**Практична робота №2. Основи алгоритмізації обчислювальних процесів. Тематичний контроль. Висновки та узагальнення.**

**Опитування №1.**

1. Дайте визначення алгоритму та перелічите його властивості.
2. Як можна визначити виконавця алгоритму та систему його команд?
3. Перелічите правила побудови алгоритмів.
4. Яки види алгоритмів ви можете визначити?
5. В який спосіб можна записати алгоритм?
6. Які позначки використовуються при запису блок-схем?
7. Яким чином програма перетворюється в машинні коди?
8. Дайте визначення алгоритмізації.
9. Які процеси та етапи алгоритмізації?
10. В чому полягає декомпозиція?
11. Коли застосовується інтеграція?
12. Наведіть приклади алгоритму та алгоритмізації в повсякденному житті.
13. Для чого використовуються проміжні величини?
14. Що таке система числення? Які типи систем числення ви знаєте?
15. Яка система числення використовується для подання чисел у пам’яті комп’ютера? Чому?
16. Охарактеризуйте архітектуру сучасних ПК.
17. Назвіть основні етапи розвитку мов програмування.
18. Назвіть основні етапи розв’язування задач за допомогою комп’ютера.
19. Що таке «припустима елементарна операція»?
20. Як Ви можете визначити інформаційну систему, чим вона характеризується?
21. Які основні характеристики інформаційної системи?
22. В чому полягає місія та основні задачі інформаційної системи.?
23. Які компоненти ІС і як вони приймають участь в обробці інформації?
24. Які основні функції ІС?
25. Назвіть етапи розвитку інформаційних систем.
26. В чому полягає системний підхід?
27. Назвіть етапи розвитку інформаційних технологій?
28. Які типи ІС Ви знаєте?
29. На які групи можна поділити задачі обробки інформації?
30. Що таке програмне забезпечення?
31. За якими ознаками можна класифікувати програмне забезпечення?
32. Що забезпечують прикладні програми?
33. Які функції виконують системні програми?
34. Для чого використовуються сервісні програми?
35. Що таке операційна система?
36. Які сучасні операційні системи Ви знаєте?
37. Що входить до функцій операційної системи?
38. Надайте визначення інформації та даних.
39. Як пов’язані інформація і дані?
40. Поняття структури даних. Як вона характеризується?
41. Класифікація структур даних у програмах користувача.
42. Визначте рівні подання структур даних.
43. Назвіть елементи логічної структури економічної інформації.
44. Надайте визначення інформаційного масиву даних та інформаційного потоку.
45. Окресліть взаємозв'язок між елементами логічної структури економічної інформації.
46. Визначте одиниці фізичної структури даних.
47. Наведіть приклади лінійних та нелінійних структур даних.
48. Чим відрізняється стек від черги?
49. Назвіть складені типи даних.
50. Поняття інформаційної моделі.
51. Назвіть етапи вирішення прикладних задач з використання ПК.
52. Що ми розуміємо під поняттям “алгоритм”?
53. Які є способи опису алгоритмів?
54. Поняття схеми алгоритму та правила її побудови.
55. Які властивості повинен мати алгоритм?
56. Які є три головні алгоритмічні конструкції?
57. З чого складаються прості (лінійні) алгоритми?
58. Який процес називається розгалуженим?
59. Який процес називається циклічним?
60. Види циклічних обчислювальних процесів, їх відмінність.
61. Дати визначення етапів формалізації алгоритмів.
62. Назвіть та охарактеризуйте найпоширеніші методи розробки алгоритмів.
63. В чому полягає аналіз алгоритмів.
64. Цілі структурного програмування.
65. В чому полягає тестування алгоритму.
66. Що розуміють під економічною інформацією і які особливості її оброблення?
67. Чи потрібен змістовний опис для задач оброблення . економічної інформації?

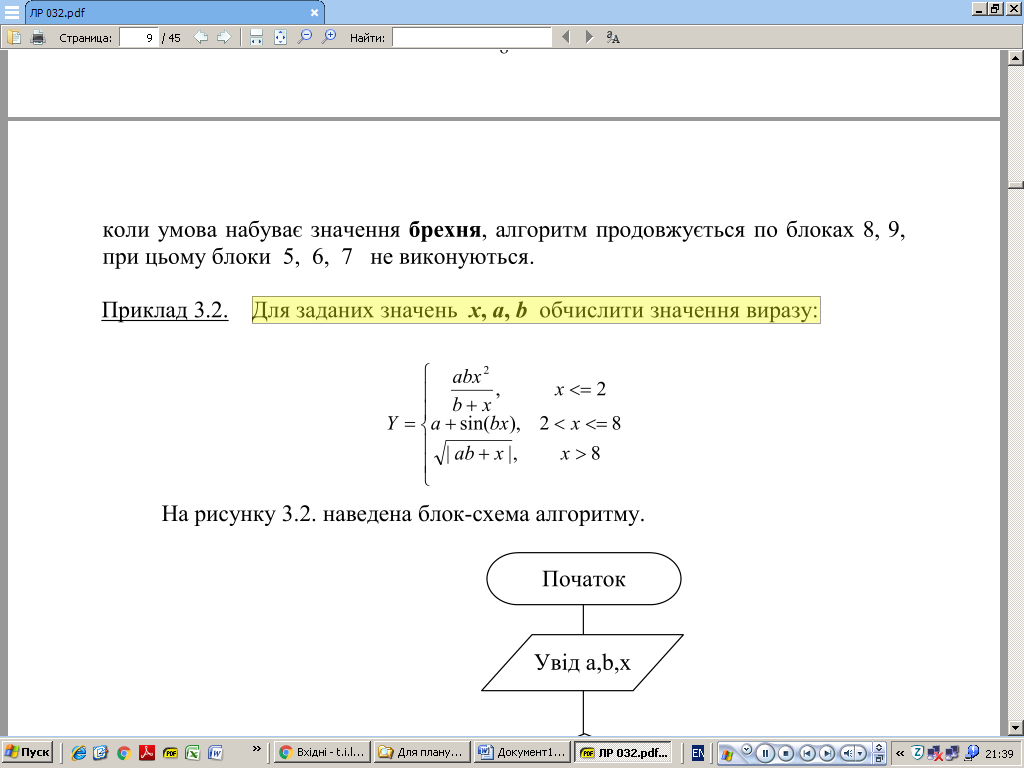
**Завдання**

Намалювати блок – схему рішення задач.

1. Знайти значення дійсних коренів квадратного рівняння

***ax 2 + bx + c = 0 , a ≠ 0***

1. Для заданих значень х, a, b обчислити значення виразу:



1. Задано натуральне число N. Визначити кількість цифр у ньому.

**Повторення**

***Алгоритм*** — це кінцева послідовність однозначних розпоряджень, виконання яких дозволяє за допомогою кінцевого числа кроків отримати рішення завдання, однозначно визначається вихідними даними.. Програму, призначену для виконання комп‘ютером можна розглядати як різновид алгоритму.

***Програма*** — алгоритм, записаний мовою програмування.

***Проект*** — сукупність файлів, які забезпечують запуск і роботу програмного застосунка.

Властивості алгоритму: **Дискретність, Визначеність / однозначність, Виконуваність і скінченність**, **Масовість, Результативність, Формальність / зрозумілість.**

Під **виконавцем** будемо розуміти деяку абстракцію, яка характеризується :

* набором дій (операцій), які виконавець здатний виконати;
* набором об’єктів (даних), над якими виконуються дії;
* набором інструкцій (команд), які виконавець розуміє і реалізує;
* мовою спілкування з виконавцем (алгоритмічною мовою).

**Правила побудови алгоритмів**

* 1. При побудові алгоритму перш за все необхідно задати безліч об'єктів, з якими буде працювати алгоритм. Формалізоване (закодування) представлення цих об'єктів має назву *даних*. Алгоритм приступає до роботи з певним набором початкових даних, які називаються вхідними, і в результаті роботи видає дані, що називаються вихідними. Таким чином, алгоритм перетворює вхідні дані у вихідні.
  2. ***Д***ля роботи алгоритму займають пам'ять комп'ютера. У пам'яті розміщуються вхідні дані, з якими алгоритм починає працювати, а також проміжні дані і вихідні дані. Пам'ять є дискретної, тобто що складається з окремих осередків. Пойменована комірка пам'яті має назву ***змінної***.
  3. ***Д***искретність. Алгоритм будується з окремих кроків (дій, операцій, команд). Безліч кроків, з яких складено алгоритм, звичайно.
  4. ***Д***етермінованість. Після кожного кроку необхідно вказувати, який крок виконується таким, або давати команду зупинки.
  5. ***З***біжність (результативність). Алгоритм повинен завершувати роботу після кінцевого числа кроків . При цьому необхідно зазначити, що вважати результатом роботи алгоритму.

**Способи записів алгоритмів**

* *Словесно-формульний опис*
* *Графічний опис*
* *Алгоритмічною мовою або мовою програмування*
* *Псевдокод*

**Транслятор** аналізує транслюються програму. генерує вихідну програму, розподіляє пам'ять для об'єктної програми. ***Компілятор*** читає всю програму цілком, робить її переклад і створює закінчений варіант програми на машинній мові, який потім і виконується. ***Інтерпретатор*** перекладає і виконує програму *рядок за рядком.*

***Послідовність дій для формування алгоритму включає в себе***:

* аналіз вихідних даних;
* виявлення найбільш значущих аспектів;
* формалізацію ключових моментів;
* подання даних символами;
* формування цілісної послідовності операцій.

***Методи розробки алгоритму*** (програми): *зверху-вниз* – складна проблема розкладається поступово на простіші; *знизу-вверх* – виділяються простіші елементи, які  об’єднуються поступово у складніші; *розширення ядра* – спочатку виділяється основна (головна) частина алгоритму (скелет алгоритму), яка поступово доповнюється рештою елементів.

В алгоритмі застосовуються наступний типи послідовності операцій:

* лінійні;
* циклічні;
* розгалуження;
* змішані.

**Основні етапи розв’язування задач за допомогою комп’ютера**

1. постановка задачі;
2. побудова моделі;
3. вибір або розробка методу розв’язування;
4. розробка алгоритму розв’язання задачі;
5. проектування і реалізація програми;
6. налагодження та тестування програми;
7. використання програми.

**Принципи побудови алгоритму**

Щоб побудувати алгоритм, необхідно дотримуватись певних умов:

* вхідні та вихідні дані задати у вигляді послідовності слів;
* процес розв’язання задачі це є процес перетворення вхідних даних у вихідні.

Процес перетворення складається із сукупності елементарних припустимих операцій формального характеру.

***Припустима елементарна операція*** – це проста, чисто механічна дія, результат якої не залежить від виконавця (машини чи людини);

● послідовність припустимих операцій не залежить від конкретних вхідних даних;

● порядок виконання припустимих операцій визначається однозначно;

● сукупність припустимих операцій визначається класом задач та типом даних.

***Інформація*** – відображення реального світу, яке має властивості: здобування, передавання, збереження (інформації). Носієм інформації є ***повідомлення***, яке формально складається з символів, позначок та іншого.

***Дані*** – інформація у вигляді повідомлення спеціальним чином представлена технічними пристроями, наприклад, ЕОМ або людиною

**Структура даних (СД)** - загальна властивість інформаційного об'єкта, з яким взаємодіє та або інша програма. Ця загальна властивість характеризується:

* множиною допустимих значень цієї структури;
* набором допустимих операцій;
* характером організованості.

Найпростіші структури даних називаються також ***типами даних***. Класифікація СД.

**Логічний рівень (ЛСД)**– подання структури даного на тій чи іншій мові програмування.

**Фізичний рівень (ФСД)** — відображення у пам'ять комп'ютера інформаційного об'єкту відповідно до логічного описування.

**Одиниці фізичної структури даних**

**Символ** - це найпростіший елемент даних, сигнал інформації (літера, цифра, знак), який окремо не має змісту.

**Поле** - множина символів, яка створює мінімальний семантичний елемент масиву.

**Агрегат даних** - це пойменована сукупність двох і більше елементів нижчого рівня, яка має окремий зміст . До агрегату даних можуть належати як елементи, так і інші агрегати даних.

**Запис** - пойменована сукупність полів, об'єднаних за змістовним принципом.

Агрегати даних і записи реалізуються на практиці шляхом організації списків, черг, стеків, таблиць.

**Файл** - іменована сукупність записів про об'єкти одного типу. Як правило, записи, що входять у файл, мають однакову структуру.

**База даних** - це іменована сукупність взаємозв'язаних даних, що відображає стан об'єктів і їх відношення в даній предметній області.

До лінійних структур належать послідовні структури, в яких елементи розміщуються в тому порядку, який необхідний під час їх обробки, наприклад, список, елементами якого є записи (рядкова структура).

До нелінійних структур даних належать складні списки, дерева, мережі, табличні та гібридні структури:

* складні списки містять вкладені елементи (підсписки) меншого обсягу;
* деревоподібні структури - елементи розміщуються на різних рівнях і сполучаються за допомогою адрес;
* сіткові структури являють собою розширення дерева за рахунок нових адрес зв'язку;
* табличні структури даних призначені для зберігання інформації про ключові ознаки даної інформаційної сукупності;
* гібридні структури даних містять фрагменти різних структур.

Зі структурою пов'язана **модель даних** - фіксована система понять і правил для представлення структури даних.

**Теорема про структуризацію:** будь-яка проста програма може бути перетворена у функціонально їй еквівалентну програму, побудовану на основі 3 наступних структур:

*Базові структури алгоритмів*

1. Проходження / слідування.
2. Якщо-то-інакше (розгалуження).
3. Цикл з передумовою.

**Тестування алгоритму** – це перевірка правильності або неправильності роботи алгоритму на спеціально заданих тестах або тестових прикладах – завданнях з відомими вхідними даними і результатами (іноді достатні їх наближення). Тестовий набір повинен бути мінімальним і повним, тобто що забезпечує перевірку кожного окремого типу наборів вхідних даних, особливо виняткових випадків.