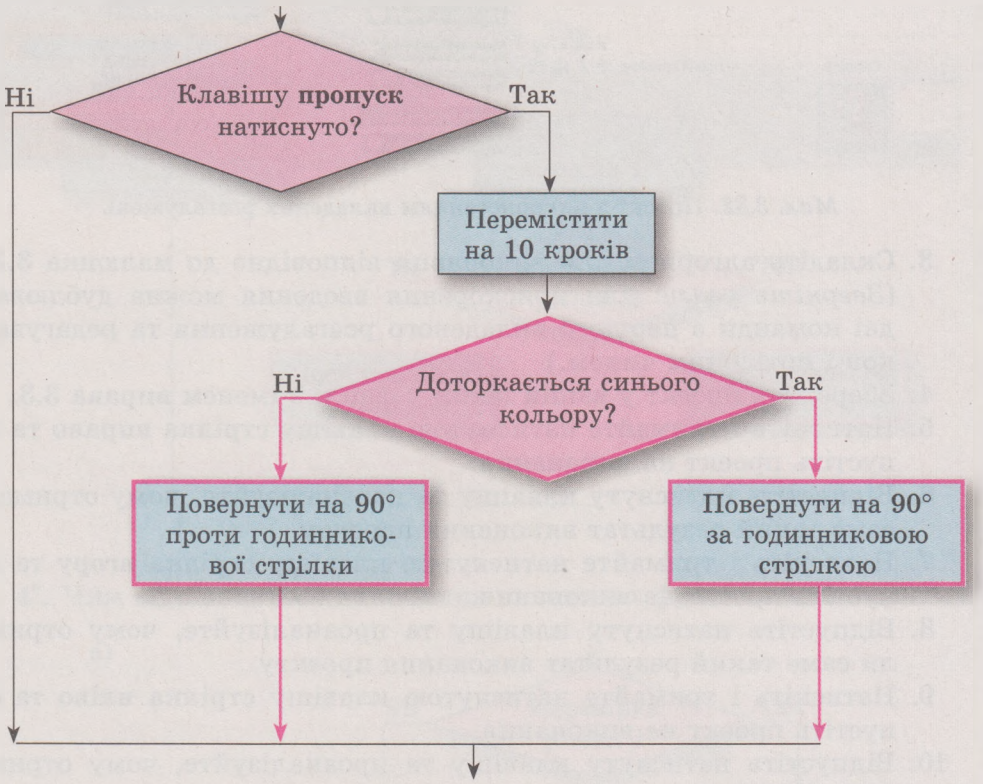
Наведемо приклад фрагмента проекта в **Scratch** з використанням вкладених розгалужень (мал. 3.30).

У наведеному фрагменті **якщо клавішу пропуск натиснуто,** виконавець переміщується на 10 кроків уперед. Якщо після цього переміщення він доторкається синього кольору, то він повертається на 90° за годинниковою стрілкою, а інакше (не доторкається синього кольору) він повертається на 90° проти годинникової стрілки.



Мал. 3.30. Приклад вкладеного розгалуження в **Scratch**

Блок-схема цього фрагмента має такий вигляд (мал. 3.31):



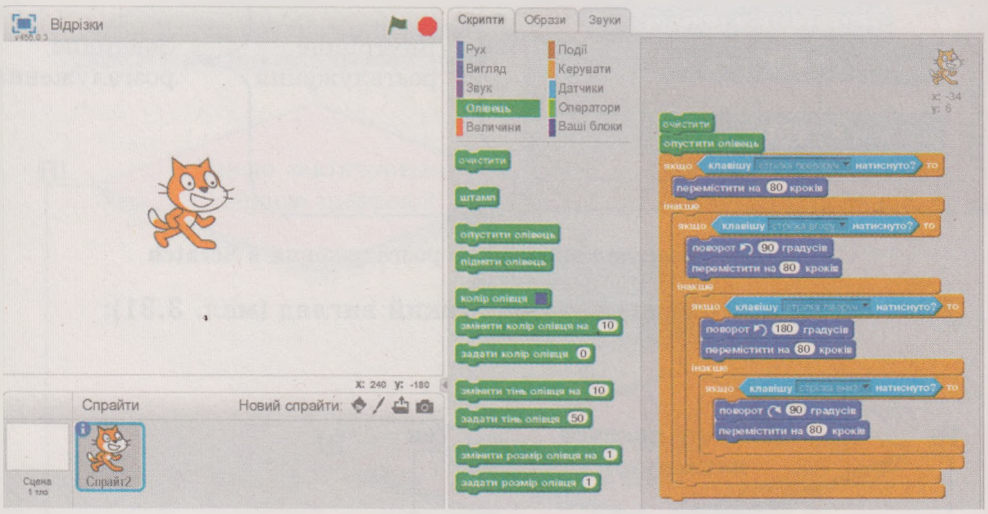
Мал. 3.31. Блок-схема фрагмента з вкладеними розгалуженнями в **Scratch**

Працюємо з комп'ютером

Увага! *Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.*

Задача. Скласти проект для малювання відрізка завдовжки 80 кроків у відповідному напрямку від положення виконавця залежно від того, яку клавішу-стрілку натиснуто.

1. Відкрийте середовище **Scratch**.
2. Розмістіть **Рудого кота** на **Сцені** як це показано на малюнку 3.32.



Мал. 3.32. Проект з використанням вкладених розгалужень

3. Складіть алгоритм для виконавця відповідно до малюнка 3.32. (*Зверніть увагу*: для прискорення введення можна дублювати дві команди з першого вкладеного розгалуження та редагувати копії потрібним чином.)
4. Збережіть проект у вашій папці у файлі з іменем **вправа 3.3**.
5. Натисніть і тримайте натиснутою клавішу **стрілка вправо** та запустіть проект на виконання.
6. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
7. Натисніть і тримайте натиснутою клавішу **стрілка вгору** та запустіть проект на виконання.
8. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
9. Натисніть і тримайте натиснутою клавішу **стрілка вліво** та запустіть проект на виконання.
10. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
11. Натисніть і тримайте натиснутою клавішу **стрілка вниз** та запустіть проект на виконання.
12. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
13. Натисніть і тримайте натиснутою будь-яку іншу клавішу та запустіть проект на виконання.
14. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
15. Закрийте середовище **Scratch**.

Найважливіше в цьому пункті

Вкладені розгалуження – це фрагмент алгоритму, у якому одне розгалуження міститься всередині іншого розгалуження.

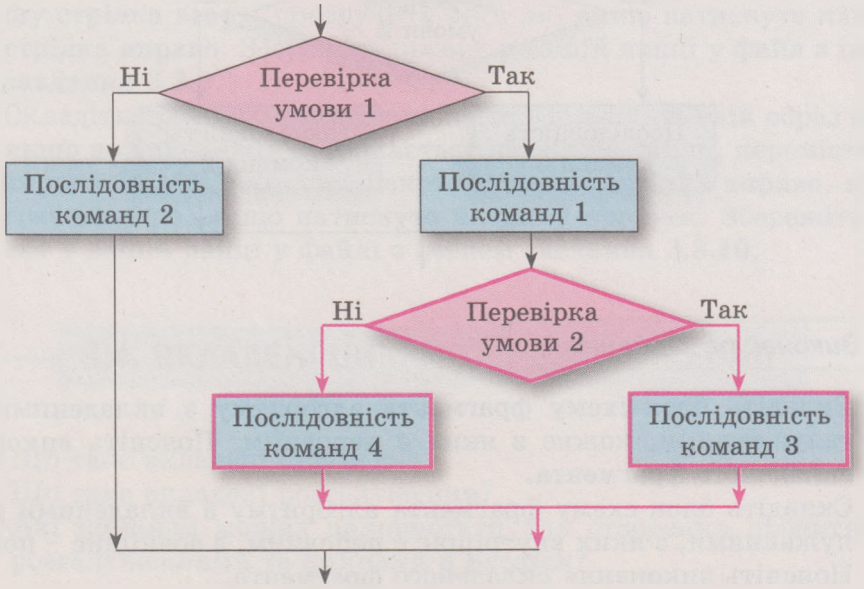
І зовнішні, і внутрішні розгалуження можуть бути як повними, так і неповними.

Дайте відповіді на запитання

1°. Які розгалуження називають вкладеними?

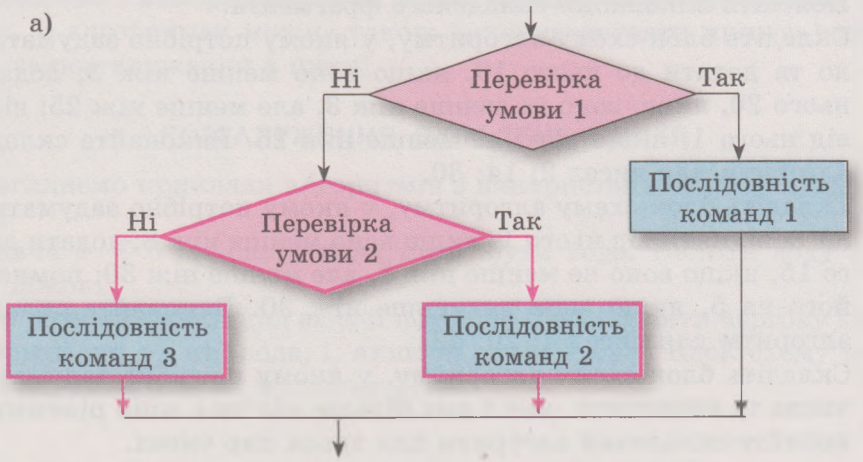
2•. Поясніть, як виконується таке вкладене розгалуження .

3•. Як виконується наведений фрагмент алгоритму?

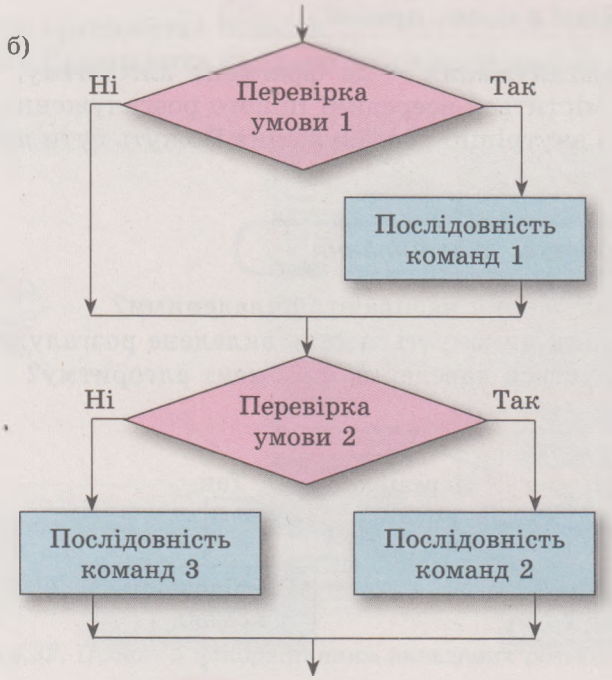


4•. Чим відрізняється виконання наведених фрагментів алгоритмів?

а)



б)



Виконайте завдання

1°. Складіть блок-схему фрагмента алгоритму з вкладеними розгалуженнями, кожне з яких є неповним. Поясніть виконання складеного фрагмента.

2°. Складіть блок-схему фрагмента алгоритму з вкладеними розгалуженнями, з яких внутрішнє є неповним, а зовнішнє – повним. Поясніть виконання складеного фрагмента.

3°. Складіть блок-схему фрагмента алгоритму з вкладеними розгалуженнями, з яких зовнішнє є неповним, а внутрішнє – повним. Поясніть виконання складеного фрагмента.

4°. Складіть блок-схему алгоритму, у якому потрібно задумати число та додати до нього 10, якщо воно менше ніж 3; додати до нього 20, якщо воно не менше ніж 3, але менше ніж 25; відняти від нього 1, якщо воно не менше ніж 25. Виконайте складений алгоритм для чисел 2; 14; 30.

5°. Складіть блок-схему алгоритму, у якому потрібно задумати число та відняти від нього 1, якщо воно менше ніж 8; додати до нього 15, якщо воно не менше ніж 8, але менше ніж 30; помножити його на 5, якщо воно не менше ніж 30. Виконайте складений алгоритм для чисел 4; 20; 52.

6°. Складіть блок-схему алгоритму, у якому потрібно задумати два числа та визначити, яке з них більше або чи є вони рівними. Виконайте складений алгоритм для трьох пар чисел.

7°. Складіть проект, у якому виконавець намалює відрізок завдовжки 50 кроків синім кольором, якщо натиснуто клавішу **стрілка вправо**, або намалює відрізок завдовжки 100 кроків червоним кольором, якщо натиснуто клавішу **стрілка вниз**. Збережіть проект у вашій папці у файлі з іменем **завдання 3.3.7**.

8°. Складіть проект, у якому виконавець змінить свій образ, якщо натиснуто клавішу **стрілка вліво**, збільшить свій розмір на 20, якщо натиснуто клавішу **стрілка вправо**, або зіграє звук на барабані, якщо натиснуто клавішу **пропуск**. Збережіть проект у вашій папці у файлі з іменем **завдання 3.3.8**.

9°. Складіть проект, у якому зміниться тло, якщо натиснуто клавішу **пропуск**, виконавець змінить образ, якщо натиснуто клавішу **стрілка вгору**, прозвучить нота **ля**, якщо натиснуто клавішу **стрілка вправо**. Збережіть проект у вашій папці у файл з іменем **завдання 3.3.9**.

10°. Складіть проект, у якому виконавець зменшить свій образ на 10, якщо до виконавця **доторкається вказівник миші**, переміститься вправо на 100, якщо натиснуто клавішу **стрілка вправо**, або зіграє ноту **до**, якщо натиснуто клавішу **пропуск**. Збережіть проект у вашій папці у файлі з іменем **завдання 3.3.10**.