

kozaknazar.github.io

vns.lpnu.ua/

q5.2.1

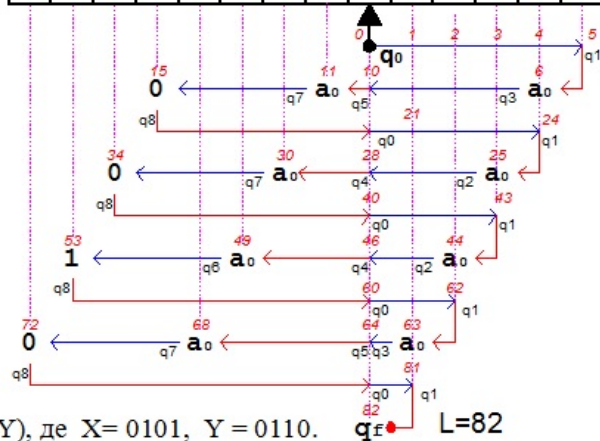
Задано прграму для машини Тюрінга:

A ^Q	q ₀	q ₁	q ₂	q ₃	q ₄	q ₅	q ₆	q ₇	q ₈
a ₀	Lq ₁				Lq ₄	Lq ₅	1Rq ₈	0Rq ₈	Rq ₈
0	Rq ₀	a ₀ Lq ₃	Lq ₂	Lq ₃	a ₀ Lq ₇	a ₀ Lq ₇	Lq ₆	Lq ₇	Rq ₈
1	Rq ₀	a ₀ Lq ₂	Lq ₂	Lq ₃	a ₀ Lq ₆	a ₀ Lq ₇	Lq ₆	Lq ₇	Rq ₈
^	Rq ₀	q _f	Lq ₄	Lq ₅					Rq ₀

P = 29

			a ₀	0	1	0	1	^	0	1	1	0	a ₀
--	--	--	----------------	---	---	---	---	---	---	---	---	---	----------------

M=14



(X ∧ Y), де X= 0101, Y = 0110. L=82

Яка програмна складність алгоритму?

Відповідь

q5.2.2

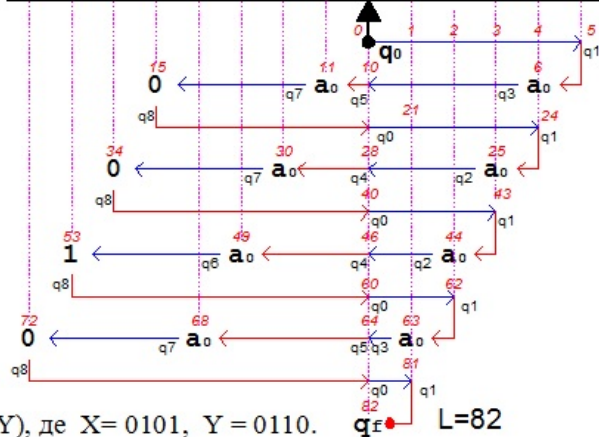
Задано прграму для машини Тюрінга:

Q A	q ₀	q ₁	q ₂	q ₃	q ₄	q ₅	q ₆	q ₇	q ₈
a ₀	Lq ₁				Lq ₄	Lq ₅	1Rq ₈	0Rq ₈	Rq ₈
0	Rq ₀	a ₀ Lq ₃	Lq ₂	Lq ₃	a ₀ Lq ₇	a ₀ Lq ₇	Lq ₆	Lq ₇	Rq ₈
1	Rq ₀	a ₀ Lq ₂	Lq ₂	Lq ₃	a ₀ Lq ₆	a ₀ Lq ₇	Lq ₆	Lq ₇	Rq ₈
^	Rq ₀	q _f	Lq ₄	Lq ₅					Rq ₀

P = 29

			a ₀	0	1	0	1	^	0	1	1	0	a ₀
--	--	--	----------------	---	---	---	---	---	---	---	---	---	----------------

M=14



(X ^ Y), де X= 0101, Y = 0110. L=82

Яка часова складність алгоритму?

Відповідь

q5.2.3

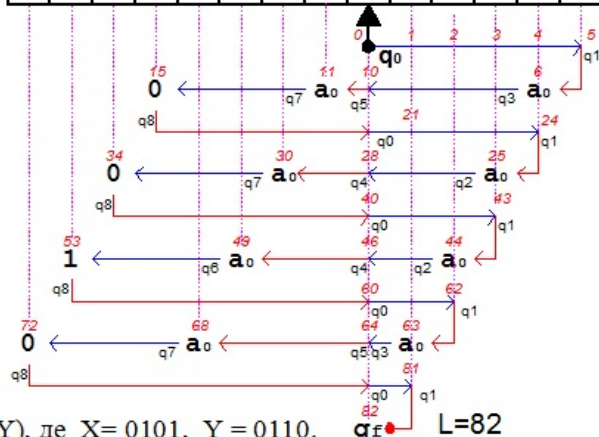
Задано прграму для машини Тюрінга:

Q A	q ₀	q ₁	q ₂	q ₃	q ₄	q ₅	q ₆	q ₇	q ₈
a ₀	Lq ₁				Lq ₄	Lq ₅	1Rq ₈	0Rq ₈	Rq ₈
0	Rq ₀	a ₀ Lq ₃	Lq ₂	Lq ₃	a ₀ Lq ₇	a ₀ Lq ₇	Lq ₆	Lq ₇	Rq ₈
1	Rq ₀	a ₀ Lq ₂	Lq ₂	Lq ₃	a ₀ Lq ₆	a ₀ Lq ₇	Lq ₆	Lq ₇	Rq ₈
^	Rq ₀	q _f	Lq ₄	Lq ₅					Rq ₀

P = 29

			a ₀	0	1	0	1	^	0	1	1	0	a ₀
--	--	--	----------------	---	---	---	---	---	---	---	---	---	----------------

M=14



(X ^ Y), де X= 0101, Y = 0110. L=82

Яка місткісна складність алгоритму?

Відповідь

q5.1.1

Над словами з алфавіту $V = \{a, b, c\}$ Задано нормальний алгоритм Маркова з такими формулами підстановки:

P1 : "ab" → "b",

P2 : "ac" → "c",

P3 : "aa" → "a".

Якою буде результуюча стрічка після виконання алгоритму, якщо вхідний рядок має вигляд "басаабаа".

(*відповідь давати латинськими літерами в нижньому регістрі без використання лапок)

Відповідь

q7.1

Для алгоритму, що заданий за допомогою функції мовою C, розрахувати обчислювальну складність для «найгіршого випадку».

```
void bulbSort(int * array, int leftIndex, int rightIndex){  
    if(!array){  
        return;  
    }  
    ++rightIndex;  
    while(leftIndex < --rightIndex){  
        for(unsigned int index = leftIndex; index < rightIndex; ++index){  
            if(array[index] > array[index + 1]){  
                int temp = array[index];  
                array[index] = array[index + 1];  
                array[index + 1] = temp;  
            }  
        }  
    }  
}
```

☐

$O(n*n)$

☐

$O(n)$

☐

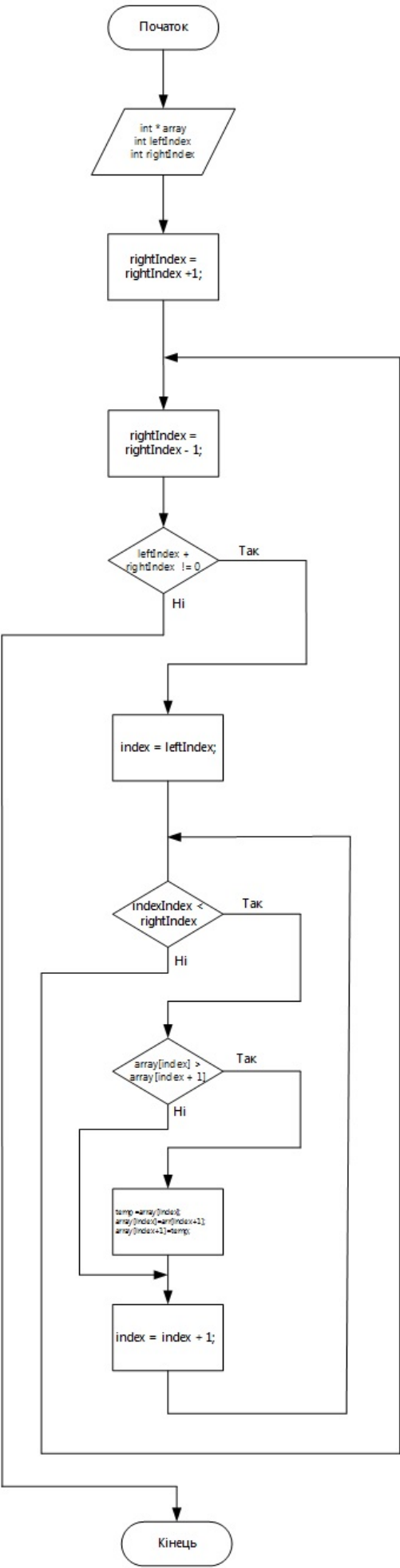
$O(2n)$

☐

$O(4n)$

q7.2

Задано блок-схему алгоритму. Розрахувати обчислювальну складність цього алгоритму для «найгіршого випадку».



$O(n*n)$

☐

$O(n)$

☐

$O(2n)$

☐

$O(4n)$

t4.1

Скласти програму (C/C++), що виконує множення матриць.

Розмір першої матриця		Розмір другої матриця	
кількість рядків	кількість стовпців	кількість рядків	кількість стовпців
4	5	5	4

t5.2

Скласти програму (C/C++), що виконує множення матриць.

Розмір першої матриця		Розмір другої матриця	
кількість рядків	кількість стовпців	кількість рядків	кількість стовпців
5	4	4	5

t5.1

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'A'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.10

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'J'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.11

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'K'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.12

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'W'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.13

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'M'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.14

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'N'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.15

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'O'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.16

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'P'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.17

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'Q'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.18

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'R'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.19

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'S'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.2

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'B'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.20

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'T'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.3

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'C'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.4

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'D'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.5

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'E'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.6

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'F'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.7

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'G'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.8

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'H'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.1.9

Застосовуючи STL скласти програму (C/C++), яка за допомогою **`std::sort`** дозволяє з вхідного тексту(**`std::string`**) вивести слова, що починаються з латинської літери 'V'. Сортювання потрібно виконати без використання додаткової пам'яті, дозволяється використовувати тільки **`std::vector`**, що зберігає індекси на початок слів. Додатково потрібно зберегти результати роботи програми у **`std::map`**.

Індекси із **`std::vector`** вивести на екран за допомогою **`std::copy`**, а слова за допомогою звичайного циклу **`for`**. Для виводу даних з **`std::map`** застосувати **`std::transform`**.

t5.2.1

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **`uBLAS`**(з набору бібліотек **`Boost`**):

$$\begin{cases} 5x - y + z = 0 \\ 3x + y + 2z = -1 \\ x + y + z = 0 \end{cases}$$

t5.2.10

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 2x - 5y - z = 8 \\ x - y + 3z = 3 \\ 3x - 4y - 2z = 7 \end{cases}$$

t5.2.11

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 2x + y - 3z = -1 \\ 3x + y - z = 1 \\ 5x + 2y + 4z = 0 \end{cases}$$

t5.2.12

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 3x + y + 2z = 2 \\ 2x + 5y + 4z = 5 \\ 5x - y + 2z = 2 \end{cases}$$

t5.2.13

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 2x - y - z = 5 \\ 3x - 5y - z = 5 \\ x + y + z = 7 \end{cases}$$

t5.2.14

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 5x - 3y - 5z = 8 \\ 3x + 7y - 3z = 10 \\ x + 2y + 5z = 5 \end{cases}$$

t5.2.15

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 5x + 5y + 2z = 1 \\ x + 2y + z = 4 \\ 3x + 7y - z = 2 \end{cases}$$

t5.2.16

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 2x + y + z = 8 \\ 3x + 2y - 5z = 4 \\ 2x + 3y - z = 0 \end{cases}$$

t5.2.17

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} x - 3y - z = 0 \\ 2x - 5y + z = 5 \\ 3x - 7y - 5z = 2 \end{cases}$$

t5.2.18

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 7x + y - z = 3 \\ 2x + 5y - z = 2 \\ 5x - y - z = 1 \end{cases}$$

t5.2.19

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 5x + y + z = 6 \\ 2x - 3y - 2z = 2 \\ 3x + 2y + 2z = 5 \end{cases}$$

t5.2.2

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 3x - 5y + z = 1 \\ 2x + y + 2z = 5 \\ x - y + z = 3 \end{cases}$$

t5.2.20

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 3x - 10y + 2z = 5 \\ 5x - y - 9z = 24 \\ 2x + 3y + 6z = 13 \end{cases}$$

t5.2.3

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 5x + 7y + 4z = 3 \\ 3x - 5y + 2z = 3 \\ 2x + y + z = 3 \end{cases}$$

t5.2.4

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 7x + y - z = 2 \\ 5x + 2y + z = 10 \\ 2x + 3y - 5z = 2 \end{cases}$$

t5.2.5

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 3x + y + 2z = 6 \\ 2x - y + 7z = 7 \\ x + y - 3z = -2 \end{cases}$$

t5.2.6

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 2x - 3y + z = 10 \\ x + y + 4z = 6 \\ x - y - 2z = 2 \end{cases}$$

t5.2.7

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 3x + 2y - 2z = 1 \\ x + y + 3z = 6 \\ x - 2y - 7z = 1 \end{cases}$$

t5.2.8

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 7x + y - z = 2 \\ 3x + 2y - 5z = 1 \\ 3x - y + 5z = 2 \end{cases}$$

t5.2.9

Написати на C++ програму для розв'язання заданої системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**):

$$\begin{cases} 5x + 3y - 4z = 8 \\ x - 3y + 5z = 6 \\ 2x - y + z = 4 \end{cases}$$

t6.1

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 125, 126, 121, 121, 117, 120, 121, 116, 116, 112, 119, 111, 116, 117, 112, 114, 106, 111, 112, 107 }

t6.10

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.11

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.12

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.13

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.14

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.15

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.16

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.17

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.18

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.19

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.2

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.20

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.21

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.22

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.23

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.24

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.25

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.26

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.27

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.28

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.29

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.3

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.30

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.4

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.5

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.6

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.7

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.8

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

t6.9

Застосовуючи парадигму функційного програмування скласти програму мовою C++, яка виконує імплементацію швидкого сортування без використання готових бібліотечних реалізацій.

** акцентується увага на тому, що завдання має бути виконано відповідно до парадигми функційного програмування*

Показати сортування такого масиву:

{ 145, 146, 141, 141, 137, 140, 141, 136, 136, 132, 139, 131, 136, 137, 132, 134, 126, 131, 132, 12 }

q1.1

Всього можна виділити чотири базових структури алгоритмів:

- лінійні;
- розгалужені;
- циклічні;
- змішані.

- ☐ Правильно
☐ Неправильно

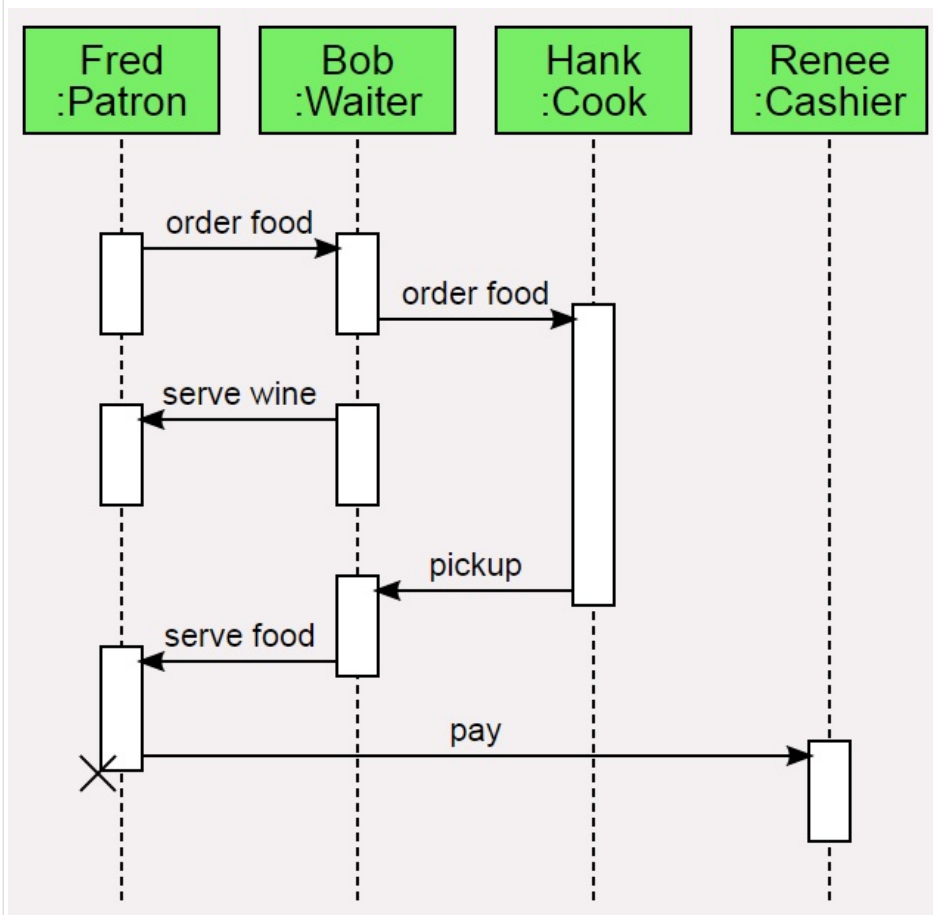
q1.2

Алгоритм – точний припис, який задає обчислювальний процес (що називається в цьому випадку алгоритмічним), що починається з довільного початкового даного (з деякої сукупності можливих для даного алгоритму початкових даних) і спрямований на отримання результату, який повністю визначається цим початковим даним

Для реалізації абстрактного алгоритму використовується уявний обчислювач, у якому не декларується апаратних засобів.

- ☐ Правильно
☐ Неправильно

q3.1



На рисунку зображено:

☐

UML-діаграму послідовності

☐

UML-діаграму прецедентів

☐

UML-діаграму станів

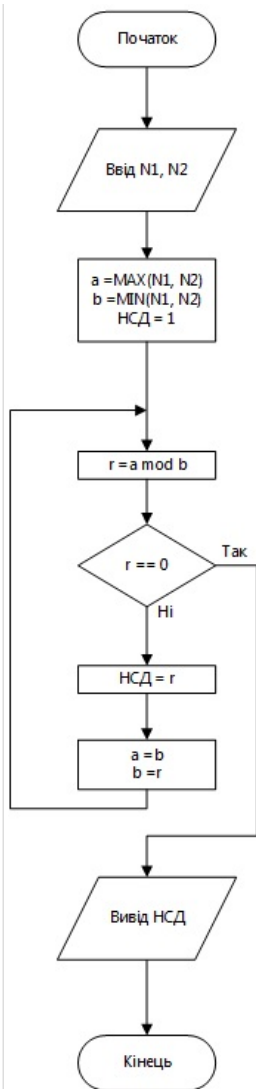
☐

UML-діаграму діяльності(UML-діаграму активності)

☐

Блок-схему алгоритму

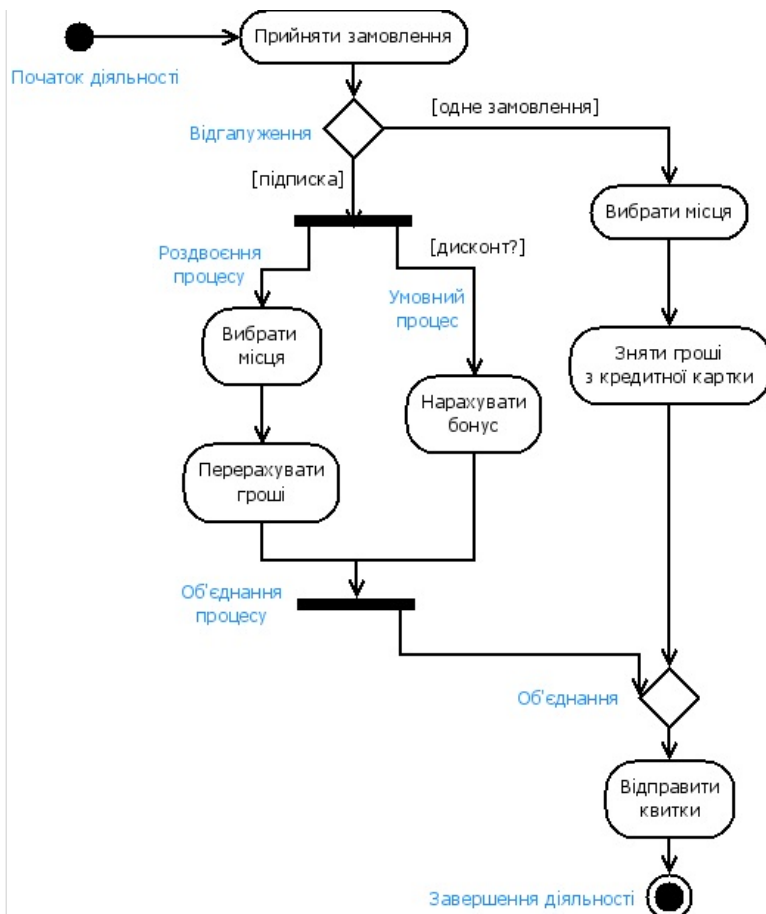
q3.2



На рисунку зображено:

- ☐ UML-діаграму послідовності
- ☐ UML-діаграму прецедентів
- ☐ UML-діаграму станів
- ☐ UML-діаграму діяльності(UML-діаграму активності)
- ☐ Блок-схему алгоритму

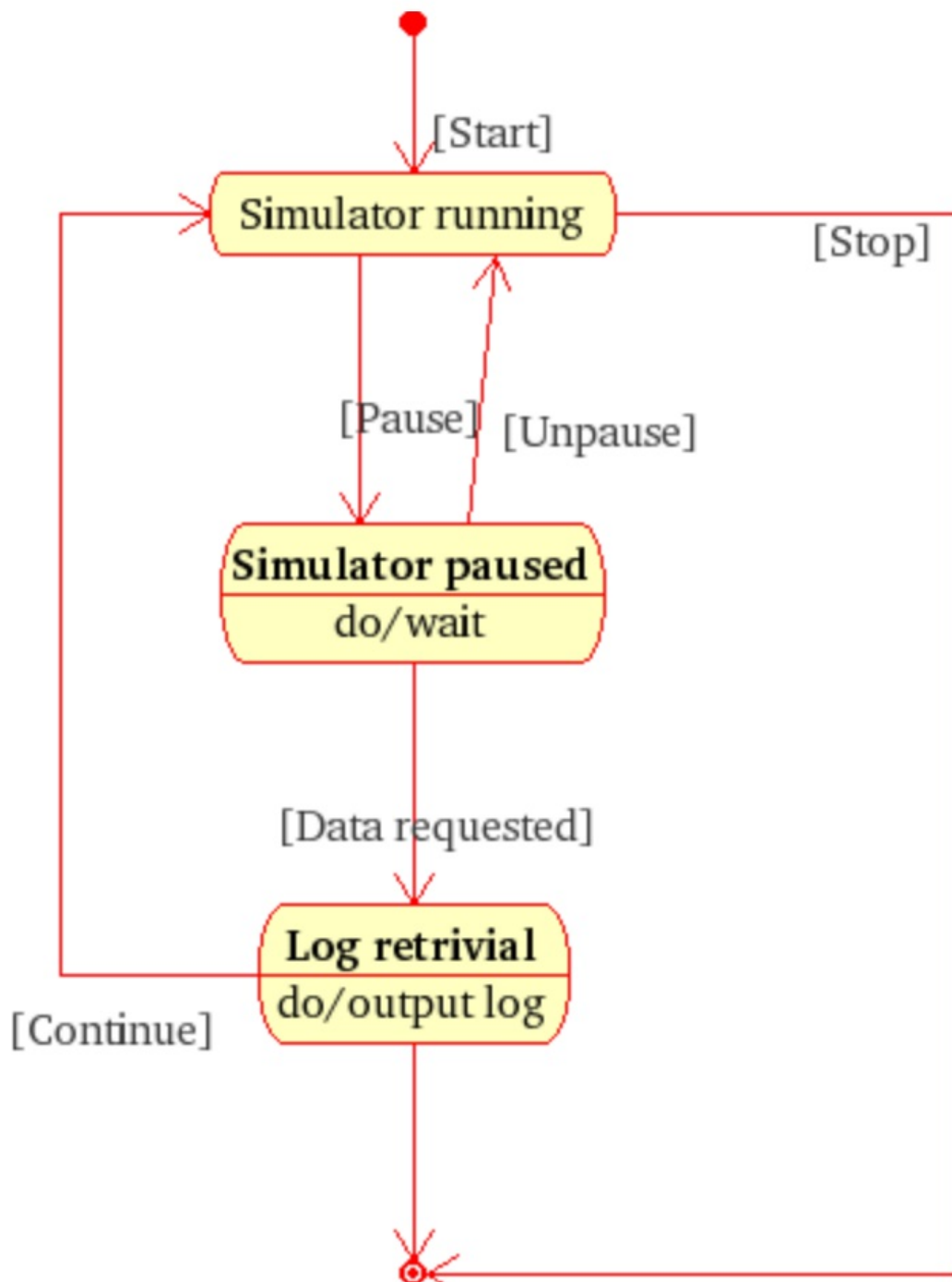
q3.3



На рисунку зображено:

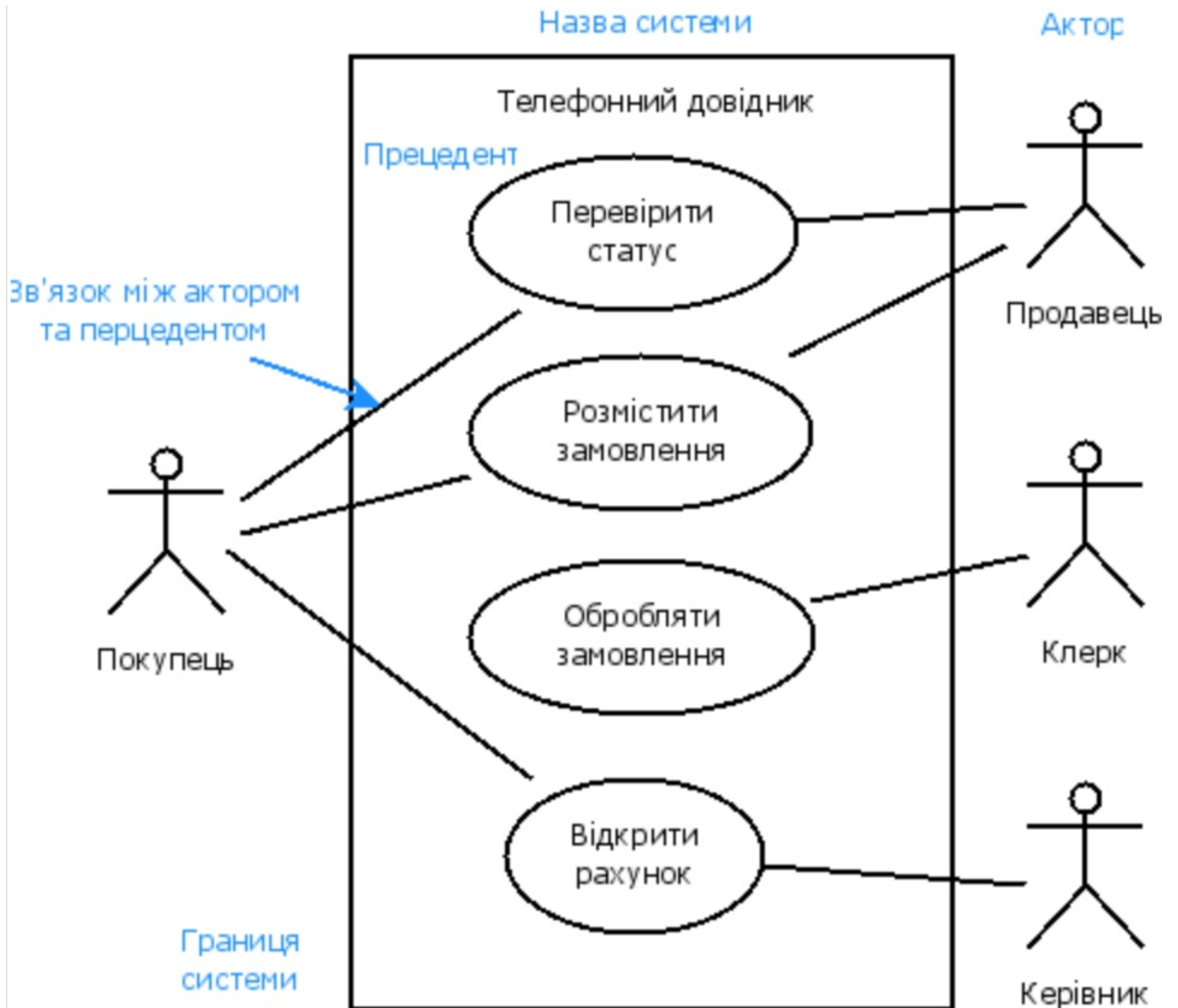
- ☐ UML-діаграму послідовності
- ☐ UML-діаграму прецедентів
- ☐ UML-діаграму станів
- ☐ UML-діаграму діяльності(UML-діаграму активності)
- ☐ Блок-схему алгоритму

q3.4



На рисунку зображено:

- ☐ UML-діаграму послідовності
- ☐ UML-діаграму прецедентів
- ☐ UML-діаграму станів
- ☐ UML-діаграму діяльності(UML-діаграму активності)
- ☐ Блок-схему алгоритму



На рисунку зображено:

- ☐ UML-діаграму послідовності
- ☐ UML-діаграму прецедентів
- ☐ UML-діаграму станів
- ☐ UML-діаграму діяльності(UML-діаграму активності)
- ☐ Блок-схему алгоритму

qба.1.1

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:

одна латинська літера в нижньому регістрі

q6a.1.2

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
одна латинська літера в верхньому регістрі

q6a.2.1

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
дві латинські літери в нижньому регістрі

q6a.2.2

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
дві латинські літери в верхньому регістрі

q6a.3.1

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
три латинські літери в нижньому регістрі

q6a.3.2

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
три латинські літери в верхньому регістрі

q6a.4.1

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
чотири латинські літери в нижньому регістрі

q6a.4.2

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
чотири латинські літери в верхньому регістрі

q6a.5.1

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
п'ять латинських літер в нижньому регістрі

q6a.5.2

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
п'ять латинських літер в верхньому регістрі

q6a.6.1

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
шість латинських літер в нижньому регістрі

q6a.6.2

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
шість латинських літер в верхньому регістрі

q6a.7.1

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
сім латинських літер в нижньому регістрі

q6a.7.2

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
сім латинських літер в верхньому регістрі

q6a.8.1

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
вісім латинських літер в нижньому регістрі

q6a.8.2

З використанням **std::regex** написати програму, яка шукає в тексті лексеми такого формату:
вісім латинських літер в верхньому регістрі

q6b.1

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

$$\langle \text{expression} \rangle = \langle \text{term_a} \rangle, \{ "+", \langle \text{term_a} \rangle \}^*.$$
$$\langle \text{term_a} \rangle = \langle \text{term_m} \rangle, \{ ("*" | "\%"), \langle \text{term_m} \rangle \}^*.$$
$$\langle \text{term_m} \rangle = \langle \text{value} \rangle | "+" \langle \text{term_m} \rangle | "-" \langle \text{term_m} \rangle | "(", \langle \text{expression} \rangle, ")".$$

q6b.10

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

$$\langle \text{expression} \rangle = \langle \text{term_a} \rangle, \{ "-", \langle \text{term_a} \rangle \}^*.$$
$$\langle \text{term_a} \rangle = \langle \text{term_m} \rangle, \{ ("*" | "/"), \langle \text{term_m} \rangle \}^*.$$
$$\langle \text{term_m} \rangle = \langle \text{value} \rangle | "+" \langle \text{term_m} \rangle | "-" \langle \text{term_m} \rangle | "(", \langle \text{expression} \rangle, ")".$$

q6b.2

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

$$\langle \text{expression} \rangle = \langle \text{term_a} \rangle, \{ "-", \langle \text{term_a} \rangle \}^*.$$
$$\langle \text{term_a} \rangle = \langle \text{term_m} \rangle, \{ ("*" | "\%"), \langle \text{term_m} \rangle \}^*.$$
$$\langle \text{term_m} \rangle = \langle \text{value} \rangle | "+" \langle \text{term_m} \rangle | "-" \langle \text{term_m} \rangle | "(", \langle \text{expression} \rangle, ")".$$

q6b.3

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

$$\langle \text{expression} \rangle = \langle \text{term_a} \rangle, \{ "+", \langle \text{term_a} \rangle \}^*.$$
$$\langle \text{term_a} \rangle = \langle \text{term_m} \rangle, \{ ("/" | "\%"), \langle \text{term_m} \rangle \}^*.$$
$$\langle \text{term_m} \rangle = \langle \text{value} \rangle | "+" \langle \text{term_m} \rangle | "-" \langle \text{term_m} \rangle | "(", \langle \text{expression} \rangle, ")".$$

q6b.4

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

$$\langle \text{expression} \rangle = \langle \text{term_a} \rangle, \{ "-", \langle \text{term_a} \rangle \}^*.$$
$$\langle \text{term_a} \rangle = \langle \text{term_m} \rangle, \{ ("/" | "\%"), \langle \text{term_m} \rangle \}^*.$$
$$\langle \text{term_m} \rangle = \langle \text{value} \rangle | "+" \langle \text{term_m} \rangle | "-" \langle \text{term_m} \rangle | "(", \langle \text{expression} \rangle, ")".$$

q6b.5

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

$$\langle \text{expression} \rangle = \langle \text{term_a} \rangle, \{ "+", \langle \text{term_a} \rangle \}^*.$$
$$\langle \text{term_a} \rangle = \langle \text{term_m} \rangle, \{ ("*" | "\&"), \langle \text{term_m} \rangle \}^*.$$
$$\langle \text{term_m} \rangle = \langle \text{value} \rangle | "+" \langle \text{term_m} \rangle | "-" \langle \text{term_m} \rangle | "(", \langle \text{expression} \rangle, ")".$$

q6b.6

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

```
<expression> = <term_a>, { "-", <term_a> }*.
<term_a>      = <term_m>, { ("*" | "&"), <term_m> }*.
<term_m>      = <value> | "+" <term_m> | "-" <term_m> | "(", <expression>, ")".
```

q6b.7

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

```
<expression> = <term_a>, { "+", <term_a> }*.
<term_a>      = <term_m>, { ("/" | "&"), <term_m> }*.
<term_m>      = <value> | "+" <term_m> | "-" <term_m> | "(", <expression>, ")".
```

q6b.8

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

```
<expression> = <term_a>, { "-", <term_a> }*.
<term_a>      = <term_m>, { ("/" | "&"), <term_m> }*.
<term_m>      = <value> | "+" <term_m> | "-" <term_m> | "(", <expression>, ")".
```

q6b.9

Розробити програму, яка за допомогою **boost::spirit** реалізовує заданий синтаксис для запису виразу

```
<expression> = <term_a>, { "+", <term_a> }*.
<term_a>      = <term_m>, { ("*" | "/"), <term_m> }*.
<term_m>      = <value> | "+" <term_m> | "-" <term_m> | "(", <expression>, ")".
```

q4.1.1

На практиці визначення порядку зростання є задачею, що цілком вирішується за допомогою кількох базових принципів. Існують три правила для визначення складності:

1. $O(c * f(n)) = O(f(n))$
2. $O(f(n) + g(n)) = O(\max(f(n), g(n)))$
3. $O(f(n) * g(n)) = O(f(n)) * O(g(n))$

Перше правило декларує, що постійні множники не мають значення для визначення порядку зростання.

- ☐ Правильно
☐ Неправильно

q4.1.2

На практиці визначення порядку зростання є задачею, що цілком вирішується за допомогою кількох базових принципів. Існують три правила для визначення складності:

1. $O(c * f(n)) = O(f(n))$
2. $O(f(n) + g(n)) = O(\max(f(n), g(n)))$
3. $O(f(n) * g(n)) = O(f(n)) * O(g(n))$

Друге правило називається **"Правило сум"**. Це правило використовується для послідовних програмних фрагментів з циклами та розгалуженнями. Порядок зростання скінченної послідовності програмних фрагментів (без врахування констант) дорівнює порядку зростання фрагменту з найбільшою часовою складністю. Якщо

алгоритм складається з двох фрагментів, функції часових складностей яких $L_1(n)$ і $L_2(n)$ мають ступені зростання $O(f(n))$ і $O(g(n))$ відповідно, то алгоритм має степінь зростання $O(\max(f(n), g(n)))$.

- ☐ Правильно
☐ Неправильно

q4.1.3

На практиці визначення порядку зростання є задачею, що цілком вирішується за допомогою кількох базових принципів. Існують три правила для визначення складності:

1. $O(c * f(n)) = O(f(n))$
2. $O(f(n) + g(n)) = O(\max(f(n), g(n)))$
3. $O(f(n) * g(n)) = O(f(n)) * O(g(n))$

Третє правило називається "**Правило добутків**". Якщо $L_1(n)$ і $L_2(n)$ мають ступені зростання $O(f(n))$ і $O(g(n))$ відповідно, то добуток $L_1(n) L_2(n)$ має степінь зростання $O(f(n)g(n))$. Прикладом може бути фрагмент програми "цикл в циклі".

- ☐ Правильно
☐ Неправильно

q4.2.1

Для опису швидкості зростання функцій використовується O-символіка. Функція $f(n)$ має порядок зростання $O(g(n))$, якщо існують додатні константи C і n_0 такі, що:

$$f(n) \leq C * g(n), \quad \text{для } n > n_0.$$

- ☐ Правильно
☐ Неправильно

q4.2.2

Щоби вказати нижню границю швидкості зростання $L(n)$ використовують позначення $W(g(n))$, що означає існування такої константи c , що для нескінченної кількості значень n виконується нерівність $L(n) \geq c * g(n)$.

- ☐ Правильно
☐ Неправильно

q4.3.1

В алгоритмах з експоненціальною складністю кількість операцій, необхідних для розв'язання задачі, зростає швидше, ніж поліном k -ї степені при зростанні розміру входу.

- ☐ Правильно
☐ Неправильно

q2.1.1

Наведений запис це нотація Бекуса-Наура (англ. *Backus-Naur form*, *BNF*):

<поштова-адреса> ::= <поштове-відділення> <вулична-адреса> <особа>

<поштове-відділення> ::= <індекс> " , " <місце> <EOL>

<місце> ::= <село> | <місто>

<вулична-адреса> ::= <вулиця> " , " <будинок> <EOL>

<особа> ::= <прізвище> <ім'я> <EOL> | <прізвище> <ім'я> <по батькові> <EOL>

- ☐ Правильно
☐ Неправильно

q2.1.2

Наведений запис це розширена нотація Бекуса-Наура (*англ. extended Backus–Naur form, EBNF*):

Integer = Sign UnsignedInteger.

UnsignedInteger = digit, {digit}.

Sign = ["+" | "-"].

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9".

- ☐ Правильно
☐ Неправильно

q8.1

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} (8 * a - b) / a + 1, & a > b, \\ 25, & a = b, \\ (a - 5) / b, & a < b. \end{cases}$$

q8.10

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} -7 * b / a + 1, & a > b, \\ -271, & a = b, \\ (a - b) / b, & a < b. \end{cases}$$

q8.11

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} a/b + 31, & a > b, \\ -9, & a = b, \\ (5 * b - 1)/a, & a < b. \end{cases}$$

q8.12

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} (2 + b)/a, & a > b, \\ -2, & a = b, \\ (a - 5)/2 * b, & a < b. \end{cases}$$

q8.13

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 5 * b/a + 1, & a < b, \\ 12, & a = b, \\ (a - 5)/b, & a > b. \end{cases}$$

q8.14

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} a/b + 1, & a > b, \\ -2, & a = b, \\ 2 * (a - b)/a, & a < b. \end{cases}$$

q8.15

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} (3 * a - 5) / b, & a < b, \\ -4, & a = b, \\ (a + b) / a, & a > b. \end{cases}$$

q8.16

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} -4 * b / a - 1, & a < b, \\ -295, & a = b, \\ (a - 235) / b, & a > b. \end{cases}$$

q8.17

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 2 * a / b + 1, & a > b, \\ -445, & a = b, \\ (b + 5) / a, & a < b. \end{cases}$$

q8.18

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 5 * a / b - 4, & a < b, \\ -55, & a = b, \\ (b - 5) / a, & a > b. \end{cases}$$

q8.19

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} -4 * b / a + 2, & a > b, \\ -57, & a = b, \\ (a - b) / b, & a < b. \end{cases}$$

q8.2

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} (a-b)/a-3, & a > b, \\ 2, & a = b, \\ (4*a+1)/b, & a < b. \end{cases}$$

q8.20

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 6*a/b-37, & a > b, \\ 3, & a = b, \\ (a-b)/a, & a < b. \end{cases}$$

q8.21

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 3*a/b+1, & a > b, \\ a+25, & a = b, \\ (b-2)/a, & a < b. \end{cases}$$

q8.22

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} (4*a-b)/b, & a > b, \\ -a, & a = b, \\ (b-a)/a, & a < b. \end{cases}$$

q8.23

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на

нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} (6 * b + 1) / a, & a > b, \\ -b, & a = b, \\ (a - 5) / b, & a < b. \end{cases}$$

q8.24

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} a/b - 1, & a < b, \\ 10 - a, & a = b, \\ (-2 * b - 5) / a, & a > b. \end{cases}$$

q8.25

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 5 * b / a + 2, & a > b, \\ -11, & a = b, \\ (a - 8) / b, & a < b. \end{cases}$$

q8.26

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 2 * a / b + 2, & a > b, \\ 8, & a = b, \\ (b - 9) / a, & a < b. \end{cases}$$

q8.27

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} -5 + b/a, & a > b \\ 45, & a = b, \\ (3 * a - 6)/b, & a < b. \end{cases}$$

q8.28

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 7 * a/b + 20, & a > b, \\ 110, & a = b, \\ (a - b)/a, & a < b. \end{cases}$$

q8.29

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} a/b + 5, & a < b, \\ -b, & a = b, \\ (2 * b - 9)/a, & a > b. \end{cases}$$

q8.3

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} -2 * b/a + 5, & a < b, \\ -5, & a = b, \\ (a - b)/b, & a > b. \end{cases}$$

q8.30

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double).Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} b/a - 2, & a > b, \\ -140, & a = b, \\ (-3 * a - 100)/b, & a < b. \end{cases}$$

q8.4

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 3 * a/b + 10, & a < b, \\ -51, & a = b, \\ (b - 4)/a, & a > b. \end{cases}$$

q8.5

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 7 * a/b - 1, & a > b, \\ -25, & a = b, \\ (b - 5)/a, & a > b. \end{cases}$$

q8.6

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} 52 * b/a + b, & a > b, \\ -125, & a = b, \\ (a - 5)/b, & a < b. \end{cases}$$

q8.7

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} (4 * b - 1)/a, & a > b, \\ 255, & a = b, \\ (a - 5)/b, & a < b. \end{cases}$$

q8.8

1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} a/b + 7, & a > b, \\ -125, & a = b, \\ (3 * b + 9)/a, & a < b. \end{cases}$$

q8.9

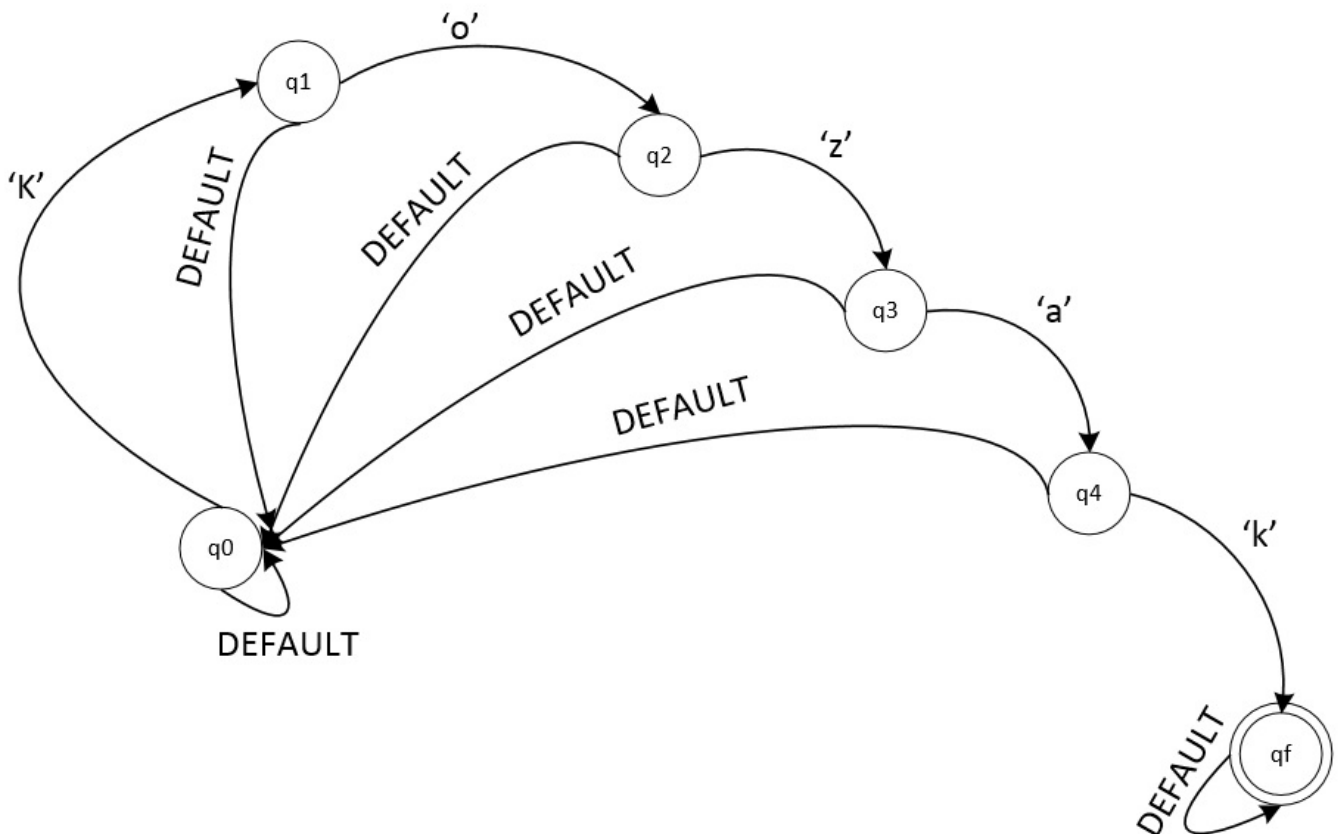
1) Написати програму мовою C, яка виконує ввід даних, обчислення арифметичного виразу та вивід даних. Вхідні дані слід вважати дійсними числами(double). Уникнути випадку некоректних обчислень при діленні на нуль.

2) Програма має також виводити інформацію про кількість елементів умови на блок-схемі алгоритму створеної програми.

$$X = \begin{cases} a/b + 11, & a > b, \\ -1, & a = b, \\ (2 * b - 9)/a, & a < b. \end{cases}$$

q5.3.1

Задано кінцевий автомат(анг. Finite State Automata):



Напишіть послідовність символів, які переведуть автомат в кінцевий стан

(*відповідь давати латинськими літерами з врахуванням регістру без використання лапок)

Відповідь