

Реферат

Об'єкт дослідження: Microsoft Office, Libreoffice, C++, Java для Android програм, Android Studio

Мета роботи: ознайомлення з роботою Microsoft Office, Libreoffice, з базовими задачами з програмування, а також з створенням простих програм на смартфоні на базі Android.

Одержані висновки та їх новизна: Libreoffice, якщо зберігати у форматі docx або xlsx, створює документи та таблиці, які може читати Microsoft Office, але у документах часто ламається розмітка, формули, і також не сумісні макроси, які причому у Libreoffice не працює так ефективно, як у Microsoft Office, також на комп'ютерах з малою кількістю ОЗУ можна налаштувати Android Studio так, щоб можна було ефективно та швидко створювати програму і вистачало ресурсів комп'ютера.

Результати досліджень можуть бути застосовані при подальшому створенні програм для смартфонів

Перелік ключових слів: WORD, EXEL, C++, JAVA, ANDROID STUDIO, LIBREOFFICE, АЛГОРИТМИ.

Зміст

Лабораторні роботи з використанням Word.....	3
Лабораторна робота №1,2.....	3
Лабораторна робота №3.....	6
Лабораторна робота №4.....	9
Лабораторна робота №5.....	15
Лабораторні роботи з використанням Exel.....	19
Лабораторна робота 7.....	19
Лабораторная работа 8.....	22
Лабораторная работа 9.....	26
Задачі з програмування.....	28
Cashbox.....	29
Character.....	33
Geometric progression nth.....	36
If sequence contains element.....	37
Intersection between ray and plane 3D.....	40
Subsequence.....	42
Калькулятор на Android.....	44
Висновок.....	48
Використані джерела.....	51

Лабораторні роботи з використанням Word

Лабораторні роботи починаючи з 2-ої виконувалися у Libreoffice 5 на операційній системі Ubuntu, а також у Microsoft Office 2007 у емуляторі Wine, використовувався для перевірки на сумісність між цими видами програм, а також якщо у Libreoffice були відсутні деякі функції

Лабораторна робота №1,2

Постановка задачі:

Було виконано такі дії:

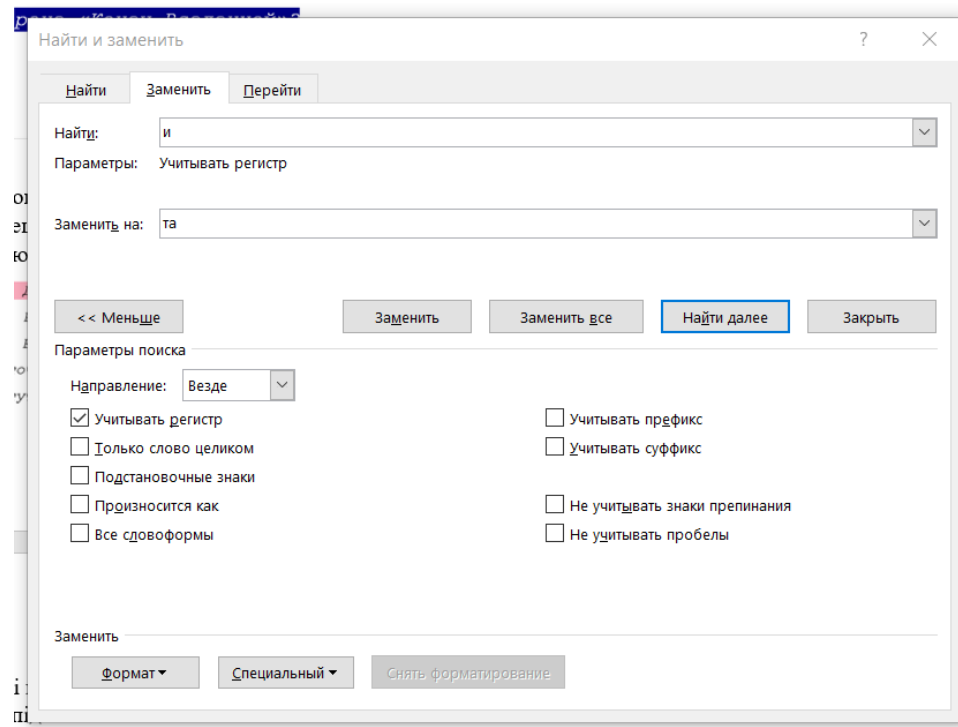
1. Ознайомитись з інтерфейсом графічного процесору Word 2003 і/або 2007.
2. Створити новий документ та набрати текст з наданого файлу, або, за згодою викладача, з іншого обраного джерела.
3. Там, де необхідно, але не менше одного разу застосувати символи нерозривного пробілу і/або дефісу, м'якого переносу.
4. Включити автоматичну розстановку переносів в документі.
5. Зберегти відформатований текст у папці своєї групи в мережі комп'ютерного класу під іменем Лаб1_Група_Прізвище.doc
6. Відкрити в Word документ, створений при виконанні першої лабораторної роботи.
7. Установити поля по 2 см. з усіх боків. Задати книжкову орієнтацію сторінки. Задати режим перегляду – «Розмітка сторінки».
8. Включи перегляд знаків, що не друкуються.
9. Оформити назву тексту прописними буквами.
10. Установити розмір тексту 14пт для всього документу.
11. Установити міжрядковий інтервал 1,5. Абзацний відступ встановити рівним 0,5 см.

12. Передостанній абзац тексту обрамити в рамку сірого кольору шириною 1пт.
13. Залити фон останнього абзацу темно-синім кольором, а текст абзацу зробити білого кольору.
14. Вставте нумерацію всіх сторінок, крім першої.
15. Знайдіть у тексті слово «та». Для цього Виправлення/Знайти, у діалоговому вікні Знайти й замінити клацніть кнопку Більше й заберіть прапорець. Ураховувати регістр. Здійсніть автоматичну заміну слова «та» на слово «і».
16. Перевірте текст на наявність граматичних помилок.
17. Зберегти відформатований текст у папці своєї групи в мережі комп'ютерного класу під іменем Лаб2_Група_Прізвище.doc

Хід роботи:

1. Набрав текст з наданого файлу, обрав розділ відповідно до свого варіанту за списком групи и 2 наступні: 9, 10 і 11 глави.
2. Використав набір клавіш Ctrl + Shift + Space для нерозривного пробілу.
3. У макеті сторінки – параметри – розстановка переносів – автоматично.
4. Цей текст зберігв під назвою Лаб1_ПА-17-1_Панасенко.doc
5. Далі продовжив роботу з цим документом, не змінюючи Лаб1_ПА-17-1_Панасенко.doc
6. Встановив поля 2 см у параметрах сторінки – поля.
7. Також, параметри сторінки – орієнтація обрав тип – книжкова.
8. Вид сторінки – розмітка сторінки.
9. Включив перегляд знаків, що не друкуються.
10. Оформив назву тексту курсивом Ctrl + I.
11. Обрав розмір шрифту 14 пт. для всього документу, головна – шрифт – розмір.

12. Головна – абзац – міжрядковий інтервал - 1,5.
13. Абзацний відступ встановив за допомогою повзунка, який знаходиться під панеллю інструментів, обрав 0,5 см.
14. Передостанній абзац: головна – абзац – границі й заливка. Обираю ширину границі - 1пт і колір – сірий.
15. Останній абзац: головна – шрифт – колір виділення тексту – темно-синій, головна – шрифт – колір шрифту – білий.
16. Клацнувши двічі на верхній або нижній колонтитул, заходжу до роботи з ним. Обираю нумерацію сторінки – нижній колонтитул, зліва. У параметрах обираю – особливий верхній колонтитул.
17. Головна – редакція – заміна. У полі “знайти” – ‘и’, у полі “замінити на що” – ‘та’ й ставлю прапорець на врахування регістру. На першій сторінці роблю рецензія – замітка – пишу, що на що замінюю.
18. Далі перевіряю й виправляю всі граматичні помилки, які були відмічені червоною хвилястою лінією під словом.
19. Зберегв документ під назвою Лаб2_ПА-17-1_ Панасенко.doc



Лабораторна робота №3

Постановка задачі:

Завдання 1. Створити наведений нижче текстовий документ. Рамку не набирати.

Завдання 2. Продовжите документ наступним текстом (рамку не набирати)

Завдання 3. Перетворити текст до виду списку по наведеному нижче зразкові, включаючи рамку.

Скопіювати створений список і перетворити його до наступного вигляду:

1. тополя – 100 г.;
2. липа – 100 г.;
3. береза – 90 г.;
4. клен – 20–30 г.

Створити багаторівневий список наступного вигляду:

1. Місто й міське середовище
 - a. Історія й перспективи урбанізації
 - b. Міське господарство
2. Водне середовище міста
 - a. Водні об'єкти міст
 - i. Джерела в міській рисі
 - b. Оцінка стану водних об'єктів
 - i. Показники якості води
 - ii. Оцінка стану якості води
 - c. Оцінка впливу на водні об'єкти
3. Розвиток міст в ХХІ сторіччі

4. Додаток

Завдання 4. Задати верхній колонтитул.

Завдання 5. Створіть новий розділ (меню Вставка\Розрив Новий розділ 3 наступної сторінки.).

Завдання 6. Встановіть всі поля по 1,5 см. Шрифт - *Times New Roman* 12 пт.

Завдання 7. Створити текстовий документ відповідно до наведеного нижче зразка (Рамку не вводити). Шрифт - *Times New Roman* 12 пт.

Завдання 8. Установити режим розміщення переносів. Для цього: меню Сервіс\Мова\Розміщення переносів. Установити прапорець Автоматичне розміщення переносів.

Завдання 9. Створити колонки в набраному тексті згідно з наведеним зразком.

Завдання 10. Зберегти створений файл у папці своєї групи під іменем Лаб3_Група_Прізвище.doc

Хід роботи:

1. Набрав текст, який представлений нижче
2. У меню Формат вибрав відповідний маркер
3. За допомогою меню Формат, закладка Маркірований, кнопка Змінити установив параметри форматування списку.
4. Вибрав відповідний відступ, як показано у наведеному текстовому документі

Практика

Панасенко

ПА-17-1

Місто представляється як комплексна система, до складу якої входять:

- ♥ Урбоекосистема, тобто видозмінена під впливом людини природна екосистема міської території;
- ♥ Соціальна підсистема;
- ♥ Техносфера міста

5. Набрав текст
6. Претворив текст до виду списку
7. Скопіював створений список та замінив маркіровку

8. Створив багаторівневий список за допомогою багаторівневого маркірування
9. У верхньому колонтитулі задав наступні табулятори:
 1. 1 см вирівнювання по Лівому краю
 2. 8,5 см вирівнювання По центру
 3. 15 см вирівнювання по правому краю
10. На першому табуляторі в колонтитулі ввів “Програмування”, на другому – Панасенко, на третьому – ПА-17-1.
11. Перемістивсь на нижній колонтитул та вставив автотекст
12. Створив новий розділ за допомогою Вставка\Розрив Новий розділ
13. Встановить всі поля по 1,5 см. Шрифт - Times New Roman 12 пт.
14. Створив текстовий документ, який представлений на зразку
15. Установив режим розміщення переносів
16. Створив колонки у заданому тексті за допомогою Меню-Формат\Колонки, Тип – три, установив прапорці Роздільник і Колонки однакової ширини, застосував до Виділеного тексту.

Практика

Панасенко

ПА-17-1

Новий розділ 3 СУПЕРПРОДУКТИ

Людство завжди з оптимізмом дивиться в майбутнє. Ми очікуємо відкриттів у медицині, які дозволять нам жити довгим, здоровішим, насиченим життям. Реальний шанс відновити й підтриму-

вати здоров'я дають супер продукти. Концентровані супер продукти, від люцерни до паростків пшениці, є потужними джерелами вітамінів, мінералів, ферментів, клітинних солей, антиоксидантів, клітковини й незамін-

них амінокислот. Вони підвищують енергію й запас життєвих сил. контролюють рівень РН, поліпшують розумову діяльність і очищають клітки від токсинів.

17. Оформив відповідно до зразка
18. Зберегв створений файл у папці своєї групи під іменем Лаб3_ПА-17-1_Панасенко.doc

Лабораторна робота №4

Постановка задачі:

Завдання 1. Створити просту таблицю по наведеному нижче зразкові.

Завдання 2. Перетворити першу строку таблиці на заголовок таблиці. Для цього виділити першу строку і обрати пункт меню *Таблиця – Заголовок*. (**Word 2007:** Встановіть курсор у будь-яке місце таблиці, на вкладці *Макет*, що з'явилася в меню зверху, оберіть *Повторять строки заголовков*).

Завдання 3. Доповнити документ таблицею відповідно до наведеного нижче зразка. Для цього:

1) Вставити назву таблиці з автоматичною нумерацією Для цього виділити таблицю і обрати пункт меню *Вставка – Посилання – Назва у меню* (**Word 2007:** *Ссылки – Вставить название*).

2) Вставити таблицю, що складається з 4 стовпців і 5 рядків. *Таблиця – Вставити – Таблиця; Автопідбір ширини стовпців – По ширині вікна* (**Word 2007:** *Вставка – таблица – Вставить таблицу*).

3) У першому стовпці об'єднати комірки першого й другого рядка. Для цього:

i. Виділити поєднувані комірки

ii. Меню *Таблиця – Об'єднати комірки* **Word 2007:** правий клік на виділених комірках, *Объединить ячейки*

4) У першому рядку об'єднати комірки другого, третього й четвертого стовпців.

5) Уведіть текст у комірки

6) Вирівняйте текст у комірках відповідно до зразка. Для вирівнювання вивчіть, і використовуйте кнопку вирівнювання на панелі інструментів *Таблиці й границі (Word 2007: вкладка Макет – Выравнивание)*

7) Змініть розмір першого й другого стовпця відповідно до зразка. Для цього встановіть курсор миші на праву границю першого (а потім другого) стовпця, і перемістіть границю вліво.

8) Установіть розміри 3 і 4 стовпців по ширині тексту. Для цього встановіть курсор на праву границю відповідного стовпця й двічі клацніть лівою кнопкою миші.

9) Розташуйте всю таблицю по центру – Меню *Таблиця\Властивості таблиці*; закладка *Таблиці*; кнопка *По центру*.

10) За допомогою меню *Формат\Границі й заливання (Word 2007: правий клік, Границы и заливка)* оформіть границі комірок .

11) Зберегти результат у папці своєї групи під іменем *Прізвище_Таблиця1*, де прізвище – Ваше прізвище.

Завдання 4. Доповнити документ таблицею відповідно до наведеного нижче зразка.

Для цього:

1) Вставте назву з автоматичною нумерацією

2) Вставте таблицю, що складається з 2 стовпців і 4 рядків

3) За допомогою кнопки *Ластик* на панелі інструментів *Таблиці й границі* стерти:

i. у першому стовпці – границю між першим і другим рядком.

ii. у третьому рядку – границю між стовпцями.

4) За допомогою кнопки *Намалювати таблицю* панелі інструментів *Таблиці й границі* розділіть другий рядок другого стовпця на дві частини.

5) Відіжміть кнопку *Намалювати таблицю*.

6) Вирівняйте тільки що створені комірки по ширині. Для цього:

iii. Виділіть комірки

iv. Меню *Таблиця – Автопідбір – Вирівняти ширину стовпців* (Word 2007: вкладка *Макет*)

7) Аналогічно розділіть 4 рядок другого стовпця

8) Заповніть таблицю й форматуйте відповідно до зразка.

9) Вставити в стовпець *Вартість, грн.* формули для розрахунку вартості. В останню строку (*Разом*) вставити формулу для розрахунку сумарної вартості. Для цього:

а) Обрати пункт меню *Таблиця – Формула (Word 2007: Макет – Данные – Формула)*.

б) Для розрахунку вартості в діалоговому вікні, що з'явилося, ввести в поле *Формула: PRODUCT(LEFT)* (що українською означає *ДОБУТОК(ЗЛІВА)*). Повторити цю дію для кожної строки таблиці.

в) Для розрахунку строки *Разом* аналогічним чином вставити у відповідну комірку таблиці формулу *SUM(ABOVE)* (що українською означає *СУМА(ЗВЕРХУ)*).

Результат повинен бути наступним:

За даними таблиці створити діаграму вартості матеріалів. Для цього:

а) Обрати пункт меню *Вставка – Рисунок – Діаграма (Word 2007: Вставка – Диаграмма)*. Обрати тип діаграми *Кругова*, що відповідає нижче наведеному прикладу.

б) В діалогове вікно з таблицею даних, що з'явилося, скопіювати стовпці *Найменування* і *Вартість, грн.* з таблиці.

в) Якщо діаграма після попереднього кроку не відповідає прикладу і складається з одного сегменту без підпису, в меню *Дані* оберіть пункт *Ряди утворюють стовпці*.

г) Налаштуйте параметри (легенду, підписи) та кольори діаграми, клацаючи правою кнопкою миші на пустому просторі діаграми та/або на окремих елементах. Результат повинен відповідати нижче наведеному прикладу.

г) Закрийте налаштування діаграми, клацнувши мишею на будь-якому місці за її межами. В подальшому для переходу у налаштування діаграми ви можете двічі клацнути мишею на ній.

Хід роботи:

Завдання 1. (Таблиця №1)

1. Створив таблицю за допомогою *Вставка – Таблица – Вставить таблицу*, розміром 5x4, зробив автопідбір ширини стрібців – по ширині вікна.
2. Установив шрифт Arial 10 пт. на всій таблиці.
3. Заповнив таблицю
4. Оформив рамки таблиці за допомогою меню *Границі й заливання* та задав товщину 1,5

Таблиця 1

Категорія річки	Загальна площа водозбору, км ²	Витрата води, м ³ /з	Швидкість річки, м/з	Коливання річки, м
Мала	до 2000	до 5	до 0,5	до 1
Середня	2000-50 000	5-100	0,2-1	1-2
Більша	понад 50 000	понад 100	понад 1	понад 2

5. Зробити заливання першого рядка за допомогою меню Границі й заливання, вкладка Заливання
6. Перетворив першу строку таблиці на заголовок таблиці.
7. Вставив назву таблиці з автоматичною нумерацією. Для цього виділив таблицю і обрав пункт меню Ссылки – Вставить название.
8. Вставив таблицю розміром 4x5
9. У першому стовпці об'єднав комірки першого й другого рядка, а у першому рядку об'єднав комірки другого, третього й четвертого стовпців.
10. Увів текст
11. Змінив розмір першого й другого стовпця відповідно до зразка.
12. Розташував всю таблицю по центру – Меню Таблица\Властивості таблиці; закладка Таблиці; кнопка По центру.
13. Оформив границі комірок

Таблиця 2

Вид поля	Інтенсивність		
	Одиниці виміру	Природне	Техногенне
Вібраційне (динамічне)	Вт/м ²	відсутнє	10-5 ÷ 10-4
Теплове	Вт/м ²	10-2 ÷ 10-1	більш 1
Електричне	А/м ²	менш 10-7	до 10

14. Зберег результат у папці своєї групи під іменем Лаб4_ПА-17-1_Панасенко.docx

Завдання 2. (Таблиця №2)

1. Вставив назву з автоматичною нумерацією
2. Вставив таблицю, розміром 2x4.

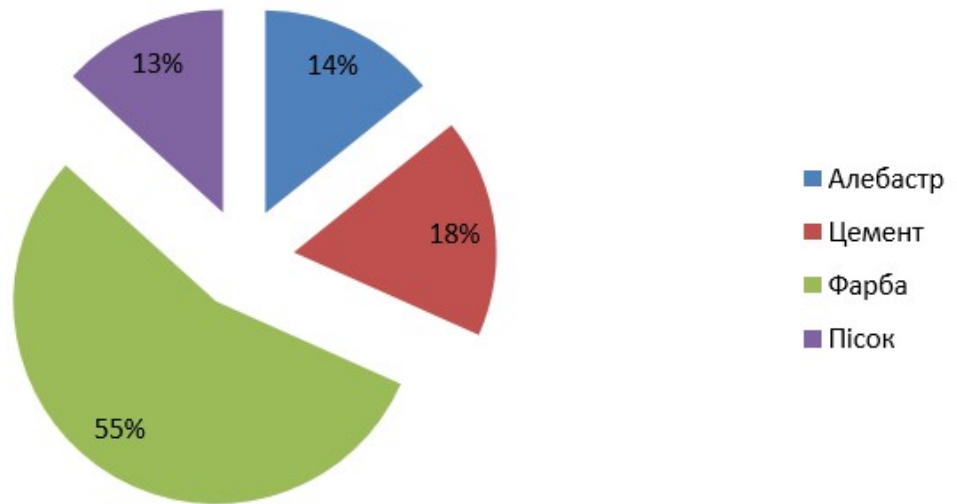
3. За допомогою кнопки Ластик на панелі інструментів Таблиці й границі стерв:
 1. у першому стовпці – границю між першим і другим рядком.
 2. у третьому рядку – границю між стовпцями.
4. За допомогою кнопки Намалювати таблицю панелі інструментів Таблиці й границі розділили другий рядок другого стовпця на дві частини, вирівняв їх по ширині.
5. Аналогічно розділив 4 рядок другого стовбця
6. Заповнив таблицю й відформатовав відповідно до зразка
7. Для розрахунку вартості в діалоговому вікні, що з'явилося, ввів в поле Формулу: PRODUCT(LEFT)
8. Повторив цю дію для кожної строки таблиці
9. Для розрахунку строки Разом аналогічним чином вставив у відповідну комірку таблиці формулу SUM(ABOVE)

Таблиця 3

№	Найменування	Кількість	Ціна одиниці, грн.	Вартість, грн.
1	Алебастр	5	10,26	51,3
2	Цемент	5	12,63	63,15
3	Фарба	8	24,85	198,8
4	Пісок	12	4,00	48
Разом				361,25

10. Вставка – Диаграмма – Кругова діаграма
11. В діалогове вікно з таблицею даних, що з'явилося, скопіював стовпці Найменування і Вартість, грн. з таблиці.
12. Налаштував параметри (легенду, підписи) та кольори діаграми.

Продажи



13.3 беріг результат у папці своєї групи під іменем Лаб4_ПА-17-1_Панасенко.docx

Лабораторна робота №5

Постановка задачі:

1. Створіть текстовий документ, що містить наведені нижче формули. Для цього:

- 1) Помістіть курсор у те місце, де повинна бути вставлена формула
- 2) Меню *Вставка – Об'єкт* у закладка *Створення* Тип об'єкта *Microsoft Equation 3.0*
- 3) Для вставки символу у формулу потрібно клацнути по кнопці на панелі інструментів *Формула* й вибрати шаблон з палітри, що з'явився. Латинські й російські букви, цифри, арифметичні знаки вводяться із клавіатури.

2. Створіть наведений нижче текст, що містить формули і їх автоматичну нумерацію:

Для вставки автоматичної нумерації формул:

- 1) Установіть курсор у місце вставки першого номера формули.
- 2) Меню *Вставка – Посилання – Назва* (**Word 2007:** виділити рисунок/формулу/інший об'єкт, клікнути правою кнопкою миші, обрати *Вставити название*)
- 3) У діалоговому вікні, що з'явилося, Назва нажати кнопку *Створити*.
- 4) У вікні, що з'явилася, Нова назва ввести ліву круглу дужку (і нажати ОК.
- 5) У вікні *Назва* закрити праву круглу дужку й нажати ОК.
- 6) Для вставки наступних номерів: меню *Вставка – Посилання – Назва*; у списку, що випадає, *Підпис* виберіть символ лівої круглої дужки (, а у вікні *Назву* закрийте праву круглу дужку й натисніть ОК.

Розмістіть формули по центру, а нумерацію – по правому краю. Для цього скористайтесь табуляторами.

3. Створіть документ, що міститиме зміст. В документ скопіюйте всі файли, отримані в результаті виконання даної і попередніх лабораторних робіт. З метою розділення лабораторних між собою перед кожною лабораторною роботою вставте заголовок *Лабораторна робота №__, ПІБ*. Кожна лабораторна повинна починатися з нової сторінки (забезпечити шляхом вставки розривів сторінок: *Вставка – Розрив сторінки*).

Word 2003:

4. Використовуючи панель *Структура*, встановіть заголовкам *Лабораторна робота №__, ПІБ* рівень 1. Також можна скористатися панеллю *Форматування*, на якій обрати стиль *Заголовок 1* (на відміну від попереднього способу, даний спосіб змінює ще і формат тексту).

5. Для створення змісту оберіть в меню пункт *Вставка – Посилання – Зміст* і вказівники. В діалоговому вікні, що з'явилося, оберіть вкладку *Зміст*, за необхідності встановіть інші параметри і натисніть ОК.

Word 2007:

Щоб виділити текст як заголовок, на вкладці *Ссылка* оберіть пункт *Оглавление – Добавить текст – Уровень 1*. Щоб створити зміст, використовуйте кнопку *Оглавление*, що знаходиться там же.

Хід роботи:

1. Вставляю формулу за допомогою Меню Вставка – Об'єкт у закладка Створення Тип об'єкта Microsoft Equation 3.0
2. Ввожу формулу за допомогою вставки символів на шаблоні з палітри, що з'явився.

$$P = \frac{\sum_{i=1}^n T_i - \prod_{j=1}^m (Z - L_j)^2}{\sqrt[3]{\omega \cdot \Psi}}$$

$$g(p) = \int_0^{\infty} e^{-pt} f(t) dt$$

$$q = \frac{M}{2(1-n)k_1\sqrt{\pi K_0}x^3} e^{-\frac{y^2}{4K_0x}}$$

3. Ввожу наведений текст
4. Для вставки автоматичної нумерації формул виділяю рисунок/формулу/інший об'єкт, клікаю правою кнопкою миші, та оберая Вставить название
5. У діалоговому вікні, що з'явилося, Назва нажимаю кнопку Створити.
6. У вікні, що з'явилася, Нова назва ввожу ліву круглу дужку (і нажимаю ОК.
7. У вікні Назва закриваю праву круглу дужку й нажимаю ОК.

$$L(1 - \mu)K_1 \sqrt{\pi \Lambda_0} x \sim$$

У загальному вигляді завдання прогнозу забруднення повітря математично може бути описана рівнянням:

$$\frac{\partial \Omega}{\partial t} + \sum_{i=1}^3 U_i \frac{\partial \Omega}{\partial x_i} = \sum_{i=1}^3 U_i \frac{\partial}{\partial x_i} k_i \frac{\partial \Omega}{\partial x_i} - a \Omega$$

(1)

При розгляді зберігаючої консервативної домішки:

$$U \frac{\partial \Omega}{\partial x} = \frac{\partial}{\partial z} k_z \frac{\partial \Omega}{\partial z} + \frac{\partial}{\partial y} k_y \frac{\partial \Omega}{\partial y}$$

(2)

8. Зберігаю результат у папці Лаб5_ПА-17-1_Панасенко.

Лабораторні роботи з використанням Excel

Лабораторна робота 7

Постановка задачі:

Завдання 1. Обчислити значення функцій і побудувати графіки отриманих функцій.

Завдання 2. Створити таблицю з 3 стовпців (№ п/п, x , y) і внести до неї значення згідно з варіантом. Апроксимувати введені значення математичною функцією.

Виділивши діапазон комірок x і y , створити діаграму-графік функції $y(x)$.

Вивести в область побудови діаграми шукане вираження для функції. Для цього необхідно встановити покажчик миші на лінію графіка й клацнути правою кнопкою. У меню, що з'явився, клацнути по рядкові *Додати лінію тренда* У діалоговому вікні *Лінія тренда* вибрати найбільш прийнятний варіант із числа пропонованих і клацнути по ньому мишею. Для поліноміальної лінії треба ще встановити передбачуваний показник ступеня. Потім клацнути мишкою по кнопці *Параметри* й клацанням миші встановити прапорці: *показувати рівняння на діаграмі* й *помістити на діаграму величину вірогідності апроксимації (R^2)*. Після цього клацнути по кнопці ОК.

Перетягти рівняння, що з'явилося, на вільне місце діаграми. Ліва частина даного рівняння є вираженням апроксимуючої функції.

Через рядок нижче таблиці підбити підсумок роботи, записавши:

Вираження шуканої функції має вигляд: $y=$. Далі записується вираження для отриманої залежності (права частина рівняння). Для зображення вираження функції у вигляді об'єкта (малюнка) необхідно використовувати майстер функцій

Microsoft Equation 3.0. Вставка об'єкта Microsoft Equation 3.0 у Microsoft Excel аналогічна до Microsoft Word.

Хід роботи:

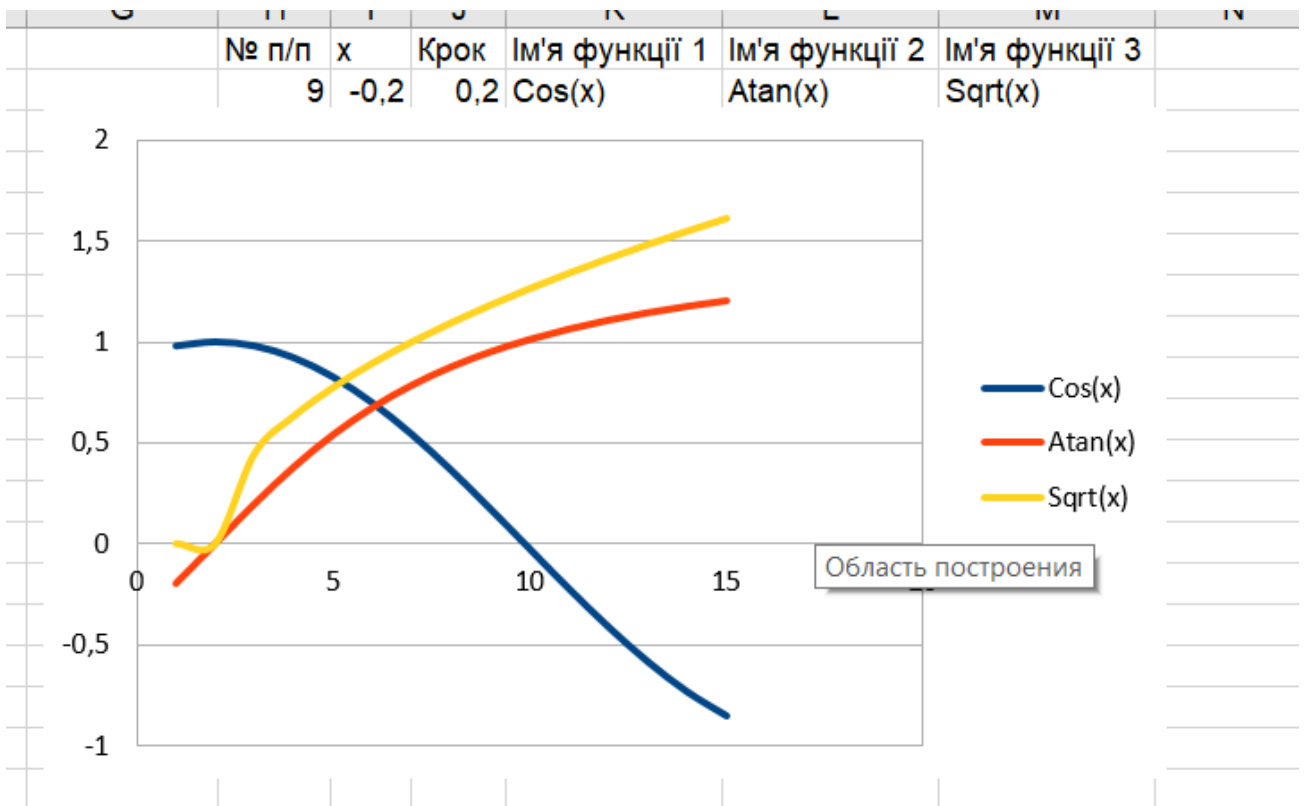
Завдання 1.

1. Формуємо таблицю за заданою формою
2. У заголовках «Ім'я функції» введемо Cos(x), Atan(x), Sqrt(x).
3. У стовпці «№ п/п» для одержання порядкових номерів використовуємо формулу =СТРОКА() – k
4. У стовпець «x» введемо формулу =a+(i-1)*h
5. У першому стовпці таблиці вводимо: в 16 рядку слово Кількість, в 17 рядку слово Сума, в 18 рядку текст Порівн. значення, в 19 рядку слово Максимум і в 20 рядку слово Мінімум
6. Використовуємо відповідні формули
7. У стовпець справа від таблиці вводимо формулу для візуального позначення максимального й мінімального зі знайдених значень однієї з обчислених функцій
8. Застосуємо до обчислених значень функції умовне форматування. Виділяємо червоним кольором фону ті комірки, в яких замість значення написано «Не існ.».

	A	B	C	D	E	F
1	№ п/п	x	Cos(x)	Atan(x)	Sqrt(x)	Atan
2	1	-0,2	0,980066578	-0,19739556	He ich.	min
3	2	0	1	0	0	
4	3	0,2	0,980066578	0,19739556	0,447213595	
5	4	0,4	0,921060994	0,380506377	0,632455532	
6	5	0,6	0,825335615	0,5404195	0,774596669	
7	6	0,8	0,696706709	0,674740942	0,894427191	
8	7	1	0,540302306	0,785398163	1	
9	8	1,2	0,362357754	0,876058051	1,095445115	
0	9	1,4	0,169967143	0,950546841	1,183215957	
1	10	1,6	-0,029199522	1,012197011	1,264911064	
2	11	1,8	-0,227202095	1,063697822	1,341640786	
3	12	2	-0,416146837	1,107148718	1,414213562	
4	13	2,2	-0,588501117	1,144168834	1,483239697	
5	14	2,4	-0,737393716	1,176005207	1,549193338	max
6	15	2,6	-0,856888753	1,203622493	1,61245155	
7	Кількість	15	15	15	14	
8	Сума	18	3,620531637	10,91450996	14,69300406	
9	Порівн. значення	1,1	0,319815742	0,693634819	1,006196347	
0	Максимум	2,6	1	1,176005207	1,549193338	
1	Мінімум	-0,2	-0,856888753	-0,19739556	0	
2						

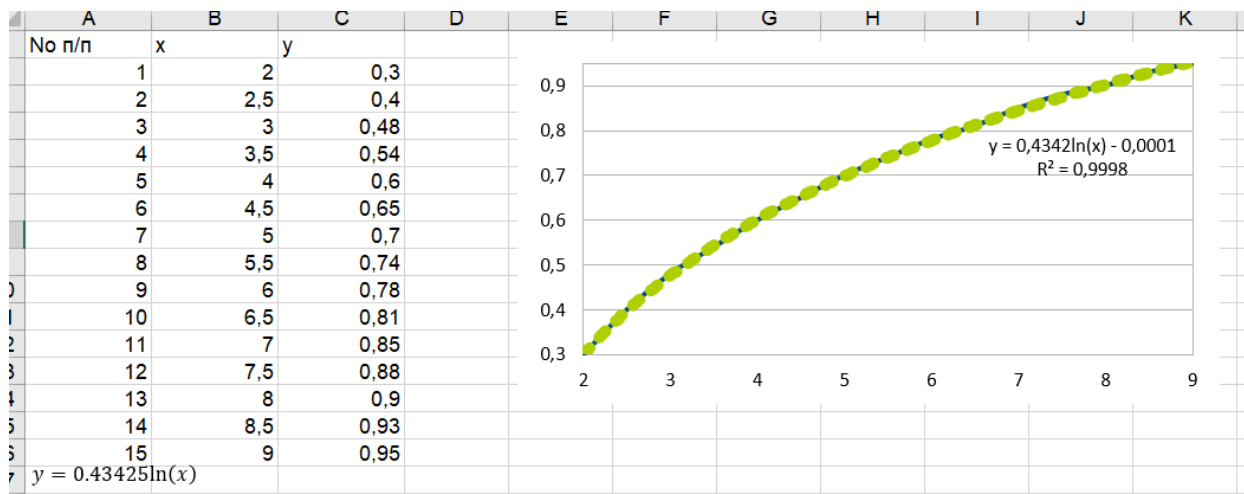
9. Записуємо макроси для увімкнення і вимкнення умовного форматування згідно попереднього пункту в виділених комірках. Назначаємо написаним макросам гарячі клавіші

10. Вставляємо до документу діаграму із графіками всіх трьох функцій.



Завдання 2.

1. Створюємо таблицю з 3 стовпців (№ п/п, x, y) і внесемо до неї значення згідно з варіантом. Апроксимуємо введені значення математичною функцією
2. Створюємо діаграму-графік функції $y(x)$ та вводимо в неї значення.
3. У діалоговому вікні Лінія тренда вибираємо найбільш прийнятний варіант із числа пропонованих і клацаємо по ньому мишею
4. Вибираємо Параметри й клацанням миші встановлюємо прапорці: показувати рівняння на діаграмі й помістити на діаграму величину вірогідності апроксимації (R^2).
5. Зберігаю результат у папці Лаб7_ПА-17-1_Панасенко.



Лабораторная работа 8

Постановка задачі:

Завдання 1. Розв'яжіть систему лінійних алгебраїчних рівнянь (СЛАР) всіма трьома способами (метод Крамера, матричний метод, пошук розв'язку), які описані в теоретичних відомостях нижче.

$$9) \begin{cases} 4x_1 + 4x_2 + 4x_3 + 8x_4 + 12 = 0, \\ -8x_2 - 2x_3 + 6x_4 - 26 = 0, \\ -2x_1 + 2x_2 - 8x_3 + 8x_4 = 0, \\ -8x_2 + 2x_3 - 6x_4 - 22 = 0; \end{cases}$$

Правильність отриманих рішень легко перевірити. У результаті Розв'язання обраної СЛАР трьома різними способами повинні вийти три однакові Розв'язання. Однак, про всякий випадок, приведемо Розв'язання систем 1)-16):

Завдання 2. Розв'яжіть нелінійне алгебраїчне рівняння за допомогою пошуку розв'язку і підбора параметра. Рішення повинно знаходитись в інтервалі від a до b .

9	$\cos(x) = \ln(1+x)$	0	$\pi/2$
---	----------------------	---	---------

Хід роботи:

1. Створюємо таблицю і вводимо значення матриці без вільного члена
2. Створюємо матрицю b і вводимо в неї значення вільного члена системи але з протилежним знаком
3. Створюємо таблицю для метода Крамера
4. Рахуємо детермінант за допомогою формули МОПРЕД
5. Підставляємо замість першого стовбця матриці матрицю b і рахуємо детермінант
6. Аналогічно підставляємо матрицю b в усі стовбці і рахуємо для кожного випадка детермінант за допомогою формули МОПРЕД
7. Отримаємо результат, розділивши детермінанти змінених стовбців та детермінант початкової матриці
8. Записуємо відповіді в окрему строку

9. Створюємо таблицю для матричного метода
10. Записуємо початкову матрицю та її детермінант
11. Робимо транспонування нашої матриці A за допомогою формули МОБР, при цьому введемо матричні формули за допомогою комбінації Ctrl+Shift+Enter
12. Рахуємо відповідь за допомогою формули СУММПРОИЗВ від першої строки матриці A^{-1} і матриці b .
13. Аналогічно робимо з іншими строками матриці A^{-1}
14. Записуємо відповіді в окрему строку
15. Створюємо таблицю для пошуку розв'язку
16. Копіюємо матрицю A
17. Копіюємо відповіді
18. Рахуємо вектор b за допомогою формули СУММПРОИЗВ строки A і строки з відповідями
19. Отримаємо значення матриці b
20. Створюємо таблицю підбір параметру
21. У стовбець x записуємо значення від a до b
22. У стовбець $f(x)$ вводимо формулу $f(x) = \cos(x) - \ln(x+1)$ і рахуємо значення $f(x)$
23. Знаходимо значення x при якому функція $f(x) = \cos(x) - \ln(x+1)$ змінить знак
24. Знаходимо значення $f(x)$ для цього числа

Завдання 1					
	x1	x2	x3	x4	b
1	4	4	4	8	-12
2	0	-8	-2	6	26
3	-2	2	-8	8	0
4	0	-8	2	-6	22
метод Крамера					
$\Delta =$	4	4	4	8	-768
	0	-8	-2	6	
	-2	2	-8	8	
	0	-8	2	-6	
$\Delta_1 =$	-12	4	4	8	-768
	26	-8	-2	6	
	0	2	-8	8	
	22	-8	2	-6	
$\Delta_2 =$	4	-12	4	8	2304
	0	26	-2	6	
	-2	0	-8	8	
	0	22	2	-6	
$\Delta_3 =$	4	4	-12	8	768
	0	-8	26	6	
	-2	2	0	8	
	0	-8	22	-6	
$\Delta_4 =$	4	4	4	-12	0
	0	-8	-2	26	
	-2	2	-8	0	
	0	-8	2	22	
ОТВЕТ	1	-3	-1	0	

Лабораторная работа 9

Постановка задачі:

1) Створити **список**, для автоматизації предметної області згідно свого варіанту завдання. Кожний запис списку повинен містити не менш як **5 полів** основних типів: текстове, числове, дата й час, поле, що обчислюється.

2) Заповнити список реальним (правдоподібним) умістом, з не менш як **15 записів**.

3) Сформулювати **2 запити** на сортування створеного списку. Ці завдання реалізувати за допомогою пункту Сортування... меню Дані по текстовому, числовому полям або по полю дата.

4) Сформулювати **2 простих запити** на пошук і відбір інформації зі створеного списку за допомогою **Автофільтру**.

5) Сформулювати **не менше 3 складних запитів** на пошук і відбір інформації зі створеного списку. Ці завдання реалізувати за допомогою **Розширеного фільтру** меню Дані/Розширений фільтр по текстовому, числовому полям або по полю дата. З цих запитів:

5.1) як мінімум один запит повинен містити умову типу И,

5.2) як мінімум один запит – умову типу ИЛИ,

5.3) не менш як по одному запиту повинні використовувати наступні можливості автофільтру: пропущені букви (?), довільна кількість пропущених букв (*), умови >, <, =, <>.

Хід роботи:

1. Створив список, для автоматизації предметної області згідно свого варіанту завдання(14) Комунально-побутові послуги: вода, газ, тепло,

каналізація, ремонт, обслуговування будинку, ціна в грн.). Кожний запис списку повинен містити не менш як 7 полів.

2. Заповнив список реальним (правдоподібним) умістом, з 18 записів.
3. Сформулював 2 запити на сортування створеного списку(сортування по даті від нової до старої та сортування стовпця "до сплати" по зростанню. Ці завдання реалізував за допомогою пункту Сортування... меню Дані.
4. Сформулював (на пошук і відбір інформації зі створеного списку за допомогою Автофільтру (меню - Дані). Такі запити:
 1. Автофільтр по назві комунально-побутових послуг згідно алфавіту
 2. Автофільтр ціни послуги, де вона змінилась
5. Сформулював 3 складних запитів на пошук і відбір інформації зі створеного списку. Ці завдання реалізував за допомогою Розширеного фільтру меню Дані\Розширений фільтр по текстовому, числовому полям або по полю дата. Такі запити:
 1. Розширений фільтр. Отфільтруємо згідно умови, "тепло" і "до сплати" більше 300 грн
 2. Розширений фільтр. Отфільтруємо згідно умови, "гаряча вода" і "оплачено" або "каналізація і холодна вода" і "сплачено"
 3. Розширений фільтр. Отфільтруємо згідно умови, "тепло" і "не сплачено".

Задачі з програмування

1. Було зроблено такі практичні завдання з програмування з сайту <https://tutor.amcbridge.com/>
 1. Character на 90%, замість нього було вирішено замінити на Cashbox, який пройшов на 100%.
 2. Geometric progression nth — 100%
 3. If sequence contains element — 100%
 4. Intersection between ray and plane 3D — 100%
 5. Subsequence — 100%

Cashbox

<https://tutor.amcbridge.com/#/projects/98e12adc-72d0-an81-7015-cf450889ty67>

Description

Develop a cashbox program.

Several products are sold in the shop. Each of the products is characterized by product code, price and amount available in stock.

Every day the shop receives a list of orders for the products.

If a required product is not available in the shop, then the requirement is ignored; if less quantity is available, then the shop sells all available amount of the product (and ignores the rest of the order for such a product).

The program must calculate the total price of the goods sold every day.

Input:

An array of triples of integer numbers – product information:

First – product code (unique)

Second – product price

Third - product balance

Split number -1

Days count

An array of pairs of integer numbers - sold products per day

First - product code

Second - quantity of goods

If there are entries with the same product code, then the quantity should be added to the product with such code. The order of entering codes is undefined.

At the end, the array – split number -1

Output:

Array of integer numbers: cost of goods sold per day.

End number -1

Input data guarantees that all input numbers and the results will be less than 2 147 483 647.

All input numbers are separated by spaces.

Project code: Cashbox

Ця програма має такий код

```
#include <iostream>
#include <cstdlib>

using namespace std;

typedef long long type; /* array type */
#define MAX 64 /* stack size for max 2^(64/2) array elements */

void quicksort_iterative(type ** array, unsigned len) {
    unsigned left = 0, stack[MAX], pos = 0, seed = rand();
    for ( ; ; ) { /* outer loop */
        for (; left+1 < len; len++) { /* sort left to len-1 */
            if (pos == MAX) len = stack[pos = 0]; /* stack overflow, reset */
            type pivot = array[left+seed%(len-left)][0]; /* pick random pivot */
            seed = seed*69069+1; /* next pseudorandom number */
            stack[pos++] = len; /* sort right part later */
            for (unsigned right = left-1; ; ) { /* inner loop: partitioning */
                while (array[++right][0] < pivot); /* look for greater element */
                while (pivot < array[--len][0]); /* look for smaller element */
                if (right >= len) break; /* partition point found? */
                type temp = array[right][0];
                array[right][0] = array[len][0];
                array[len][0] = temp;
                temp = array[right][1];
                array[right][1] = array[len][1];
                array[len][1] = temp;
                temp = array[right][2];
                array[right][2] = array[len][2];
                array[len][2] = temp;
            } /* partitioned, continue left part */
        }
        if (pos == 0) break; /* stack empty? */
        left = len; /* left to right is sorted */
        len = stack[--pos]; /* get next range to sort */
    }
}
```

```

int Search_Binary (type ** arr, type left, type right, type key)
{
    type midd = 0;
    while (1)
    {
        midd = (left + right) / 2;

        if (key < arr[midd][0])          // если искомое меньше значения в
ячейке
            right = midd - 1;          // смещаем правую границу поиска
        else if (key > arr[midd][0])    // если искомое больше значения в
ячейке
            left = midd + 1;           // смещаем левую границу поиска
        else
            return midd;               // иначе (значения равны)
                                         // функция возвращает индекс ячейки

        if (left > right)               // если границы сомкнулись
            return -1;
    }
}

int main() {
    type ** arr, i, j, k, l = 0, n = 0, d = 0, sum;
    arr = (type **) calloc(0, sizeof(type[3]));
    for (cin >> i; i != -1; cin >> i) {
        n++;
        cin >> j >> k;
        arr = (type **) realloc(arr, n * sizeof(type[3]));
        arr[n-1] = (type*) calloc(3, sizeof(type));
        arr[n-1][0] = i;
        arr[n-1][1] = j;
        arr[n-1][2] = k;
    }
    quicksort_iterative(arr, n);
    //~ for (i = 0; i < n; i++) {
        //~ for (j = 0; j < 3; j++)
            //~ cout << arr[i][j] << " ";
        //~ cout << endl;
    //~ }
    cin >> d;
    for (i = 0; i < d; i++) {
        sum = 0;
        for (cin >> j; j != -1; cin >> j) {
            cin >> k;
            l = Search_Binary(arr, 0, n - 1, j);
            if (l != -1) {
                if (k <= arr[l][2]) {
                    arr[l][2] -= k;
                    sum += k * arr[l][1];
                } else if (arr[l][2] > 0) {
                    sum += arr[l][2] * arr[l][1];
                    arr[l][2] = 0;
                }
            }
        }
    }
}

```

```

        cout << sum << " ";
    }
    cout << "-1";
    return 0;
}

```

У цій програмі використовується швидке сортування, який має складність $n \log n$, та бінарний пошук, який має складність $\log n$.

У головній програмі виконуються такі дії:

1. Створюються необхідні змінні
2. Забираються вхідні данні інформації про продукти у двовимірний масив, де кожен рядок означає новий продукт, перший стовпчик код продукту, другий — ціну, третій — кількість
3. Сортуються данні за кодом продукту
4. Забирається кількість операцій
5. Циклічно виконуються усі операції, забираючи походу потрібні данні та знаходячи потрібний продукт через бінарний пошук та походу виводиться результат

Програма виконує усі операції за $n \log n$ швидкість.

Character

<https://tutor.amcbridge.com/#/projects/98e12adc-72d0-an81-7015-cf450889ty67>

Description

Develop a cashbox program.

Several products are sold in the shop. Each of the products is characterized by product code, price and amount available in stock.

Every day the shop receives a list of orders for the products.

If a required product is not available in the shop, then the requirement is ignored; if less quantity is available, then the shop sells all available amount of the product (and ignores the rest of the order for such a product).

The program must calculate the total price of the goods sold every day.

Input:

An array of triples of integer numbers – product information:

First – product code (unique)

Second – product price

Third - product balance

Split number -1

Days count

An array of pairs of integer numbers - sold products per day

First - product code

Second - quantity of goods

If there are entries with the same product code, then the quantity should be added to the product with such code. The order of entering codes is undefined.

At the end, the array – split number -1

Output:

Array of integer numbers: cost of goods sold per day.

End number -1

Input data guarantees that all input numbers and the results will be less than 2 147 483 647.

All input numbers are separated by spaces.

Example:

Input:

2 15 10 1 10 20 3 5 50 -1 2 1 10 2 10 -1 1 5 2 5 3 25 -1

Output:

250 175 -1

Ця програма має такий код:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    long long int MaxHP, HP, MP, HealCost, a = 0, b = 0;
    scanf("%lli%lli%lli%lli", &MaxHP, &HP, &MP, &HealCost);
    if (HP > MaxHP) HP = MaxHP;
    if (HP <= 0) {
        printf("0");
        return 0;
    }
    while (a != -1 || b != -1) {
        scanf("%lli", &a);
        if (a == -1) break;
        scanf("%lli", &b);
        if (b == -1) break;
        if (a == 1) {
            HP -= b;
        } else if (a == 2 && MP >= HealCost) {
            MP -= HealCost;
            HP += b;
        } else if (a == 3)
            MP += b;
        if (HP <= 0) {
            printf("0");
            return 0;
        } else if (HP > MaxHP) HP = MaxHP;
    }
}
```

```
    printf("%lli",HP);  
    return 0;  
}
```

Виконує такі дії:

1. Забирає початкові данні. (максимальне здоров'я, поточне здоров'я, поточну кількість мари, вартість зцілення)
2. Перевіряє данні на правильність та виправляє помилки
3. Далі циклічно, доки не отримаємо на вході -1, виконує такі дії:
 1. Забирає поточний номер дії та її величина на яку потрібно виконати дію
 2. Виконує відповідну дію
 3. Робить перевірки отриманих значень
4. Виводить отримане здоров'я та виходить.

Швидкість виконання лінійна.

Geometric progression nth

Description

Find the n-th member of a geometric progression with initial term a and common ratio q. The input data are two double numbers and one integer number separated by spaces. The first number is the initial term of the geometric progression a. The second number is the common ratio q of the geometric progression. The third number is the number of members of the geometric progression n. Input data guarantees that all input numbers and the result will be greater than -9 223 372 036 854 775 808.0, and less than 9 223 372 036 854 775 807.0. Output data must be one double number - the value of the nth member of the geometric progression. The result is considered correct if it is correct up to 8 decimal places, i. e. two real numbers A and B are considered equal if $|A-B| < 1e-8$

Example: Input: 1.0 2.0 3 Output: 4.0

Ця програма має такий код

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    long double b0,q,qn=1;
    int n;
    scanf("%Lf%Lf%i",&b0,&q,&n);
    for (int i=1;i<n;i++)
        qn*=q;
    printf("%.8Lf",b0*qn);
    return 0;
}
```

Реалізація дуже проста, спочатку ми забираємо числа b0 — початковий елемент, q — знаменник прогресії, n — елемент який потрібно забрати

Визраз обчислюється по формулі $b_n = b_1 q^{n-1}$, степінь виконана у вигляді циклу, хоча можна скористатися функцією pow.

Потім після обчислення ми виводимо відповідь.

If sequence contains element

Description

Suppose we are given two sequences of numbers $A[0], A[1], \dots, A[n]$ and $B[0], B[1], \dots, B[m]$. Print a sequence $C[0], C[1], \dots, C[n]$ where $C[i]$ is 1 if element $A[i]$ is in sequence B, and 0 otherwise. The input data are positive integer numbers - the elements of the sequences. All numbers are separated by spaces. The number of members of the sequences isn't known in advance. Sequences end with number -1, and -1 is not a member of the sequences. First numbers are elements of sequence A. In the end of the sequence A is number -1. -1 is not a member of the sequence A or the sequence B. Next numbers are elements of the sequence B. In the end of the sequence B is number -1. -1 is not a member of the sequence B. Input data guarantees that the inputs and the result will be less than 2 147 483 647. Output data must be a sequence of zeroes and ones. Sequence must end with number -1.

Project code: SequenceContainsOrNot

Example:

Input:

1 2 4 3 6 5 -1 1 6 3 7 8 -1

Output:

1 0 0 1 1 0 -1

Ця програма має такий код

```
#include <stdio.h>
#include <stdlib.h>

typedef long long type; /* array type */
#define MAX 64 /* stack size for max 2^(64/2) array elements */

void quicksort_iterative(type array[], unsigned len) {
    unsigned left = 0, stack[MAX], pos = 0, seed = rand();
    for ( ; ; ) { /* outer loop */
        for (; left+1 < len; len++) { /* sort left to len-1 */
            if (pos == MAX) len = stack[pos = 0]; /* stack overflow, reset */
            type pivot = array[left+seed%(len-left)]; /* pick random pivot */
            seed = seed*69069+1; /* next pseudorandom number */
            stack[pos++] = len; /* sort right part later */
        }
    }
}
```

```

        for (unsigned right = left-1; ; ) { /* inner loop: partitioning */
            while (array[++right] < pivot); /* look for greater element */
            while (pivot < array[--len]); /* look for smaller element */
            if (right >= len) break; /* partition point found? */
            type temp = array[right];
            array[right] = array[len]; /* the only swap */
            array[len] = temp;
        } /* partitioned, continue left part */
    }
    if (pos == 0) break; /* stack empty? */
    left = len; /* left to right is sorted */
    len = stack[--pos]; /* get next range to sort */
}

int Search_Binary (long long int arr[], long left, long right, long key)
{
    long long int midd = 0;
    while (1)
    {
        midd = (left + right) / 2;

        if (key < arr[midd]) // если искомое меньше значения в ячейке
            right = midd - 1; // смещаем правую границу поиска
        else if (key > arr[midd]) // если искомое больше значения в ячейке
            left = midd + 1; // смещаем левую границу поиска
        else // иначе (значения равны)
            return midd; // функция возвращает индекс ячейки

        if (left > right) // если границы сомкнулись
            return -1;
    }
}

int main() {
    long long int *a,*b,x,n=0,m=0;
    a = (long long*) calloc(1,sizeof(long long));
    b = (long long*) calloc(1,sizeof(long long));
    scanf("%lli",&x);
    while (x != -1) {
        n++;
        a[n-1] = x;
        a = (long long*) realloc(a,(n+1)*sizeof(long long));
        scanf("%lli",&x);
    }
    scanf("%lli",&x);
    while (x != -1) {
        m++;
        b = (long long*) realloc(b,(m+1)*sizeof(long long));
        b[m-1] = x;
        scanf("%lli",&x);
    }
    quicksort_iterative(b,m);
    //~ for (int i=0;i<n;i++)
        //~ printf("%lli ",a[i]);
}

```

```

    //~ printf("\n");
    //~ for (int i=0;i<m;i++)
        //~ printf("%lli ",b[i]);
    //~ printf("\n");
    for (int i=0;i<n;i++) {
        if (Search_Binary(b,0,m-1,a[i]) != -1)
            printf("1 ");
        else
            printf("0 ");
    }
    printf("-1");
    free(a);
    free(b);
    return 0;
}

```

Тут виконуються такі дії:

1. Спочатку створюємо необхідні змінні
2. Потім забираємо усі вхідні данні, масив А та В
3. Сортуємо масив В
4. Потім перевіряємо чи є елементи з А у В за допомогою Search_Binary
5. Звільнюємо пам'ять

Програма має $n \cdot \log n$ швидкість виконання.

Intersection between ray and plane 3D

Description

Suppose we are given a ray and a plane in 3D space. Find the intersection between the ray and the plane. The ray is specified by the initial point of the ray and by its direction vector. The plane is specified by the point lying on the plane and by its normal vector. The input data are 12 double numbers. The first three numbers are coordinates of the initial point of the ray. The second three numbers are coordinates of the direction vector of the ray. The third three numbers are coordinates of the point lying on the plane. The last three numbers are coordinates of the normal vector of the plane. Input data guarantee that lengths of direction vector and normal vector are greater than $1e-8$. All input numbers are separated by spaces. Input data guarantees that inputs and the results will be less than 9 223 372 036 854 775 807.0. Output data must be: 1)if the ray and the plane don't intersect output is integer number 0; 2)if the ray and the plane intersect output data is integer number 1 and coordinates of the point of intersection of the ray and the plane. All numbers must be separated by spaces; 3)if the ray is in the plane output data is integer number 5 and coordinates of initial point of the ray, and coordinates of the normalized direction vector of the ray. All numbers must be separated by spaces. The result is considered correct if it is correct up to 8 decimal places, i. e. two real numbers A and B are considered equal if $|A-B| < 1e-8$

Project code: IntersectionBetweenRayAndPlane3D

Ця програма має такий код

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

int main() {
    double ax, ay, az, ux, uy, uz, bx, by, bz, vx, vy, vz, k, D, len, p, l;
    scanf("%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf", &ax, &ay, &az, &ux, &uy, &uz, &bx, &by, &bz, &vx, &vy, &vz);
    k = ux*vx+uy*vy+uz*vz;
    D = -vx*bx -vy*by -vz*bz;
    p = vx*ax+vy*ay+vz*az+D;
    if (k == 0 && p == 0) {
        len = sqrt(ux*ux + uy*uy + uz*uz);
        ux /= len; uy /= len; uz /= len;
```



```

        printf("5 %.8lf %.8lf %.8lf %.8lf %.8lf %.8lf",ax,ay,az,ux,uy,uz);
    } else if ((k < 0 && p > 0) || (k > 0 && p < 0)) {
        l = -p/(vx*ux+vy*uy+vz*uz);
        printf("1 %.8lf %.8lf %.8lf",ax+ux*l,ay+uy*l,az+uz*l);
    } else printf("0");
    return 0;
}

```

У даній програмі можливі три випадки коли промінь:

1. лежить на площині
2. перетинає площину
3. не перетинає і не лежить на площині

Перший випадок виконується коли нормальний вектор площини та напрямлений вектор променя перпендикулярні, а початкова точка лежить на площині.

Другий випадок можливий якщо вектори мають гострий кут, при умові що знаходяться по одну сторону площини, або якщо вектори мають тупий кут, при умові що вектори по різну сторону площини

Третій випадок буде якщо усі вище сказані умови не виконуються.

Кут знаходимо через скалярний добуток:

$$\cos \alpha = \frac{|\vec{a}| * |\vec{b}|}{\vec{a} * \vec{b}}, \text{ де } \vec{a} \text{ і } \vec{b} - \text{це вектори, } \alpha - \text{кут між ними}$$

А перетин знаходимо

$$d = \frac{AM_x + BM_y + CM_x + D}{\vec{a} * \vec{b}}, \text{ де } \vec{a} - \text{нормальний вектор площини з координатами } (A, B, C),$$

\vec{b} – напрямлений вектор прямої, M – будь – яка точка на площині

Subsequence

Description

Suppose we are given two sequences of numbers. Check whether the second sequence is a subsequence of the first sequence. The sequence $B[0], \dots, B[m]$ is called a subsequence of the sequence $A[0], \dots, A[n]$, if there exists such increasing sequence of indexes n_k , that for all k from 0 to m : $B[k]=A[n_k]$. The input data are integer numbers. All numbers are separated by spaces. First number n is a number of members of the first sequence. Next n numbers are elements of the first sequence. Next number m is a number of members of the second sequence. Next m numbers are elements of the second sequence. Input data guarantees that the inputs and the result will be less than 2 147 483 647. Output data must be 1 if the second sequence is a subsequence of the first sequence, and 0 - otherwise.

Project code: Subsequence

Example:

Input:

5 1 2 3 4 5 2 1 4

Output:

1

Input:

5 1 2 3 4 5 2 4 1

Output:

0

Ця програма має такий код:

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

int main() {
    long int n,m,*a,*b,j=0,k=0;
    scanf("%li",&n);
```

```

a = new long int[n];
for (int i=0;i<n;i++)
    scanf("%li",&a[i]);
scanf("%li",&m);
b = new long int[m];
for (int i=0;i<m;i++)
    scanf("%li",&b[i]);
for (int i=0;i<m;i++) {
    while (a[j]!=b[i] && j<=n)
        j++;
    if (j==n) {
        printf("0");
        return 0;
    } else if (a[j]==b[i]) k++;
}
if (k == m) printf("1");
else printf("0");
delete a;
delete b;
return 0;
}

```

Виконує такі дії:

1. Спочатку створюємо необхідні змінні
2. Потім забираємо усі вхідні данні, масив А та В
3. Проходимось по масиву В і одночасно по А, якщо ми прийшли у кінець А, то виводимо нуль і виходимо, якщо ні, то виводимо 1.

Калькулятор на Android

Програму написано мовою Java за допомогою середовища Android Studio

3. У проекті використано бібліотеки для роботи з великими числами `java.math.BigDecimal`. Проект містить такі модулі:

- `java/top/elfiny/calculator/Calculator.java`
- `java/top/elfiny/calculator/CalculatorModel.java`
- `java/top/elfiny/calculator/Number.java`
- `res/layout/activity_calculator.xml`

Опис модулів:

1. `class Calculator` — потрібен як моделлю калькулятора, яка робить усі обчислення та вивід.
2. `class CalculatorModel` — у ньому записані основні правила обчислення калькулятора,
3. `class Number` — клас одного числа, у якому число зберігається у трьох варіантах: введеній строчці, числовому форматі, виведеному тексту
4. `res/layout/activity_calculator.xml` — тут дизайнером описаний вид калькулятора

Методи класу `Calculator`:

1. `onCreate` — виконується при запуску програми, оброблює доступні операції та створює модель `CalculatorModel`
2. `typeNumberById` — записує число яке написано в параметрах кнопки
3. `typeDot` — перетворює число у число з плаваючою точкою
4. `clear` — затирає поточне число
5. `reload` - синхронізує дані моделі з видом
6. `makeOperation` — робить операцію яка написана в параметрах кнопки

7. reset — скасовує усі операції

Методи класу Calculator:

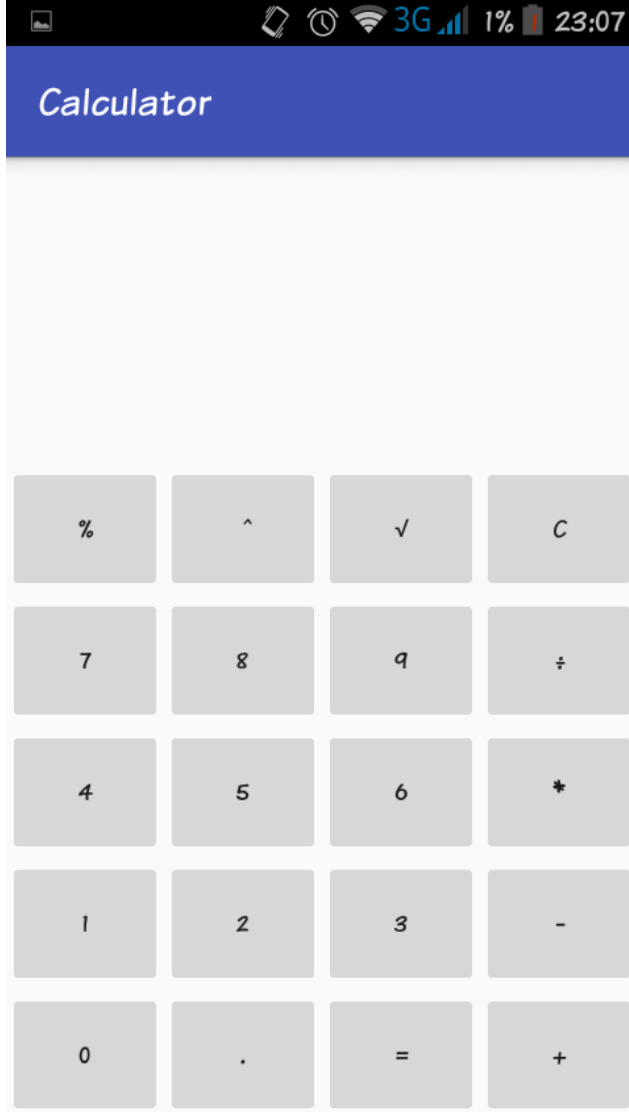
1. CalculatorModel — конструктор, підготовує програму до роботи
2. isOperation — перевіряє чи запросили ми операцію
3. appendNumber — додає цифру до числа
4. getActiveNumber — забирає поточний текст на екран
5. getLastFormula — забирає останнє виконання
6. clearActive — стриляє поточне число
7. makeOperation — робить операцію, або зберігає її
8. reset — скасовує всі операції

Методи класу Number:

1. Number — конструктор який створює пусте число
2. render — генерує число у текст
3. getNumber — забирає поточне число у текстовому вигляді
4. getBigDecimal — забирає число у числовому вигляді
5. setNumber — задає число
6. append — додає цифру до числа
7. makeNegative — робить число негативним
8. operation — робить операцію з іншим числом
9. sqrtNewtonRaphson — обчислює корінь за заданими параметрами
10. bigSqrt — обчислює корінь з стандартними параметрами
11. roundToMax — робить число красивим, видуючи цифри після 30 цифр після коми та непотрібні нулі які стоять після коми
12. isNull — перевіряє не нульове це число

Приклади роботи програми:

Початок роботи програми



Приклад роботи



[illegible]

A screenshot of an Android calculator application. The status bar at the top shows the time 23:07, 1% battery, and 3G connectivity. The app's title bar is blue with the word "Calculator" in white. The main display area is white and shows a large "Err" message in a grey, sans-serif font. Below the display is a grid of 16 buttons arranged in 4 rows and 4 columns. The buttons are light grey with black text. The first row contains: a percentage sign (%), a power symbol (^), a square root symbol (√), and a clear button (C). The second row contains: the digit 7, the digit 8, the digit 9, and a division symbol (÷). The third row contains: the digit 4, the digit 5, the digit 6, and a multiplication symbol (*). The fourth row contains: the digit 1, the digit 2, the digit 3, and a subtraction symbol (-). The bottom row contains: the digit 0, a decimal point (.), an equals sign (=), and an addition sign (+).

Висновок

Лабораторна робота №1-2

Знадобились навички роботи з Word, зробив:

- примітки,
- границі,
- змінювати колір фону і тексту,
- змінювати розмір полей,
- встановлювати автоматичний перенос,
- гарячі клавіші для різної дій над текстом,
- робота з колонтитулами,
- заміна конкретних слів або символів у тексті на інші.

Лабораторна робота №3

Знадобились навички роботи з Word, зробив:

- Маркіровку
- Відступ тексту
- Багаторівневий список
- Колонки

Лабораторна робота №4

Знадобились навички роботи з Word, зробив:

- Об'єднати комірки таблиці,
- Розташовувати всю таблицю по центру,
- Використовувати формули,

- Створювати кругову діаграму,
- Налаштовувати Легенду та Підписи діаграми.

Лабораторна робота №5

Знадобились навички роботи з Word, зробив:

- Введення формули за допомогою Microsoft Equation 3.0,
- Вставляння автоматичну нумерацію формул,
- Використовування формули,

Лабораторна робота №7

Знадобились навички роботи з Excel, зробив:

- Формувати таблицю за заданою формою,
- Використовувати відповідні формули
- Апроксувати введені значення математичною функції
- Працювати з Лінією тренда

Лабораторна робота №8

Знадобились навички роботи з Excel, зробив:

- Рахувати детермінант за допомогою формули МОПРЕД
- Розв'язання систем лінійних алгебраїчних рівнянь методом Крамера
- Розв'язання систем лінійних алгебраїчних рівнянь матричним методом
- Транспонувати матрицю

Лабораторна робота №9

Знадобились навички роботи з Excel, зробив:

- Сортування
- Автофільтр

Задачі з програмування

Знадобились навички програмування:

- Швидке сортування
- Бінарний пошук
- Заповнення через динамічний масив
- Вхід та вихід даних

А також навички з рішень задач по трьох вимірній геометрії

Калькулятор на смартфоні

Навчився працювати з Android Studio, вивчені нові модулі:

- Строки
- Довга арифметика з плаваючою точкою
- Дизайнер в Android Studio

Використані джерела

1. Iterative Quicksort / C / Algorithm Implementation/Sorting/Quicksort - Wikibooks, open books for an open world
https://en.wikibooks.org/wiki/Algorithm_Implementation/Sorting/Quicksort#Iterative_Quicksort (дата звернення 18.06.18)
2. Двоичный (бинарный) поиск в массиве. C++ | PureCodeCpp
<http://purecodecpp.com/archives/1977>
3. Configure Android Studio | Android Developers
https://developer.android.com/studio/intro/studio-config#low_memory
4. Download Android Studio and SDK tools | Android Developers
<https://developer.android.com/studio/>
5. Інформаційні технології, 2015, Литвин О.І., канд. техн. наук, доцент
6. Інформатика та програмування, навчальний посібник, 2014, Карімов І. К.