

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

Додаток 1

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації
і управління

Звіт

з лабораторної роботи №5
з дисципліни «Алгоритми та структури даних-1.
Основи алгоритмізації»
«Дослідження арифметичних циклічних алгоритмів»
Варіант 6

Виконав
студент

ІП-15, Волинець Кирило Михайлович

(шифр, прізвище, ім'я, по батькові)

Перевірів

Всечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

Лабораторна робота 5

Дослідження арифметичних циклічних алгоритмів

Мета - дослідити особливості роботи арифметичних циклів та набути практичних навичок їх використання під час складання програмних специфікацій.

Індивідуальне завдання

Варіант 6

Дано натуральне число n . Перевірити, чи можна подати $n!$ у вигляді добутку трьох послідовних цілих чисел.

Побудова математичної моделі.

Складемо таблицю імен змінних.

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Початкове число	Цілочисельний	n	Вхідні
Ітератор	Цілочисельний	i	Проміжні
Факторіал числа	Дійсний	n_fact	Проміжні
Друге число послідовності	Цілочисельний	$a2$	Проміжні
Результат перевірки на відповідність умові	Логічний	$is_suitable$	Результат
Округлення вгору	Цілочисельний (функція)	$round_up()$	Обчислення
Знаходження остачі від ділення	Операнд	$\%$	Обчислення
Возведення у степінь	Операнд	\wedge	Обчислення

Розглянемо число, що є добутком 3-ох натуральних чисел. Це число також рівне кубу середнього геометричного цих 3-ох чисел. А середнє геометричне завжди менше або рівне за середнє арифметичне, що є другим числом послідовності. Тому, знайшовши середнє геометричне (кубічний корінь добутку), та округливши його в більшу сторону можна знайти потенційне друге число серед послідовних $a2$. Залишається тільки перевірити інші 2 числа. Знайти факторіал (n_fact) числа n можна класичним способом – за допомогою циклу (i).

Розв'язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

Крок 1. Знайдемо факторіал числа n

Крок 2. Знайдемо потенційне друге число

Крок 3. Перевіримо на відповідність умові (представлення у вигляді добутку 3-ох послідовних цілих чисел) та виведемо результат

Псевдокод:

Крок 1:

початок

введення n

$n_fact = 1$

повторити для i від 1 до $i \leq n$:

$n_fact = n_fact * i$

все повторити

Знайти потенційне друге число

Перевірити умову

Вивести результат

кінець

Крок 2:

початок

введення n

$n_fact = 1$

повторити для i від 1 до $i \leq n$:

$n_fact = n_fact * i$

все повторити

$a2 = \text{round_up}(n_fact^{1/3})$

Перевірити умову

кінець

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

Крок 3:

початок

введення n

n_fact = 1

повторити для i від 1 до i ≤ n:

 n_fact = n_fact * i

все повторити

a2 = round_up(n_fact^{1/3})

is_suitable = true

якщо n_fact % (a2 + 1) ≠ 0:

 is_suitable = false

все якщо

інакше якщо n_fact % a2 ≠ 0:

 is_suitable = false

все якщо

інакше якщо n_fact % (a2 - 1) ≠ 0:

 is_suitable = false

все якщо

якщо is_suitable = true:

 вивести “Так, можна”

все якщо

інакше:

 вивести “Ні, не можна”

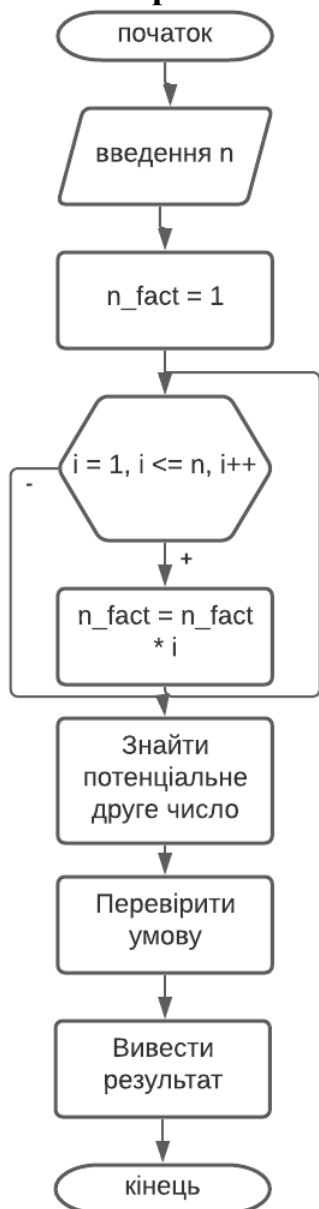
все інакше

кінець

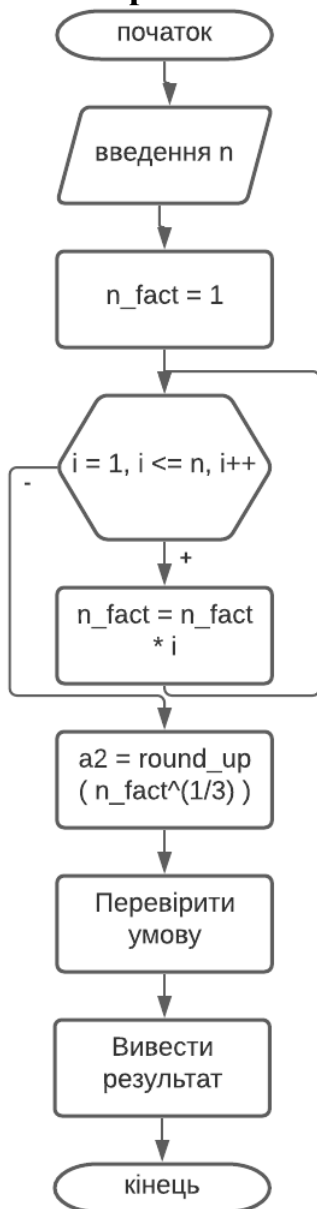
Основи алгоритмізації

Блок-схема:

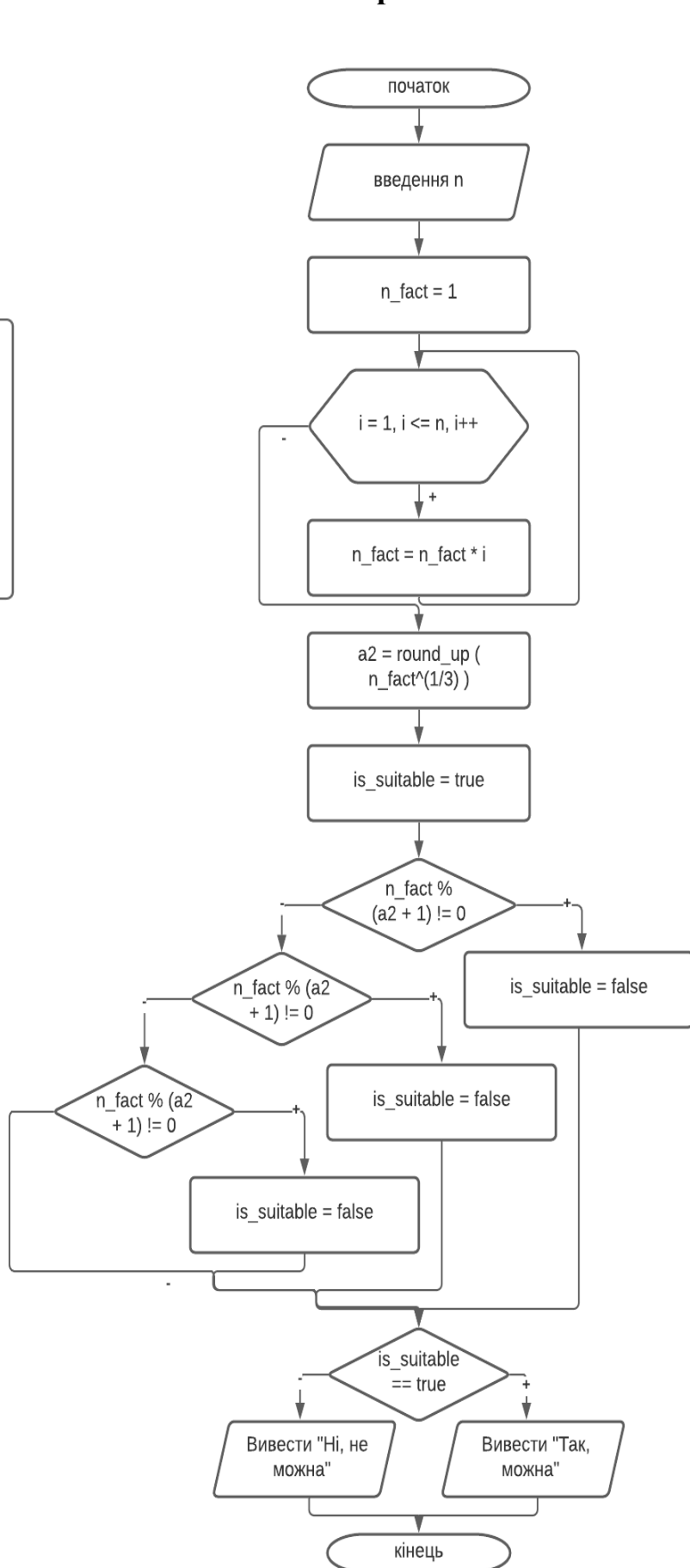
Крок 1:



Крок 2:



Крок 3:



Алгоритми та структури даних.

Основи алгоритмізації

Перевірка

	Випадок 1	Випадок 2	Випадок 3
1	початок	початок	початок
2	Введення $n = 1$	Введення $n = 6$	Введення $n = 7$
5	$n_fact = 1 * 1 = 1$	$n_fact = 1 * 1 = 1$ $n_fact = 1 * 2 = 2$ $n_fact = 2 * 3 = 6$ $n_fact = 6 * 4 = 24$ $n_fact = 24 * 5 = 120$ $n_fact = 120 * 6 = 720$	$n_fact = 1 * 1 = 1$ $n_fact = 1 * 2 = 2$ $n_fact = 2 * 3 = 6$ $n_fact = 6 * 4 = 24$ $n_fact = 24 * 5 = 120$ $n_fact = 120 * 6 = 720$ $n_fact = 720 * 7 = 5040$
6	$a2 = \text{round_up}(1^{(1/3)})$ $a2 = \text{round_up}(1)$ $a2 = 1$	$a2 = \text{round_up}(720^{(1/3)})$ $a2 = \text{round_up}(8.96281)$ $a2 = 9$	$a2 = \text{round_up}(5040^{(1/3)})$ $a2 = \text{round_up}(17.14524)$ $a2 = 18$
7	$n_fact \% (a2 + 1) = 1 \% 2 = 1$ $is_suitable = false$	$n_fact \% (a2 + 1) = 720 \% 10 = 0$ $is_suitable = true$	$n_fact \% (a2 + 1) = 5040 \% 19 = 5$ $is_suitable = false$
9		$n_fact \% a2 = 720 \% 9 = 0$ $is_suitable = true$	
11		$n_fact \% (a2 - 1) = 720 \% 8 = 0$ $is_suitable = true$	
13	Виведення “Ні, не можна”	Виведення “Так, можна”	Виведення “Ні, не можна”
14	кінець	кінець	кінець

Висновки

Ми дослідили особливості роботи арифметичних циклів та набули практичних навичок їх використання під час складання програмних специфікацій. В результаті виконання лабораторної роботи ми отримали алгоритм для визначення можливості розкладення факторіала числа на добуток 3-ох послідовних цілих чисел, декомпозиували задачу на 3 кроки: знайшли факторіал числа, потенційне друге число послідовності, та перевірили факторіал на виконання умови. В процесі випробування ми розглянули випадки з мінімальним значенням, значенням з вірною умовою і без, та отримали вірні результати.