



МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт з дисципліни
«АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ»
Частина 1
МОВА «С»
для студентів напрямку 122 – «Комп'ютерні науки»

Дніпро - 2019

УДК 681.3.06+519.68

Методичні вказівки до виконання лабораторних робіт з дисципліни «Алгоритмізація та програмування». Частина 1. Мова «С». Для студентів напряму 122 - «Комп'ютерні науки» Укл.: О.О Кавац Н.Л. Дорош, Т.М. Фененко.– Дніпро: НМетАУ, 2019. – 51 с.

Методичні вказівки є першою частиною комплексу навчально-методичних матеріалів з дисципліни «Алгоритмізація та програмування»; містять теоретичні положення з мови програмування С, приклади програм, завдання з питань використання базових засобів мови С для самостійного виконання.

Призначені для студентів напряму 122 – «Комп'ютерні науки», а також для слухачів курсів підвищення кваліфікації, студентів і аспірантів інших спеціальностей.

Укладачі: О.О Кавац, к.т.н., доц.
Н.Л. Дорош, к.т.н., доц.
Т.М. Фененко, ст. викл.

Відповідальний за випуск О.І. Михальов, д-р техн. наук, проф.

Рецензент О.І. Дерев'янка, канд. тех. наук., доц. (ДНУ)

Друкується за авторською редакцією.

Затверджено на засіданні НМК,
протокол № 5/14-15 від 25.05.2019

Підписано до друку 20.04.2019. Формат 60x84 1/16. Папір типогр. Друк різнограф. Облік.- вид. арк 3,18. Умов. друк. арк. 3,14.

Тираж 100 пр. Замовл. № 30/2.

Національна металургійна академія України.
49600, Дніпро, пр. Гагаріна, 4

ЗМІСТ

Лабораторна робота №1	4
Робота в середовищі MS DOS	4
Лабораторна робота №2	8
Інтегрована оболонка NORTON COMMANDER.....	8
Лабораторна робота №3	10
Лнійні обчислювальні процеси. Обчислення площини фігури	10
Лабораторна робота №4	16
Обчислення арифметичних виразів.....	16
Лабораторна робота №5	22
Організація розгалужень	22
Лабораторна робота №6	27
Циклічні обчислювальні процеси. Робота з табличними даними.	27
Лабораторна робота №7	31
Циклічні обчислювальні процеси. Робота з одновимірними масивами	31
Лабораторна робота №8	34
Циклічні обчислювальні процеси. Робота з багатовимірними масивами	34
Лабораторна робота №9	38
Робота зі структурами.....	38
ЛІТЕРАТУРА	41
ДОДАТОК 1	42
ДОДАТОК 2.....	51

Лабораторна робота №1 РОБОТА В СЕРЕДОВИЩІ MS DOS.

Мета: Навчитися виконувати базові команди та операції у середовищі MS DOS.

Теоретичні відомості

Діалог користувача з дисковою операційною системою (DOS) здійснюється в формі команд - рядків символів, які вводяться користувачем у відповідь на запрошення DOS. Введення команд DOS закінчується натисненням клавіші [ENTER]. Команди можуть набиратися як великими, так і малими буквами.

Майже уся комп'ютерна інформація зберігається на різних носіях у вигляді файлів – іменованих областях пам'яті на диску або іншому фізичному носії. Позначення файла складається з імені файла: набору цифро-буквених символів завдовжки 1 - 8 символів (рядкові букви латинського алфавіту та символи ~ ' & @ () % { } _ # ` \$), крапки, що розділяє і необов'язкового розширення, яке вказує тип (характер інформації у файлі), завдовжки 1 - 3 символ. Імена файлів реєструються на дисках в каталогах (директоріях).

Каталог - це спеціальна область на диску, в якій зберігаються імена файлів, час їх створіння і останнього оновлення, розміри, властивості (атрибути) і т.д. Для позначення каталогів використовують імена не більше ніж з 8 цифро-буквених символів. На кожному магнітному диску є каталог найвищого рівня, який називають кореневим (позначка - ім'я диску). В ньому реєструються файли і каталоги 1-го рівня, а в них файли і каталоги 2-го рівня і т.д. . Запит DOS, як правило, містить інформацію про поточний диск та каталог.

Приклади:

A:\> - диск A, кореневий каталог.

C:\DOS> - диск C, каталог DOS.

C:\STUD\ITS - диск C, каталог ITS, який знаходиться у каталозі STUD.

Звернення до директорій можна виконати, вказавши маршрут :

[<ім.'я диску>:][\] [<шлях>\]<ім'я каталогу>

(дужками "[" і "]" позначаються необов'язкові елементи, а в "<" і ">" назви основних понять)

де <ім'я диску> - це одна із букв латинського алфавіту; "\" - позначення кореневого каталогу або роздільника між іменами каталогів; <шлях> - послідовність імен каталогів, розділених символом "\";

Для позначення надкаталогу (батьківського каталогу) використовують символ "..".

Ім'я файлу фактично є його адресою, тому один каталог не може містити в собі два файли, що мають однакове ім'я. Звернення до файлів можливі за допомогою вказівки спеціального маршруту, який прийнято називати повне ім'я файлу: **[<ім'я диску>:][\] [<шлях>\]<им'я файлу>]. [<тип>].**

Засоби редагування команд

[BACKSPACE] (стрілка вліво над [ENTER]) - витирає попередній символ.

← → ↑ ↓ стрілки переміщення курсору відповідно вліво на один символ, вправо на один символ, вниз на один рядок, вверх на один рядок.

[DEL] - витирає символ над курсором.

[ESC] - знищення всієї команди.

[INS] - включення або виключення режиму вставки.

Допомога

Для одержання короткої допомоги по певній команді необхідно ввести команду і знак питання у такому форматі: ...> команда /?

Наприклад: C:\>dir /? • вивести допомогу для команди dir.

Для виклику загальної допомоги необхідно ввести команду HELP або HELP <тема>

Основні команди MS DOS

Команди роботи з файлами:

Зміна поточного диска:

Формат команди : X: - де X - ім'я бажаного диску. Наприклад: A: - перехід на дискету.

Зміна поточного каталогу;

Формат команди : **CD [ДИСКОВОД:] ШЛЯХ**

Наприклад: **cd ** - перейти в кореневий каталог поточного диску.

cd \STUD - перейти в каталог STUD.

cd .. - перейти на один каталог назад.

Перегляд змісту каталогу:

Формат команди: **DIR [ДИСКОВОД:] [ШЛЯХ] [ІМ'Я ФАЙЛА]**

Найбільш вживані параметри:

/p - поєкранний вивід заголовку каталога.

/w - вивід даних в широкому форматі; виводяться лише імена файлів та підкаталогів по 5 записів в кожному рядку.

/o - вивід інформації у впорядкованому вигляді: спочатку виводяться імена підкаталогів у алфавітному порядку, а потім імена файлів у алфавітному порядку.

Наприклад: **dir** - вивести зміст поточного каталогу.

dir *.exe - вивести файли з розширенням exe з поточного каталогу.

dir A: - вивести зміст дискети.

dir C:\p - вивести зміст кореневого каталогу C:\ поєкранно.

dir A:\ITS /w - вивести зміст каталогу ITS у широкому форматі.

Створення каталогу:

Формат команди: **MD [ДИСКОВОД:]ШЛЯХ**

Наприклад: md ABC - створення каталогу ABC в поточному каталозі.

md A:\MY_DOC - створення на дискеті каталогу MY_DOC.

Знищення каталогу:

Формат команди: **RD [ДИСКОВОД:]ШЛЯХ**

Наприклад: **rd A:\TEMP_DOC** - знищити на дискеті каталог TEMP_DOC.

rd TMP - знищити каталог TMP в поточному каталозі.

Копіювання файлів:

Формат команди: **COPY ІМ'Я_ФАЙЛУ_1 ІМ'Я_ФАЙЛУ_2**

Наприклад: **copy a.txt b.txt** - копіювання змісту файлу a.txt у файл b.txt (файли знаходяться в поточному каталозі).

copy A:*.txt D:\ITS - копіювання файлів з розширенням txt, що знаходяться на дискеті A:\ на диск D:\ в каталог ITS. В даному випадку при копіюванні файли зберігають свої імена.

copy D:\ITS\lab1.txt A:\my_lab.txt - копіює файл із D:\ITS на дискету, при цьому змінюючи ім'я на my_lab.txt.

Об'єднання файлів:

Формат команди:

COPY ІМ'Я_ФАЙЛУ_1 + ІМ'Я_ФАЙЛУ_2 + ...+ ІМ'Я_ФАЙЛУ_N ІМ'Я_ФАЙЛУ_ПРИЙМАЧА , де файли 1,2...N - файли, що об'єднуються; ім'я файлу-приймача - ім'я результуючого файлу, де зберігається об'єднана інформація з файлів 1 ...N.

Якщо в команді не задане ім'я файлу-приймача, то об'єднана інформація запишеться за адресою та іменем першого файлу.

Наприклад:

copy file1.txt + file2.txt + file3.txt newfile.txt

Об'єднується інформація з файлів file1.txt file1.txt file1.txt у файл newfile.txt , всі файли знаходяться в поточному каталозі.

copy C:\lab.txt+D:\lab2.txt - об'єднання інформації файлів lab.txt кореневого каталогу C:\ та файла lab2.txt кореневого каталогу D:\. Результатом буде файл C:\lab.txt.

Переміщення файлів в інший каталог:

Формат команди: **MOVE [/Y] ІМ'Я_ФАЙЛУ ІМ'Я_КАТАЛОГУ**

Ключ /y застосовується, якщо немає потреби в запиті про дозволення перезапису файла.

Наприклад: **move *.txt D:\ITS** - пересилка всіх файлів з розширенням txt з поточного каталогу у каталог D:\ITS; **move /y lab.txt C:** - пересилка файлу lab.txt на диск C:\ в кореневий каталог без запиту про перезапис файлу (якщо такий вже існує на C:\).

Створення текстових файлів:

Формат команди: **COPY CON ІМ'Я_ФАЙЛУ**

Запис файлу закінчується натисненням [F6] або [CTRL]/[z].

Наприклад; **copy con A:\mydoc.txt** - створення на дискеті текстового файлу mydoc.txt/ **copy con ABC.txt** - створення файлу ABC.txt в поточному каталозі.

Знищення файлів:

Формат команди: **DEL ІМ'Я_ФАЙЛУ**

Наприклад: **del *.txt** - знищення всіх файлів з розширенням txt в поточному каталозі. **del C:\?a.txt** - знищення у кореновому каталозі диска C усіх файлів з розширенням dos, в іменах яких друга буква "a", а перша будь-яка.

Перейменування файлів:

Формат команди: **REN ІМ'Я_ФАЙЛУ_1 ІМ'Я_ФАЙЛУ_2**, де **ІМ'Я_ФАЙЛУ_1** - початкова назва файлу, **ІМ'Я_ФАЙЛУ_2** - нове ім'я файлу.

Наприклад: **ren lab.txt lab_its.txt** - перейменування файлу lab.txt поточного каталогу на lab_its.txt; **ren *.txt *.doc** - перейменовує в поточному каталозі всі файли з розширеннями txt у однойменні файли з розширенням doc, тобто відбувається зміна розширення групи файлів.

Перегляд файлу:

Формат команди: **TYPE ІМ'Я_ФАЙЛУ**

Наприклад: **type a.txt** - вивести на екран зміст файлу a.txt із поточного каталогу.
type A:\top.txt - вивести зміст файлу top.txt з кореневого каталогу дискети. Команди роботи з каталогами:

Послідовний перегляд даних частинами розміром в один екран.

Формат команди:

MORE [ДИСК:] [ШЛЯХ] ІМ'Я ФАЙЛУ або
MORE < [ДИСК:] [ШЛЯХ] ІМ'Я ФАЙЛУ або
ІМ'Я КОМАНДИ | MORE , де

[диск:][шлях]ім'я файлу - задає файли, які відображаються по фрагментах; ім'я команди - команда, висновок якої відображається на екрані.

Завдання

Завдання виконувати поетапно, записуючи кожен етап у зошиті з командами, реакцією комп'ютера та коментарями.

1. Перейти до диска C:, та переглянути його зміст.
2. Створити на ньому каталог STUDENT, та вивести його зміст на екран, переконавшись, що каталог є пустим..
3. У каталозі STUDENT створити файл readme.tmp, який містить дату проведення заняття.
4. У каталозі STUDENT зробити підкаталог 2-го рівня ITS(номер групи), наприклад, ITS-02V.
5. У каталозі ITS створити файл file2.doc, який містить П.І.Б. студента і групу.
6. Скопіювати файл readme.tmp у каталог ITS.
7. Об'єднати файли під новим ім'ям lab1.doc, вивести його зміст на екран, та з'ясувати розмір нового файлу.
8. Змінити розширення файлів з doc на txt.
9. Знищити файли readme.tmp, та file2.txt.
10. Зробити каталог HELP у каталозі ITS
11. Записати у файл help.txt інформацію про команду more, і вивести її на екран.
12. Дописати у файл help.txt інформацію про команду xcopy, вивести на екран, застосувавши команди type і more.
13. Знищити файл help.txt і каталог HELP.

Лабораторна робота №2

ІНТЕГРОВАНА ОБОЛОНКА NORTON COMMANDER

Мета: Навчитися працювати у середовищі NORTON COMMANDER.

Теоретичні відомості

Програми Norton Commander і Volkov Commander (скорочено NC і VC) є одними з найпопулярніших програм-оболонок для роботи з операційною системою DOS, на їх основі було розроблено безліч інших сервісних програм-оболонок.

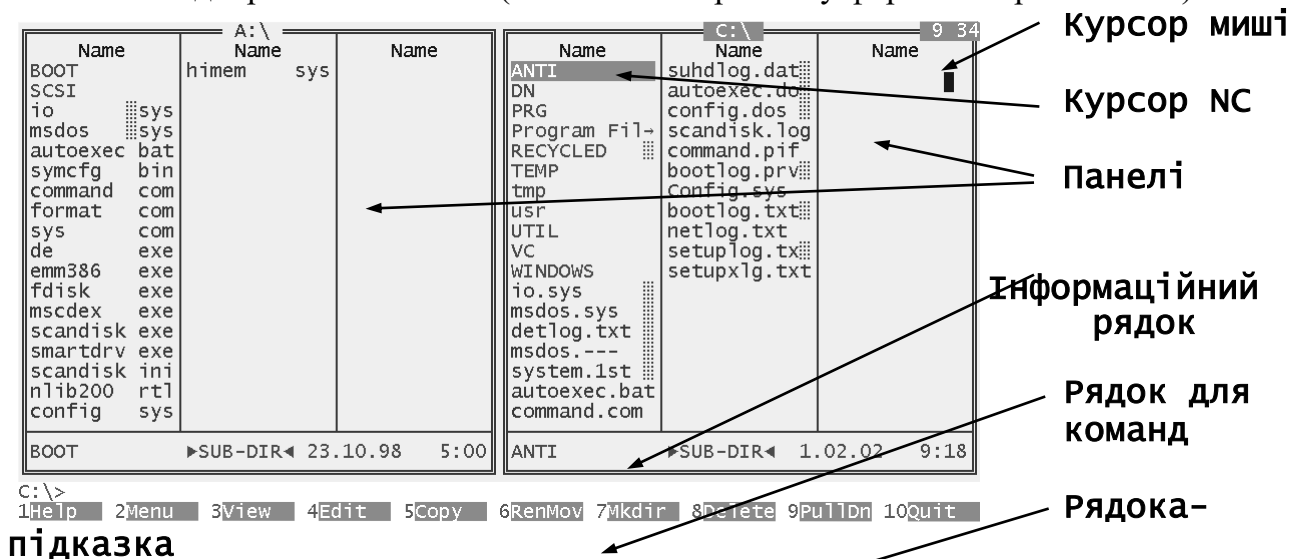
Запуск програми здійснюється набором в командному рядку NC (VC).

Зовнішній вигляд програми показаний на малюнку. На ньому розрізняють наступні елементи: два прямокутні вікна, обмежені подвійною рамкою – називатимемо їх панелями. Нижче цих панелей розташовується командний рядок із запрошенням DOS – в ній можна виконувати команди MS DOS. Ще нижче розташовується рядок-підказка, який состоїть зі значень функціональних клавіш.

Зміст панелей Norton Commander

В кожній панелі NC може зображатися: зміст каталогу - один з основних режимів, дерево каталогів, зведена інформація, зміст файла і інші.

В режимі відображення дерева каталогів зверху виводиться ім'я диска і каталогу, зміст яких відображається нижче(звичайно в короткому форматі в три колонки).



Внизу розташовується інформаційний рядок, зміст якого залежить від вибраного об'єкту (на який вказує курсор: каталог – SUB DIR, або файл)

DOC	NWLITE	TELEFON	config	cfg	help	htm	zakladki	txt
DOC			config.cfg					
			2539	31.12.01				

При роботі з програмою на екрані ми бачимо декілька курсорів: курсор миші, курсор в командному рядку і курсор NC на одній з панелей.

Введемо наступні позначення: панель, на якій встановлений курсор, називатимемо активною; диск, каталог, файл, які вибрані на активній панелі – поточними.

Для управління панелями NC можуть бути використані наступні клавіші:

TAB – зміна активної панелі

Alt+F1 (або Alt+F2) - вивести в лівій (або правій) панелі меню вибору дисків.

Ctrl+F1 (або Ctrl+F2) – вивести або сховати ліву (праву) панель.

Ctrl+O – прибрати панелі з екрану або вивести панелі на екран.

Ctrl+P - прибрати одну з панелей (не поточну) з екрану / вивести панель на екран.

Ctrl U - поміняти панелі місцями.

Ctrl+L - показ зведеної інформації про диск і каталог на неактивну панель.

Зміна форми виводу інформації про файли і каталоги

F9, “L”(або “R”), “B” - коротка форма змісту на лівій (або правій) панелі;

F9”, “L”(або “R”), “F” - повна форма змісту на лівій (або правій) панелі;

F9”, “L”(або “R”), “T” - вивід в лівій (або правій) панелі дерева каталогів на диску;

Зміна порядку виводу інформації про файли і каталоги;

F9”, “L”(або “R”), “N” (або “Ctrl”+”F3”) - сортування на ім'я;

F9”, “L”(або “R”), “X” (або “Ctrl”+”F4”) - сортування по розширенню;

F9”, “L”(або “R”), “M” (або “Ctrl”+”F5”) - сортування за часом;

F9”, “L”(або “R”), “S” (або “Ctrl”+”F6”) - сортування за розміром;

Виконання команд MS DOS.

Команди вводяться за допомогою клавіатури і закінчуються натисканням клавіші ENTER.

Для редагування команд використовують наступні клавіші: “←“ “→“, “Home”, “End”, “Ctrl”+”←“, “Ctrl”+”→“, “Ctrl”+”S”, “Ctrl”+”D”.

Проглянути в командному рядку раніше введені команди DOS можна, натиснувши клавіші “Ctrl”+”E“. При натисненні “Ctrl”+”X“ буде виведена команда, наступна за поточною.

“Alt”+”F8” – показ на екрані списку раніше введених команд.

Швидкий пошук інформації :

“Alt” + ”F7” - **Find File** - пошук файлу або каталогу на диску.

“Alt” + ”перші букви імені каталогу або файлу - **Search** - пошук каталогу на диску.

“Alt” + ”F10” - **Speed search** - пошук директорія по дереву каталогів.

Функціональні клавіші NC

F1 - Help – довідкова інформація.

F2 - Menu - запуск команд, вказаних у файлі меню, який складається користувачем.

F3 - View - перегляд файлу. Перегляд текстових файлів. Також можна переглядати документи інших форматів, при наявності відповідних програм-переглядачів.

F4 - Edit - редагування файлу. Для редагування може бути використано вбудований редактор NC або будь-який інший редактор, який вказано користувачем.

F5 - Copy - копіювання файлу. В середині екрану з'являється запит про те, куди копіювати файл. По замовчанню файл копіюється в каталог, зображений на іншій панелі. Можна набрати інше ім'я каталогу. Потім для копіювання необхідно натиснути ENTER, для скасування команди - ESC.

F6 — Renov - перейменування файлу (каталогу) або пересилання файлу в інший каталог. Можна задати нове ім'я файлу (каталогу) або ім'я каталогу, в який необхідно переслати даний файл. Для початку перейменування або пересилання необхідно натиснути ENTER, для скасування команди - ESC.

F7 - Mkdir - створення підкаталогу.

F8 - Delete - знищення файлу або підкаталогу.

F9 - PullDn - виведення меню, яке містить режими роботи NC (верхня лінійка меню).

F10 - Quit - вихід з Norton Commander.

Створення нового файлу

SHIFT+ F4 – редагування файлу с запитом його імені. Також цю команду можна використовувати для створення нового файлу.

Виділення файлів

Виділення окремого файлу або каталогу виконується з допомогою клавіши Ins. Додаткове натискання цієї клавіши - скасовує виділення. Щоб виділити групу файлів (скасувати виділення групи файлів) необхідно в правій частині клавіатури натиснути плюс + (мінус -), та задати маску файлів, які необхідно виділити (або для яких необхідно скасувати виділення). Для інвертування виділення (щоб зробити виділені файли невиділеними і навпаки) необхідно в правій частині клавіатури натиснути на клавішу * (зірка). При натисканні цієї клавіши підкаталоги залишаються невиділеними.

Завдання

Завдання виконувати поетапно, записуючи кожен етап у зошиті з командами, реакцією комп'ютера та коментарями.

1. Перейти на лівій панелі на диск C: ;
2. Проглянути вміст каталогу VC, використовуючи повну, а потім скорочену форми змісту. Провести сортування файлів поточного директорія по імені файла, а потім по розширенню.
3. Визначити кількість та загальний об'єм файлів у корені диску C: ;
4. Створити каталог WORK_NC, який розташовано у каталозі C:\STUD\ITS(номер групи).
5. Створити файл first.txt у каталозі ITS. (у файлі розташована інформація П.І.Б. студента, дата проведення заняття і назва заняття).
6. Перенести файл first.txt у каталог work_nc.
7. Створити файл second.txt у каталозі ITS. (у файлі розташована інформація про загальний об'єм диску C: і про вільне місце на диску C:).
8. Додати у файл first.txt інформацію про кількість оперативної пам'яті.
9. За допомогою команд DOS об'єднати два файли.
10. Переглянути зміст нового файлу.
11. Змінити атрибути файлу на read-only.
12. Виконати пошук усіх файлів з розширенням txt на диску C:.
13. Знищити каталог C:\STUD\ITS\WORK_NC разом з його файлами.

Лабораторна робота №3

ЛІНІЙНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ.

ОБЧИСЛЕННЯ ПЛОЩИНИ ФІГУРИ

Мета: Навчитися використовувати арифметичні операції та складати прості програми.

Завдання

Написати програму обчислення заштрихованої площини фігури, згідно з варіантом.

Теоретичні відомості

Операції мови C.

Операція	Найменування операції	Приклад	Особливості
Арифметичні			
+ - * /	додавання віднімання множення ділення	a+b a-b a*b a/b	ділення цілих - результат цілий тип, дробова частина відкидається, ділення дійсних - результат дійсний
%	залишок від ділення цілих чисел	a%b	знак залишку співпадає зі знаком чисельника
-	унарний мінус	-a	зміна знаку числа
Логічні операції			
&& !	і (множення) або (складання) не (заперечення)	a&&b a b !a	результат типу int, рівний 0 - брехня, 1 - істина таблиця істинності a b && !a 1 0 0 1 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 1
Відношення (порівняння)			
== != > < >= <=	дорівнює не дорівнює більше менше не менше не більше	a==b a!=b a>b a<b a>=b a<=b	результат типу int зі значенням 0 при невиконанні умови порівняння інакше 1
Бітові операції			
& ^ ~	і (кон'юнкція) або (диз'юнкція) виключає або заперечення бітів	7&5=5 7 5=7 7^5=2 ~6=1	таблиця істинності 1+1=1 0+1=0 0+0=0 1+0=0 таблиця істинності 1+1=1 1+0=1 0+1=1 0+0=0 таблиця істинності 1+1=0 1+0=1 0+1=1 0+0=0 110 => 001
Операції зсуву			
« »	зсув ліворуч на задане число бітів зсув праворуч на задане число бітів	a«b a»b	зсунути у числі a біти ліворуч на b позицій, результат того ж типу, що і перший аргумент. b не може бути негативним або більше, ніж число бітів у лівому аргументі. Такий зсув еквівалентний множенню значення a операнда на 2 ^{Δb} зсунути у числі a біти праворуч на b позицій, результат того ж типу, що і перший аргумент. b не може бути негативним або більше, ніж число бітів у лівому аргументі
Операції присвоєння			
= +=	 присвоєння суми	a=b a+=b	Перетворення даних до одного типу відбувається автоматично a=a+b Короткі числа -> у довгі із збереженням значення, довгі -> у короткі із втратою старших бітів, float -> у int з відкиданням

			дробової частини
-=	присвоєння різниці	a-=b	a=a-b
*_	присвоєння добутку	a=a*b	a=a*b
/-	присвоєння частки	a/=b	a=a/b
%=	присвоєння залишку від ділення	a%=b	a=a%b
«=	присвоєння лівого зсуву	a«=b	
»=	присвоєння правого зсуву	a»=b	
&=	присвоєння бітової кон'юнкції	a&=b або a&b	
=	присвоєння диз'юнкції	a =b	
^=	присвоєння виключаючого або	a^=b	
Унарні операції			
++	префіксне прирощення	++a або a=a+1	Префіксна форма: спочатку збільшити (зменшити), потім використати b=2*++a => a=a+1 b=2*a
--	префіксне зменшення	--a або a=a-1	
++	постфіксне прирощення	a++	Постфіксна форма: спочатку використовують старе значення, потім його змінюють
--	постфіксне зменшення	a--	b=2*a++ => b=2*a a=a+1

Деякі функції вводу-виводу

Функція	Формат виведення	Призначення	Приклад	Файл з об'явою	Результат
printf()	printf(формат, ap1, ap2,...)	Виведення у стандартний вихідний файл значень аргументів згідно з форматом виведення	printf("%d", c);	stdio.h	значення типу int, що дорівнює кількості виведених символів
scanf()	scanf(формат, ap1, ap2,...)	Введення із стандартного вхідного файлу значень змінних згідно з форматом і присвоєння значень аргументам. Аргументами є адреси змінних	char NAM[20]; int a; scanf("%s", NAM); scanf("%d", &a);	stdio.h	значення типу int, що дорівнює кількості змінних, які отримали значення

getch()	getch(void)	Введення символу з консолі без відображення на екрані	int c; c=getch();	conio. h	тип int, що дорівнює коду введеного символу
-------------	-------------	---	----------------------	-------------	---

Приклад

Обчислимо площу основних геометричних фігур, таких як коло, трикутник, трапеція, квадрат, прямокутник.

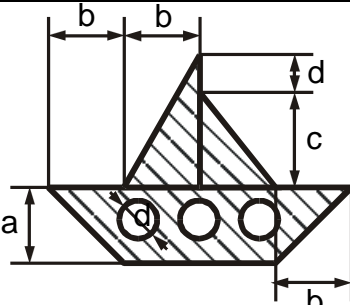
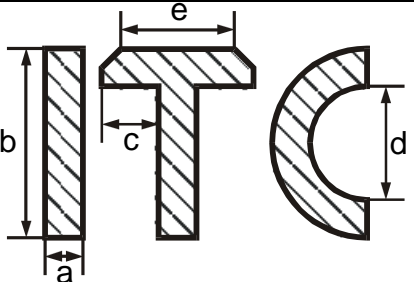
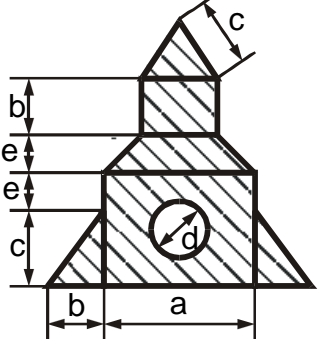
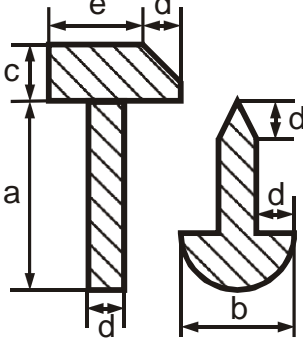
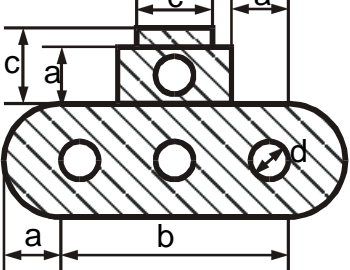
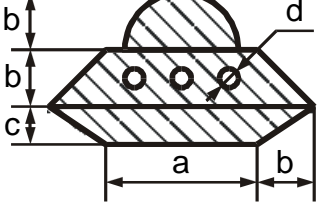
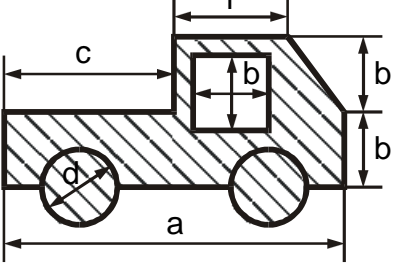
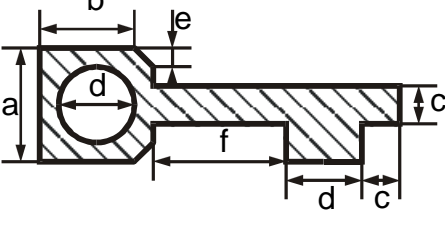
```
#include <stdio.h> // стандартный ввод-вывод
#include <conio.h> // очистка экрана, пауза
#include <math.h> // для математических функций
int main(){
    double d,h,a,s;
    clrscr();
    printf("Введите диаметр круга(основание трапеции): ");
    scanf("%lf",&d);
    printf("Введите высоту прямоугольника(треугольника, трапеции): ");
    scanf("%lf",&h);
    printf("Введите основание прямоугольника(треугольника, квадрата, трапеции): ");
    scanf("%lf",&a);
    s=M_PI*d*d/4.;
    printf("Площадь круга диаметра %.2lf = %lf\n", d,s);
    printf(" она равняется площади круга "
           "радиуса %.2lf = %lf\n", d/2.,M_PI*pow(d/2.,2));
    printf("Площадь квадрата %.2lf x %.2lf = %lf\n", a,a, a*a);
    printf("Площадь прямоугольника %.2lf x %.2lf = %lf\n", a,h, a*h);
    printf("Площадь трапеции с основаниями %.2lf и %.2lf "
           "и высотой %.2lf = %lf\n", a, d, h, (a+d)*h/2.);
    printf("Площадь треугольника с основанием %.2lf и "
           "высотой %.2lf = %lf\n", a, h, a*h/2.);
    printf("Площадь равностороннего треугольника со стороной %.2lf"
           " = %lf\n", a, a*a*sqrt(3.)/4.);
    s=(a+d+h)/2.;
    printf("Площадь треугольника со сторонами %.2lf, "
           "%.2lf, %.2lf = %lf\n", a, h,d , sqrt(s*(s-a)*(s-d)*(s-h)));
    getch();
    return 0;
}
```

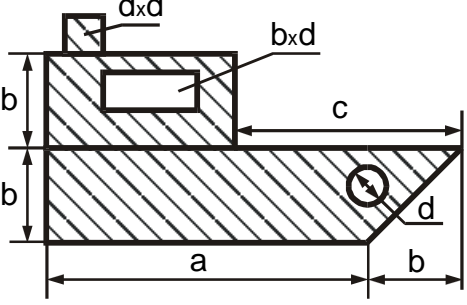
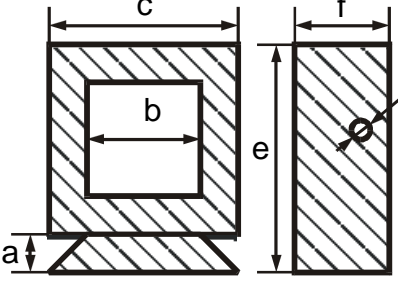
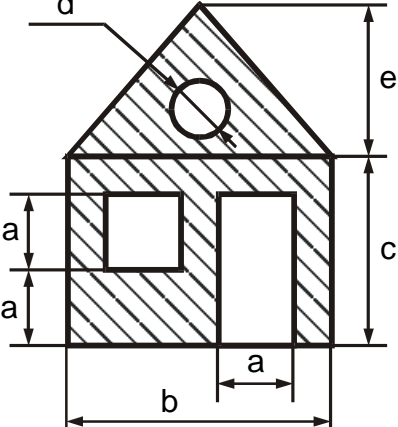
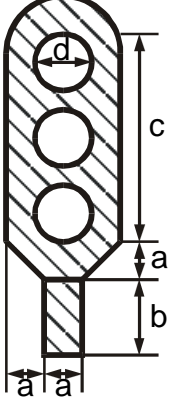
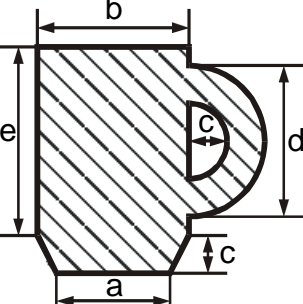
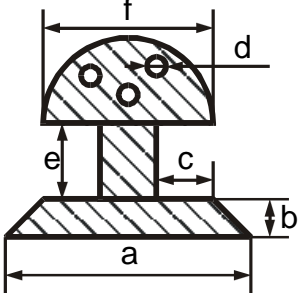
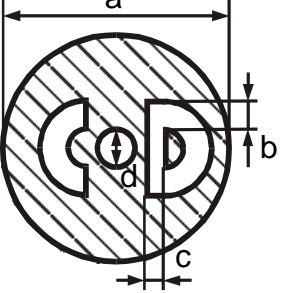
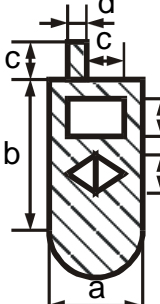
Результати:

Введите диаметр круга(основание трапеции): 2
Введите высоту прямоугольника(треугольника, трапеции): 3
Введите основание прямоугольника(треугольника, квадрата, трапеции): 4
Площадь круга диаметра 2.00 = 3.141593

она равняется площади круга радиуса 1.00 = 3.141593
 Площадь квадрата 4.00 x 4.00 = 16.000000
 Площадь прямоугольника 4.00 x 3.00 = 12.000000
 Площадь трапеции с основаниями 4.00 и 2.00 и высотой 3.00 = 9.000000
 Площадь треугольника с основанием 4.00 и высотой 3.00 = 6.000000
 Площадь равностороннего треугольника со стороной 4.00 = 6.928203
 Площадь треугольника со сторонами 4.00, 3.00, 2.00 = 2.904738

Індивідуальні завдання до ЛР

№	Завдання	№	Завдання
1	 <p>, де $a=4$; $b=4$; $c=5$; $d=2$;</p>	2	 <p>, де $a=2$; $b=10$; $c=3$; $d=6$; $e=6$</p>
3	 <p>, де $a=8$; $b=3$; $c=4$; $d=3$; $e=2$</p>	4	 <p>, де $a=10$; $b=6$; $c=3$; $d=2$; $e=5$</p>
5	 <p>, де $a=3$; $b=12$; $c=4$; $d=2$</p>	6	 <p>, де $a=8$; $b=3$; $c=2$; $d=1$</p>
7	 <p>, де $a=18$; $b=4$; $c=9$; $d=4$; $f=6$</p>	8	 <p>, де $a=6$; $b=5$; $c=2$; $d=4$; $e=1$; $f=7$</p>

9	 <p>де $a=17$; $b=5$; $c=12$; $d=2$</p>	10	 <p>, де $a=2$; $b=6$; $c=10$; $d=1$; $e=12$; $f=5$</p>
11	 <p>, де $a=4$; $b=14$; $c=10$; $d=3$; $e=8$</p>	12	 <p>, де $a=2$; $b=4$; $c=11$; $d=3$</p>
13	 <p>, де $a=6$; $b=8$; $c=2$; $d=8$; $e=10$</p>	14	 <p>, де $a=13$; $b=2$; $c=3$; $d=1$; $e=4$; $f=9$</p>
15	 <p>, де $a=12$; $b=1.5$; $c=1$; $d=2$</p>	16	 <p>, де $a=5$; $b=8$; $c=2$; $d=1$</p>

17	<p>, де $a=2$; $b=3$; $c=1$; $d=8$</p>	18	<p>, де $a=6$; $b=3$; $c=2$; $d=2$</p>
19	<p>, де $a=16$; $b=10$; $c=2$; $d=4$; $e=2$; $f=3$</p>	20	<p>, де $a=9$; $b=3$; $c=5$; $d=1$</p>

Лабораторна робота №4

ОБЧИСЛЕННЯ АРИФМЕТИЧНИХ ВИРАЗІВ

Мета. Вивчити основні арифметичні операції і функції математичної бібліотеки С. Навчитися їх застосовувати для обчислення арифметичних виразів.

Завдання

Робота складається з двох типів завдань, в кожному з яких необхідно обчислити значення арифметичного виразу.

У першому завданні усі аргументи задані у вигляді арифметичних виразів - тобто приклади можуть бути обчислені безпосередньо (без введення даних).

У другому типі завдання використовуються змінні, тому в процесі обчислень необхідно запитати і ввести конкретні значення змінних з клавіатури. Слід звернути увагу на те, що значення змінних повинні належати області допустимих значень. В цій лабораторній роботі приналежність ОДЗ необхідно проконтролювати "теоретично", а бажаючи можуть включити комп'ютерну перевірку даних, що вводяться, на приналежність ОДЗ. Введення і висновок даних повинні супроводитися запрошеннями-коментарями.

Теоретичні відомості

Таблиця стандартних математичних функцій. (файл з прототипами math.h)

функція (прототип)	Короткий опис дії
<code>int abs(int i)</code>	Повертає абсолютне значення цілого аргументу i .
<code>long labs(long x)</code>	Повертає абсолютне значення цілого аргументу <code>long x</code> .
<code>double fabs(double x)</code>	Повертає абсолютне значення дійсного аргументу x подвійної точності.
<code>double fmod(double x, y)</code>	Повертає залишок від розподілу x на y . $fmod(13.657, 2.333)=1.982$;

double y)	
double sin (double x)	Функція синусу. Кут (аргумент) задається у радіанах.
double cos (double x)	Функція косинусу. Кут (аргумент) задається у радіанах.
double tan (double x)	Функція тангенсу. Кут (аргумент) задається у радіанах.
double asin (double x)	Функція арксинусу. Значення аргументу повинне знаходитися у діапазоні від -1 до +1 .
double acos (double x)	Функція арккосинусу. Значення аргументу повинне знаходитися в діапазоні від-1 до +1.
double atan (double x)	Функція арктангенсу.
double sinh (double x)	Повертає значення гіперболічного синусу для x.
double cosh (double x)	Повертає значення гіперболічного косинусу x.
double tanh (double x)	Повертає значення гіперболічного тангенсу для x.
double exp (double x)	Обчислює значення e ^x (експоненціальна функція).
double log (double x)	Повертає значення натурального логарифму (ln x).
double log10 (double x)	Повертає значення десяткового логарифму (lg x).
double pow (double x, double y)	Повертає значення x ^y , тобто x у ступені y.
double sqrt (double x)	Повертає додатне значення квадратного кореня x.
double floor (double x)	Знаходить найбільше ціле, не перевищуюче значення x. floor (9.2)=9; floor (-9.8)=-10;
double ceil (double x)	Округляє до найближчого цілого не меншого x. ceil(9.2)=10; ceil(-9.8)=-9;
M_PI	Число Пі
M_E	Число E

Обчислимо декілька арифметичних виразів.

Приклад першого типу:

$$\frac{\cos\left(\frac{e}{3}\right)^2 + \sin^2 \frac{\pi}{6} + \sqrt[5]{245.6}}{\sqrt{|\lg \pi - \ln \pi|} + 3.2e^{2.1}}$$

Лістинг програми:

```
#include <stdio.h> // для ф-й вводу/вивода
#include <conio.h> // для очистки екрана и паузы
#include <math.h> // для математических функций, Пи и E
int main ()
{
    double x;
    clrscr(); // очистка экрана
```

```

x =
(cos(M_E*M_E/3./3.)+pow(sin(M_PI/6.),2)+pow(245.6,1./5.))/
(sqrt( fabs(log10(M_PI)-log(M_PI)))+3.2*exp(2.1));

printf("(cos(M_E*M_E/3./3.)+pow(sin(M_PI/6.),2)+pow(245.6,1./5.
.))\n");
printf("-----
----- = %lf\n",x)
printf("      (sqrt( fabs(log10(M_PI)-
log(M_PI)))+3.2*exp(2.1))\n");
getch();          // пауза
return 0;
}

```

Результат виконання програми:

```

(cos(M_E*M_E/3./3.)+pow(sin(M_PI/6.),2)+pow(245.6,1./5.))
----- =
0.146191
(sqrt( fabs(log10(M_PI)-log(M_PI)))+3.2*exp(2.1))

```

Приклад другого типу:

$$\frac{8 \arccos\left(\frac{a}{2}\right)}{\sqrt{15-b+c}}$$

Лістинг програми:

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
int main ()
{
double x,a,b,c;
clrscr();
printf("Введіть через пробіл (табуляцію, ввід) значення
аргументов - a, b, c ");
scanf("%lf %lf %lf", &a, &b, &c); // ввід даних - scanf,
%lf - типа double
x= 8.*acos(a/2.)/(sqrt(15.-b)+c);
printf("Для a = %lf b = %lf c= %lf\n", a, b, c);
printf("8.*acos(a/2.)/(sqrt(15.-b)+c) = %lf", x);
getch();
return 0;
}

```

Запустивши програму на виконання, вводимо вхідні дані, що попередньо нами проаналізовані:

- аргументом $\arccos()$ повинно бути значення, що не перевершує за модулем 1:

$$\frac{a}{2} \leq |1| \Rightarrow -1 \leq \frac{a}{2} \leq 1 \Rightarrow -2 \leq a \leq 2;$$

- Підкореневим виразом повинен бути додатній вираз, тобто $15 - b \geq 0 \Rightarrow b \leq 15$;

- Знаменник виразу не повинен дорівнювати нулю $c \neq -\sqrt{15 - b}$

Враховуючи вищевикладене, вводимо, наприклад: a=1,4; b=6; c=4.

Результат виконання програми:

Введіть через пробіл (табуляцію, ввід) значення аргументів - а, b, с 1.5 6. 4.

Для а = 1.500000 b = 6.000000 с = 4.000000

8.*acos(a/2.)/(sqrt(15.-b)+c) = 0.825982

Завдання, що робляться у аудиторії:

$$\text{а) } \frac{1,8e^{-2.5} + \sqrt{2 + \cos^2\left(\frac{\pi}{3}\right)}}{\ln|e - 3.2^5|}; \quad \text{б) } \frac{12.7 \lg 12.8 + \operatorname{arctg}\left(\frac{1}{e}\right) + \sin\left(\frac{\pi}{6}\right)^2}{e^{1.1\pi} + \sqrt[7]{1 + \cos^2\left(\frac{\pi}{3}\right)}}$$

результат: 0.283723 ; результат: 0.448878

Індивідуальні завдання до ЛР
Вирази першого типу

№	Арифметичний вираз	№	Арифметичний вираз
1	$\frac{7,6e^{-2.5} + \ln\left(1 + 2 \sin \frac{\pi}{4}\right)}{\sqrt{\cos^2 \frac{\pi}{8} + 24,3}}$ результат: 0.300124	2	$\frac{5,2 \log_{10} 24,5 + 18,3 \sin^2 \frac{\pi}{3}}{\sqrt{1 + e^{5,53}}}$ результат: 1.316657
3	$\frac{2,6e^{-1.5} + \sqrt{3 \lg 8 + 2\pi}}{8,57 \operatorname{tg} \frac{\pi}{8} - \sin^2 26,7}$ результат: 1.403580	4	$\frac{\sqrt[5]{3 \ln 11,7} \cdot e^{-1,8}}{4 \cos\left(\frac{\pi}{8}\right)^2 + 12,3}$ результат: 0.015169
5	$\frac{7,76 \lg 13,8 + \sqrt{1 + \operatorname{tg}^2 \frac{\pi}{4}}}{\cos\left(\frac{\pi}{6}\right) - \sqrt[3]{2,3e}}$ результат: -10.509900	6	$\frac{\sqrt{25 \left \operatorname{tg} \frac{\pi}{4} - e^{4,2} \right }}{12 \cdot \sqrt[3]{\sin \frac{\pi}{8}}}$ результат: 0.741087
7	$\frac{2,83 \cdot \sin \frac{\pi}{8} + \ln(57,4) \sqrt[3]{28,8}}{2,5 \sqrt{1 + \cos^2 \frac{\pi}{4}}}$ результат: 4.408214	8	$\frac{\operatorname{arctg} \frac{\pi^2}{4} \cdot \sqrt{3,12 + e^\pi}}{2 \ln \sqrt{62,8}}$ результат: 1.467736
9	$\frac{\sin \frac{\pi}{8} + \sqrt[3]{1 + \cos \frac{\pi}{4}}}{e^{-2,5} + \sqrt{ 2,5 - 3 \ln 6,8 }}$ результат: 0.837011	10	$\frac{5 \lg \sqrt{56,3 + \sqrt{198,15}}}{(3,57 + 5 \cos \frac{\pi}{3}) \operatorname{tg}\left(\frac{\pi}{4}\right)}$ результат: 0.760885
11	$\frac{2,67 \ln(5,8 + e^{-3,6}) + \sqrt[3]{23,8}}{\sqrt{17,56} + 2 \sin \frac{\pi}{3}}$ результат: 1.280282	12	$\frac{12,84 + \cos^2(\lg 18 - 3)}{4,25 - \frac{1,8\pi}{(2,13 - e^{3,2})^2}}$ результат: 3.036273

13	$\frac{5,74e^{-1,8} - \sqrt{1 + 2 \sin^2 \frac{\pi}{3}}}{\sqrt[3]{\ln 4,5 + \cos(8,6\pi)}}$ результат: 0.595858	14	$\frac{7 \ln(\sqrt{e^{5,2+\pi}} + \pi^3)}{3 \sin^2\left(\operatorname{arctg} \frac{1}{0,15}\right)}$ результат: 10.884133
15	$\frac{\lg \sqrt{4,2\pi} - 15,87\pi^{3,7}}{2\left(\sin \frac{\pi}{4} + 1,3\right)^4}$ результат: 6.092506	16	$\frac{\sqrt[3]{ \sin 5,25 - 7,8 + \cos \frac{\pi}{6} - e^{4,2} }}{\left(\frac{\pi}{8} + \sqrt{9,13}\right)}$ результат: 1.232286
17	$\frac{\sqrt{8,25 + 2 \cos^2 \frac{\pi}{8}} + e^{2,7}}{\sqrt[3]{2,6 + 5 \ln \frac{\pi}{3}}}$ результат: 12.749580	18	$\pi + \frac{4 \cos^2(e + 7,14\pi)}{e^{3,28} + \sqrt{\frac{\pi^3}{(e + \frac{\pi^2}{e})}}}$ результат: 3.280513
19	$\frac{4,5(\sin^2 \frac{\pi}{3} + \sqrt[3]{\cos \frac{\pi}{5}})}{\lg 11,7 + \sqrt{2,48} e^{\frac{\pi}{2}}}$ результат: 1.673549	20	$\frac{e^{e-\pi} - \sqrt[5]{\lg \sqrt{\pi} + \ln \sqrt{e}}}{0,42 \operatorname{arctg}(-15,36)}$ результат: 0.724861
21	$\frac{ 2,48 - \ln 81,4 }{2,5\left(\cos^3 \frac{\pi}{3} + \sqrt{e^{2,5} + 4\pi}\right)}$ результат: 0.150544	22	$\frac{4\pi^{2,51} + e^{4 \sin 18,2}}{\sqrt[5]{e + \arcsin 0,837}}$ результат: 54.522484
23	$\frac{4,57 \ln\left(\sin \frac{\pi}{4} + 3 \cos^2 \frac{\pi}{8}\right)}{1 + 2\sqrt{\lg 3 + e^{-1,6}}}$ результат: 2.045573	24	$\frac{\sqrt{9,2\left(\sqrt[3]{\pi} + e^{2\pi+1,2}\right)}}{4 \arccos 0,734 + 3 \ln \pi}$ результат: 19.927369
25	$\frac{8,6\left(e^{-1,8} + \sqrt{2,8 + \sin \frac{\pi}{3}}\right)}{\lg 24 - \cos\left(\frac{\pi}{4}\right)}$ результат: 26.575187	26	$\frac{\pi + \frac{e^5}{\pi^3 + 5,1}}{\left \sin 3,33 - \frac{1}{8} \operatorname{arctg} 1,48\right }$ результат: 23.441525
27	$\frac{2,5e^{3,1} - 15,6 \cos^2 \frac{\pi}{3}}{\sqrt{1 + 3,7 \ln 7,4}}$ результат: 17.796124	28	$\frac{\ln^2 \pi + 2,5 \operatorname{tg} 4e}{\sqrt{\sin^2 \frac{\pi}{6} + \cos^2 \frac{\pi}{4}}}$ результат: 24.970707
29	$\frac{12,3 + 6,8e^{-1,4} - 1,5 \sin \frac{\pi}{3}}{2 \operatorname{tg} \frac{\pi}{4} + \sqrt{2 \ln 6}}$ результат: 3.256553	30.	$\frac{9,8e^{\sqrt{2\pi + \cos 2,4}}}{\sqrt[3]{ \pi - e^{3,2} } + 5,6 \lg 12,7}$ результат: 11.529059

Вирази другого типу

1	$\frac{2c - d + \sqrt{23}}{\frac{a}{4} - 1}$	2	$\frac{c + 4d - \sqrt{123}}{1 - \frac{a}{2}}$
3	$\frac{-2c + d82}{\operatorname{tg}\left(\frac{a}{4} - 1\right)}$	4	$\frac{\lg 2c + d - 52}{\frac{a}{4} + 1}$
5	$\frac{\operatorname{arctg} \frac{c}{4} - d + 62}{a^2 - 1}$	6	$\frac{-2c - \ln d + 53}{\frac{a}{4} - 1}$
7	$\frac{2c - \lg \frac{d}{4}}{a^2 - 1}$	8	$\frac{\operatorname{tg} c - 23d}{2a - 1}$
9	$\frac{2c - \frac{d}{23}}{\ln\left(1 - \frac{a}{4}\right)}$	10	$\frac{4c + d - 1}{c - \operatorname{tg} \frac{a}{2}}$
11	$\frac{2c - d\sqrt{42}}{c + a - 1}$	12	$\frac{\sqrt{\frac{25}{c}} - d + 2}{d + a^2 - 1}$
13	$\frac{\operatorname{arctg}\left(c - \frac{d}{2}\right)}{2a - 1}$	14	$\frac{4\lg c - \frac{d}{2} + 23}{a^2 - 1}$
15	$\frac{c \cdot \operatorname{tg}(b + 23)}{\frac{a}{2} - 4d - 1}$	16	$\frac{\frac{c}{d} + \ln(3\frac{a}{2})}{c - a + 1}$
17	$\frac{2c + 51\lg d}{d - a - 1}$	18	$\frac{2c + \ln\left(\frac{d}{4}\right) + 23}{a^2 - 1}$
19	$\frac{42c - \frac{d}{2} + 1}{a^2 - \ln(b - 5)}$	20	$\frac{\operatorname{arctg}\left(\frac{2c}{d}\right) + 2}{d - a^2 - 1}$
21	$\frac{\operatorname{arctg}\left(\frac{12}{c}\right) + 73}{a^2 - 1}$	22	$\frac{\frac{2c}{a} - d^2}{d + \operatorname{tg}(a - 1)}$
23	$\frac{\sqrt{\frac{53}{a} + d} - 4a}{1 + ab}$	24	$\frac{\sqrt{15a + b} - \frac{a}{4}}{ba - 1}$
25	$\frac{-\frac{25}{a} + c - \operatorname{tg} b}{1 + \frac{cb}{2}}$	26	$\frac{\lg(4a - 1) + \frac{b}{2}}{bc - 5}$
27	$\frac{8\lg(b + 1) - c}{\frac{a}{2} + bc}$	28	$\frac{4a - \ln(b - 1)}{\frac{c}{b} + 18a}$
29	$\frac{3\arcsin(c + 1) + b}{\sqrt{42 - a}}$	30	$\frac{\frac{12}{a} + 5b}{\ln d + 4a}$

Лабораторна робота №5 ОРГАНІЗАЦІЯ РОЗГАЛУЖЕНЬ

Мета: Навчитися використовувати умовний оператор для організації розгалужених процесів.

Завдання

Розібрати приклади, та модифікувати їх, згідно зі завданням. Виконати два індивідуальних завдання з обчислення значення шматково-безперервної функції, згідно зі своїм варіантом.

Теоретичні відомості

Умовний оператор має дві форми:

if (вираз_умова) оператор_1; else оператор_2; – повна
if (вираз_умова) оператор_1; – скорочена

у разі істинності (тобто при ненульовому значенні) виразу-умови виконується тільки **оператор_1**, при нульовому значенні виразу-умови виконується тільки **оператор_2**, вираз_умова уявляє собою: арифметичний вираз, відношення або логічний вираз. **оператор_1** і **оператор_2** у операторі **if** можуть бути як окремими, так і складовими операторами.

Завдання для роботи у аудиторії.

1. Скласти блок-схему обчислення коренів квадратного рівняння

$$ax^2 + bx + c = 0:$$

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
int main()
{ double a,b,c,d,q,p,x1,x2;
  clrscr();
  printf("Введіть a,b,c ");
  scanf("%lf%lf%lf",&a,&b,&c);
  d = b*b-4.0*a*c;
  p = -b / (2. * a);
  q = sqrt(fabs(d))/(2.*a);
  if (d >= 0.0)
  { x1=p+q;
    x2=p-q;
    printf("\nx1 = %lf\tx2 = %lf\n",x1,x2);}
  else
    printf("КОРНЕЙ НЕТ\n");
  getch();
  return 0; }
```

2. Ввести програму в пам'ять і отримати рішення на екрані для початкових даних:

а) $a = 2; b = 6; c = 3;$ б) $a = 2; b = 0,8; c = 1,6.$

3. Скласти програму для знаходження загального рішення квадратного рівняння.

Виведення рішення на екран передбачити в наступних формах:

а) ДІЙСНІ КОРЕНІ:

$$X1 = -.633975$$

$$X2 = -2.36608$$

б) КОМПЛЕКСНІ КОРЕНІ

$$X1 = -.2 + .87178*I \quad X2 = -.2 - .87178*I$$

Ввести програму в пам'ять і отримати рішення на екрані для вище приведених даних.

Приклад обчислення шматково-безперервної функції:

$$z = \begin{cases} e^x y - b & y > 6 \\ x^2 y^2 \sin a & y \leq 6 \end{cases}$$

де $x = \sin b^a$, $y = a \operatorname{tg} x + b$; $a=2,2$; $b=10,4$

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
int main(){
    double a=2.2, b=10.4,x,y,z;
    clrscr();
    x=sin(pow(b,a));
    y=a*tan(x)+b;
    if (y > 6)
        z=exp(x)*y-b;
    else
        z=x*x*y*y*sin(a);
    printf("x = %lf\t y = %lf      z = %lf\n",x,y,z);
    getch();
    return 0; }
```

Результат:

$$x = 0.015601 \quad y = 10.434325 \quad z = 0.198387$$

Індивідуальні завдання до ЛР

Завдання 1:

1	$Z = z = \begin{cases} 3 + 1,5e^{-0,6x} & x < 0 \\ \sqrt[3]{2x^2 + y^2} & x \geq 0 \end{cases}, \text{ де}$ $x = b \sin a, y = \ln b - e^x$ <p>$a=1; b=2,3$ $a=6; b=4,1$</p>	2	$z = \begin{cases} 4b + \lg(1 + x^2) & y < 12 \\ \sin^2(a + y^2) & y \geq 12 \end{cases}, \text{ де}$ $x = e^{0,5+a}; y = \frac{x^2}{8,2b+a}$ <p>$a=2; b=7,4$ $a=4; b=15,3$</p>
3	$z = \begin{cases} \sqrt{2,5y + x^2} & x \leq 1,5 \\ 6,7 \ln(y) & x > 1,5 \end{cases}, \text{ де}$ $x = b \cos a; y = \sqrt{b + 1,7x^2}$ <p>$a=0,15; b=2,3$ $a=3,15; b=7,8$</p>	4	$z = \begin{cases} \sin 5x + \cos y^2 & x < 1,3 \\ \sqrt[5]{7,2x^2 + 12,4y^2} & x \geq 1,3 \end{cases}, \text{ де}$ $x = b \sin a; y = 1,2x + e^a$ <p>$a=0,25; b=2,8$ $a=5,25; b=7,6$</p>
5	$z = \begin{cases} \sqrt{\cos^2 x + 2} & x < 2,5 \\ 3y - \sin 8x^2 & x \geq 2,5 \end{cases}, \text{ де}$ $x = \operatorname{tg} a - ab; y = b \cos(1,5a)$	6	$z = \begin{cases} \sqrt{x} \cos a & y > 1 \\ y + a \lg x & y \leq 1 \end{cases}, \text{ де}$ $x = b + e^{1,2+a}; y = x \sin a$

	a=0,1; b=1,5 a=1,3; b=1,9		a=1,1; b=4,7 a=4,5; b=8,3
7	$z = \begin{cases} \sin x^2 + y^2 & x \leq 1 \\ b\sqrt{x^2 + y^2} & x > 1 \end{cases}, \text{ де}$ $x = b - e^{a/2,4}; y = 1,5x + \lg 2a$ a=1,25; b=4,7 a=4,25; b=8,3	8	$z = \begin{cases} \sqrt[3]{2b + bx^2} & x < 0,4 \\ (a + xy)e^x & x \geq 0,4 \end{cases}, \text{ де}$ $x = 1 - \operatorname{tg}^2 \frac{a}{b}; y = a^2x + b$ a=2,5; b=4,1 a=4,5; b=4,9
9	$z = \begin{cases} \sin \alpha \cdot e^x & y < 2 \\ \sqrt{ x + y^2} & y \geq 2 \end{cases}, \text{ де}$ $x = b^2 \cos a; y = a \sin ba$ a=0,4; b=2,8 a=3,4; b=7,6	10	$z = \begin{cases} (y^2 - x^2) / (xy) & x < y \\ (x^2 - y^2) / e^{xy} & x \geq y \end{cases}, \text{ де}$ $x = b \sin 4a; y = a - x \cos 2a$ a=0,2; b=3,5 a=2,8; b=5,1
11	$z = \begin{cases} \sqrt{x^2 + y^2} e^{2x} & y > 2 \\ \ln(b + x) + y & y \leq 2 \end{cases}, \text{ де}$ $x = a + \operatorname{tg}^2 a; y = b \cos 2a$ a=0,1; b=7,3 a=1,1; b=12,5	12	$z = \begin{cases} x \sin a + y \cos a & y > 2 \\ y + e^{ ax } & y \leq 2 \end{cases}, \text{ де}$ $x = \sqrt{1,5 + a} - b \operatorname{tg} a; y = bx - \ln a$ a=0,2; b=4,9 a=0,8; b=7,6
13	$z = \begin{cases} \sqrt{x^2 + y} & x < 4 \\ (b + xy) & x \geq 4 \end{cases}, \text{ де}$ $x = b \sin a; y = x + \operatorname{arctg} 5a$ a=4,2; b=5,7 a=8,6; b=9,4	14	$z = \begin{cases} \sqrt{xy + a} & y > 0 \\ \sin a^2 + b \cos x^2 & y \leq 0 \end{cases}, \text{ де}$ $x = e^{0,5a}; y = b - ax$ a=1,1; b=3,4 a=3,6; b=7,7
15	$z = \begin{cases} \sqrt{x + by^2} & x > 0 \\ e^{-x} + y^2 & x \leq 0 \end{cases}, \text{ де}$ $x = b \ln a; y = bx + \sin a$ a=0,5; b=1,3 a=1,5; b=7,4	16	$z = \begin{cases} 5x^2 + 1,8y^2 & y \geq 0 \\ (1,7 + by)e^x & y < 0 \end{cases}, \text{ де}$ $x = b^2 \cos a; y = 5b - x^2 a$ a=0,2; b=2,5 a=4; b=3,2
17	$z = \begin{cases} 1 + \cos(x + y) & x \leq 2 \\ b \ln y & x > 2 \end{cases}, \text{ де}$ $x = b - 2,5e^a; y = (\sqrt[3]{ x } + b)a$ a=0,1; b=5,5 a=3,1; b=9,3	18	$z = \begin{cases} \sqrt{x^2 + y^2} & y \leq 20 \\ x - y \cos^2 bx & y > 20 \end{cases}, \text{ де}$ $x = b \sin^2 a - b; y = bx^2 - e^a;$ a=0,5; b=1,5 a=4; b=5,4
19	$z = \begin{cases} e^{x-y} & x > 0,1 \\ \frac{x^2}{2} + \frac{y^2}{8} & x \leq 0,1 \end{cases}, \text{ де}$	20	$z = \begin{cases} b^{-x} \cos y & y < 0 \\ \sqrt[3]{bx - y} & y > 0 \end{cases}, \text{ де}$ $x = b \operatorname{tg} a; y = 5 \ln x - b;$

	$x = \sin(b + a^2); y = btg a - a;$ $a=2,5; b=2,5$ $a=6,5; b=8,2$		$a=0,2; b=6,7$ $a=1,2; b=9,1$
21	$z = \begin{cases} \sqrt[4]{y} + b \ln x + 18.2 & x \geq 1.5 \\ \sqrt{y + (x + 1)^2} & x < 1.5 \end{cases},$ де $x = (b^2 - a) \cos ba;$ $y = a e^{x+0.4};$ $a=3; b=4,9$ $a=9; b=9,5$	22	$z = \begin{cases} x + \frac{y}{2} & y < -4 \\ x^2 + \sin y^2 & y > -4 \end{cases},$ де $x = b \cdot e^{1/a}; y = tg a + b \cos x;$ $a=2,2; b=2,6$ $a=3,8; b=6,4$
23	$z = \begin{cases} \ln bx + y & x \geq 0 \\ x + \frac{y}{a} & x < 0 \end{cases},$ де $x = a - b \lg a;$ $y = \sin^2 a + b \cos x;$ $a=4,2; b=5,3$ $a=12,6; b=8,1$	24	$z = \begin{cases} \ln x + tga & y > 20 \\ \sqrt{ x - by } & y \leq 20 \end{cases},$ де $x = b \cdot e^{4.5-a}; y = x - ba^2;$ $a=2,2; b=1,8$ $a=2,8; b=5,4$
25	$z = \begin{cases} y + 2 \cos^2 x & x > 1 \\ x^2 \ln b & x \leq 1 \end{cases},$ де $x = btg a; y = xa - \sin bx;$ $a=3,2; b=1,3$ $a=4,1; b=4,5$	26	$z = \begin{cases} y \sqrt{(x + 2)} & y \leq 1 \\ e^{y-1} & y > 1 \end{cases},$ де $x = b \sin a^2; y = 4.4x + \sqrt{e^a};$ $a=0,2; b=4,3$ $a=1,7; b=6,7$
27	$z = \begin{cases} x^2 - by^2 + e^a & x > y \\ axy & x \leq y \end{cases},$ де $x = a \lg(b + a); y = a^2 x - b \sin x;$ $a=3; b=6,1$ $a=7; b=12,4$	28	$z = \begin{cases} (x^2 - y^2) e^y & x \leq 0 \\ \sin x^y \cdot e^x & x > 0 \end{cases},$ де $x = b - e^{1.2a}; y = \ln xa ;$ $a=0,8; b=6,4$ $a=3,2; b=8,3$
29	$z = \begin{cases} \sqrt{x^2 + y^2} & x < 0 \\ e^x + by^2 & x \geq 0 \end{cases},$ де $x = \arctan ba \cdot \sin ba;$ $y = a^2 - x^2;$ $a=2,3; b=2,14$ $a=6,3; b=4,8$	30	$z = \begin{cases} a \lg y + x^2 & x \leq 2 \\ b(x^2 - y) & x > 2 \end{cases},$ де $x = b \cos ba; y = e^{ x } - a^2;$ $a=3,2; b=5,4$ $a=5,2; b=9,3$

Завдання 2:

Значення а и в (взяти з попереднього завдання) значення а – з першої умови, в – з другої умови.

1	$X = \begin{cases} ab+1, & \text{якщо } a>b, \\ 25, & \text{якщо } a=b, \\ (a-5)/b, & \text{якщо } a<b; \end{cases}$	2	$X = \begin{cases} ab-3, & \text{якщо } a>b, \\ 2, & \text{якщо } a=b, \\ (a^3+1)/b, & \text{якщо } a<b; \end{cases}$
3	$X = \begin{cases} a/b+5, & \text{якщо } a<b, \\ -5, & \text{якщо } a=b, \\ (a^2-b)/b, & \text{якщо } a>b; \end{cases}$	4	$X = \begin{cases} a/b+1, & \text{якщо } a<b, \\ -1, & \text{якщо } a=b, \\ (ab-5)/a, & \text{якщо } a>b; \end{cases}$
5	$X = \begin{cases} a/b-1, & \text{якщо } a>b, \\ -25, & \text{якщо } a=b, \\ (a^3-5)/a, & \text{якщо } a<b; \end{cases}$	6	$X = \begin{cases} ab+21, & \text{якщо } a>b, \\ -5, & \text{якщо } a=b, \\ 3a/b+1, & \text{якщо } a<b; \end{cases}$
7	$X = \begin{cases} 5a+b, & \text{якщо } a>b, \\ -125, & \text{якщо } a=b, \\ (a-5)/b, & \text{якщо } a<b; \end{cases}$	8	$X = \begin{cases} ab-1, & \text{якщо } a>b, \\ 255, & \text{якщо } a=b, \\ (a-5)/b, & \text{якщо } a<b; \end{cases}$
9	$X = \begin{cases} ba+1, & \text{якщо } a>b, \\ -10, & \text{якщо } a=b, \\ (a-5)/b, & \text{якщо } a<b; \end{cases}$	10	$X = \begin{cases} a/b+31, & \text{якщо } a>b, \\ -25, & \text{якщо } a=b, \\ (5a-1)/a, & \text{якщо } a<b; \end{cases}$
11	$X = \begin{cases} 2a+b, & \text{якщо } a>b, \\ -2, & \text{якщо } a=b, \\ (a-5)/b, & \text{якщо } a<b; \end{cases}$	12	$X = \begin{cases} b/a+1, & \text{якщо } a<b, \\ 25, & \text{якщо } a=b, \\ (a^3-5)/b, & \text{якщо } a>b; \end{cases}$
13	$X = \begin{cases} b/a+61, & \text{якщо } a>b, \\ -5, & \text{якщо } a=b, \\ (b-a)/b, & \text{якщо } a<b; \end{cases}$	14	$X = \begin{cases} a/b+1, & \text{якщо } a>b, \\ -2, & \text{якщо } a=b, \\ (a-b)/a, & \text{якщо } a<b; \end{cases}$
15	$X = \begin{cases} (3a-5)/b, & \text{якщо } a<b, \\ -4, & \text{якщо } a=b, \\ (a^3+b)/a, & \text{якщо } a>b; \end{cases}$	16	$X = \begin{cases} b/a-1, & \text{якщо } a<b, \\ -295, & \text{якщо } a=b, \\ (a-235)/b, & \text{якщо } a>b; \end{cases}$
17	$X = \begin{cases} 2a/b+1, & \text{якщо } a>b, \\ -445, & \text{якщо } a=b, \\ (b+5)/a, & \text{якщо } a<b; \end{cases}$	18	$X = \begin{cases} a/b+1, & \text{якщо } a>b, \\ a+25, & \text{якщо } a=b, \\ (ab-2)/a, & \text{якщо } a<b; \end{cases}$
19	$X = \begin{cases} ba+1, & \text{якщо } a>b, \\ 3425, & \text{якщо } a=b, \\ (2a-5)/b, & \text{якщо } a<b; \end{cases}$	20	$X = \begin{cases} a^2-b, & \text{якщо } a>b, \\ -a, & \text{якщо } a=b, \\ (ab-1)/b, & \text{якщо } a<b; \end{cases}$
21	$X = \begin{cases} ab+1, & \text{якщо } a>b, \\ -b, & \text{якщо } a=b, \\ (a-5)/b, & \text{якщо } a<b; \end{cases}$	22	$X = \begin{cases} a/b-1, & \text{якщо } a<b, \\ 25-a, & \text{якщо } a=b, \\ (b-5)/a, & \text{якщо } a>b; \end{cases}$

23	$X = \begin{cases} b/a + 2, & \text{якщо } a > b, \\ -11, & \text{якщо } a = b, \\ (a - 8)/b, & \text{якщо } a < b; \end{cases}$	24	$X = \begin{cases} a/b + 2, & \text{якщо } a > b, \\ 8, & \text{якщо } a = b, \\ (b - 9)/a, & \text{якщо } a < b; \end{cases}$
25	$X = \begin{cases} ab + 5, & \text{якщо } a < b, \\ -5, & \text{якщо } a = b, \\ (b - a)/b, & \text{якщо } a > b; \end{cases}$	26	$X = \begin{cases} a/b + 1, & \text{якщо } a < b, \\ -71, & \text{якщо } a = b, \\ (a - b)/a, & \text{якщо } a > b; \end{cases}$
27	$X = \begin{cases} b/a - 7, & \text{якщо } a > b, \\ 43, & \text{якщо } a = b, \\ (a^3 - b)/b, & \text{якщо } a < b; \end{cases}$	28	$X = \begin{cases} -5 + a/b, & \text{якщо } a > b, \\ 45, & \text{якщо } a = b, \\ (3a - 6)/b, & \text{якщо } a < b; \end{cases}$
29	$X = \begin{cases} a/1,5b + 4,4, & \text{якщо } a > b, \\ -20, & \text{якщо } a = b, \\ (b + 3a)/2, & \text{якщо } a < b; \end{cases}$	30	$X = \begin{cases} 3b/a + 7, & \text{если } a < b, \\ 15, & \text{если } a = b, \\ (a^3 - 2)/b^2, & \text{если } a > b; \end{cases}$

Лабораторна робота №6
ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ.
РОБОТА З ТАБЛИЧНИМИ ДАНИМИ.

Мета: Навчитися використовувати оператори циклів, для обчислень значень функцій за допомогою циклів, та друку результатів у вигляді таблиць.

Завдання

Вивести у вигляді таблиці значення шматково-безперервної функції, функцію взяти з попередньої роботи.

а) у вигляді “вертикальної” таблиці з обрамленням, для значень, від a_n (начало), до a_k (кінець інтервалу), з кроком da ;

b
=...

a	x	y	z
.

б) у вигляді “горизонтальної” таблиці. В залежності від варіанту:

- для перелічених значень аргументу;
- для значень, що отримуються розбиттям початкового інтервалу на завдану кількість рівних відрізків
- для значень, що отримуються розбиттям початкового інтервалу на завдану кількість рівних точок.

b
x
y
z

Теоретичні відомості

У роботі можна використовувати наступні оператори циклу:

- 1) `while` (вираз-умова) тіло_циклу;
- 2) `do`

тіло_циклу
while (вираз-умова);
3) for (ініціювання_циклу; вираз-умова; список_виразів)
 тіло_циклу;

При обчисленні значень функції, початкові дані можуть завдаватися різними способами, наприклад:

- завдання початку інтервалу, кінця інтервалу, кроку розбиття;
- завдання початку інтервалу, кінця інтервалу, кількості точок, у яких треба обчислити значення функції.
- завдання початку інтервалу, кінця інтервалу, кількості відрізків, на які треба розбити інтервал.
- завдання масиву значень аргументу, для яких треба обчислити значення функції.

Наприклад, треба обчислити крок в залежності від кількості точок:

$dx = (x_{END} - x_{BEGIN}) / (n - 1)$ де n - кількість точок, у яких треба обчислити значення функції (кількість елементів масиву); dx - крок; x_{BEGIN} , x_{END} - початок та кінець інтервалу;

Або, навпаки, треба обчислити кількість точок в залежності від кроку:

$n = (x_{END} - x_{BEGIN}) / dx + 1$, де n - кількість точок, у яких треба обчислити значення функції (кількість елементів масиву); dx - крок; x_{BEGIN} , x_{END} - початок та кінець інтервалу;

Якщо значення функції треба буде використовувати далі у програмі, їх необхідно зберігати за допомогою масиву. У такому випадку треба буде обчислити кількість елементів масиву та виділити під нього пам'ять. Оскільки мова C не підтримує масиви зі змінною кількістю елементів, необхідно виділяти пам'ять динамічно, у тому випадку, коли неможливо обчислити розмір, перед визначенням локального масиву.

Визначення локального масиву A з 8 елементів, кожен з яких є дійсне число подвійної точності, та ініціювання його початковими значеннями виглядає так:
double A[8]={2.3,1.,2.2,4.5,6.,-8.0,0,5.9};

Звертання до i -го елемента масиву A : **A[i]**

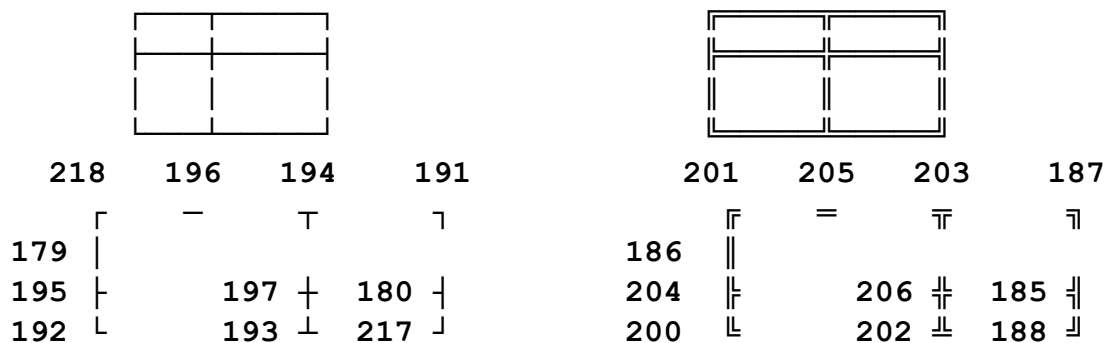
Для динамічного виділення пам'яті використовують функції:

void *malloc(загальний_розмір_блоку_пам'яті); або

void *calloc(кількість_елементів, розмір_одного_елементу);

також **void free(void *показчик_на_блок_пам'яті);** - звільнення пам'яті.

Для того щоб таблиця виглядала більш досконало, можна "промалювати" межі елементів таблиці, застосовуючи псевдографічні символи. Наведемо ASCII - коди деяких з них:



При роботі з псевдографічними символами, зручно зробити літерний масив, що складається з потрібних символів. Для кращої сумісності, наприклад, з Windows, коди символів потрібно задавати не числовим кодом, а літералом, тобто у дужках. Щоб ввести такий символ, потрібно набрати його код на допоміжній-цифровій клавіатурі, натискаючи при цьому клавішу ALT. Приводимо фрагмент програми.

```
unsigned char s[]={201,205,'т', 'т' };
printf(" _____ \n");
printf("%c%c%c%c%c%c%c\n", s[0],s[1],s[1],s[2],s[1], '=', 'т');
for(i=0;i<8;i++) printf("%c", 'т');
```

Приклади:

1) Обчислити значення функції z, розбивши інтервал [2,2;5,2] на 10 рівних відрізків.

$$z = \begin{cases} e^x y - b & y > 6 \\ x^2 y^2 \sin a & y \leq 6 \end{cases}, \text{ де } x = \sin b^a; y = \operatorname{atg} x + b; b=10,4;$$

на інтервалі від [2,2;5,2]; da=0,3

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
int main(){
    double x, y, a, b, an, ak, da, z;
    int n;
    clrscr();
    n=10; an= 2.2; ak= 5.2; b=10.4;
    da= (ak-an)/n;
    for (a=an; a<=ak;a+=da) {
        x=sin(pow(b,a));
        y=a*tan(x)+b;
        if (y >6)
            z=exp(x)*y-b;
        else
            z=x*x*y*y*sin(a);
        printf("x = %lf\t y = %lf\t z = %lf\n",x,y,z);
    }
    getch();
    return 0;
}
```

Результат виконання програми:

```
x = 0.015601    y = 10.434325    z = 0.198387
x = -0.088618   y = 10.177873    z = -1.085263
x = -0.718773   y = 6.112994     z = -7.420833
x = -0.206581   y = 9.310231     z = -2.827428
```

2) Приклад програми, що динамічно виділяє пам'ять під масиви:

```
#include <stdio.h>
#include <stdlib.h> // для calloc, malloc
```

```

int main(){
    double x,y,z,f, *m;
    int k,i;

    k=10; // число элементов массива

    /* Попытка выделения памяти с проверкой.Если неудача -выход
из программы */
    if ((m = (double *) calloc(k,sizeof(double))) == NULL) {
        printf("Неудачная попытка выделения памяти\n");
        exit(1); } /* программа завершена */

    /* вариант обработки элементов массива */
    for (i=0;i<k;i++) m[i]=5*i;

    free (m); /* освобождение участка памяти, выделенного
calloc */
    return 0;
}

```

Індивідуальні завдання до ЛР

№	Завдання	№	Завдання
1	b=2,3; a _N =1; a _K =6; а) крок = 0,5 б) a=1; 1,8; 2,4; 3; 4,1; 5; 6	2	b=7,4; a _N =2; a _K =4; а) крок = 0,2 б) розбити інтервал на 10 відрізків
3	b=2,3; a _N =0,15; a _K =3,15; а) крок = 0,4 б) обчислити у 6 точках	4	b=2,8; a _N =0,25; a _K =5,25; а) крок = 0,5 б) a=0,25; 1; 1,5; 1,7; 2,6; 3,1; 4; 5; 5,25
5	b=1,5; a _N =0,1; a _K =1,3; а) крок = 0,1 б) розбити інтервал на 6 відрізків	6	b=4,7; a _N =1,1; a _K =4,5; а) крок = 0,4 б) обчислити у 11 точках
7	b=8,3; a _N =1,25; a _K =4,25; а) крок = 0,5 б) a=1,25; 1,4; 1,9; 2,3; 2,8; 3,5; 4,25	8	b=4,1; a _N =2,5; a _K =4,5; а) крок = 0,2 б) розбити інтервал на 5 відрізків
9	b=2,8; a _N =0,4; a _K =3,4; а) крок = 0,3 б) обчислити у 7 точках	10	b=3,5; a _N =0,2; a _K =2,8; а) крок = 0,2 б) a=0,2; 1,5; 1,75; 2,2; 2,6; 2,8
11	b=7,4; a _N =0,1; a _K =1,1; а) крок = 0,1 б) розбити інтервал на 5 відрізків	12	b=4,9; a _N =0,2; a _K =0,8; а) крок = 0,05 б) обчислити у 7 точках
13	b=5,7; a _N =4,2; a _K =8,6; а) крок = 0,44 б) a=4,2; 4,7; 5,5; 6; 7,1; 8,6	14	b=3,4; a _N =1,1; a _K =3,6; а) крок = 0,25 б) розбити інтервал на 5 відрізків
15	b=1,3; a _N =0,5; a _K =1,5; а) крок = 0,1 б) обчислити у 6 точках	16	b=2,5; a _N =0,2; a _K =4; а) крок = 0,5 б) a=0,2; 1; 1,3; 2,2; 3,3; 4
17	b=5,5; a _N =0,1; a _K =3,1;	18	b=1,5; a _N =0,5; a _K =4;

	а) крок = 0,3 б) розбити інтервал на 5 відрізків		а) крок = 0,5 б) обчислити у 5 точках
19	b=8,2; a _N =2,5; a _K =6,5; а) крок = 0,4 б) a=2,5; 3; 4; 4,3; 4,9; 6,5	20	b=6,7; a _N =0,2; a _K =1,2; а) крок = 0,125 б) розбити інтервал на 4 відрізків
21	b=4,9; a _N =3; a _K =9; а) крок = 1 б) обчислити у 7 точках	22	b=2,6; a _N =2,2; a _K =3,8; а) крок = 0,25 б) a=2,2; 2,8; 2,9; 3,3; 3,5; 3,8
23	b=5,3; a _N =4,2; a _K =12,6; а) крок = 0,84 б) розбити інтервал на 6 відрізків	24	b=1,8; a _N =2,2; a _K =2,8; а) крок = 0,04 б) обчислити у 6 точках
25	b=1,3; a _N =3,2; a _K =4,1; а) крок = 0,09 б) a=3,2; 3,3; 3,5; 3,75; 4; 4,1	26	b=4,3; a _N =0,2; a _K =1,7; а) крок = 0,15 б) розбити інтервал на 5 відрізків
27	b=6,1; a _N =3; a _K =7; а) крок = 0,5 б) обчислити у 5 точках	28	b=6,4; a _N =0,8; a _K =3,2; а) крок = 0,3 б) a=0,8; 1; 2; 2,6; 2,8; 3,2
29	b=4,8; a _N =2,3; a _K =6,3; а) крок = 0,4 б) розбити інтервал на 5 відрізків	30	b=5,4; a _N =3,2; a _K =5,2; а) крок = 0,2 б) обчислити у 6 точках

Лабораторна робота №7

ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ. РОБОТА З ОДНОВИМІРНИМИ МАСИВАМИ

Мета: Навчитися обробляти одновимірні масиви.

Завдання

Треба виконати індивідуальні завдання 1 і 2 згідно свого варіанту.

Кожна з програм повинна виводити початкові дані масиву(ів), результати обробки або модифікований масив. Якщо масивів декілька і їх значення співвідносяться один до одного, наприклад, координати точок, треба їх виводити на екран попарно(у вигляді таблиці).

Кожна з програм повинна бути універсальною, тобто треба програмувати, наприклад, максимальні значення індексів масивів у вигляді змінних.

Значення елементів масивів необхідно отримати, використовуючи генератор псевдо-випадкових чисел, згідно з умовами завдання.

Теоретичні відомості

Функції отримання псевдо-випадкових чисел.

rand - повертає ціле псевдо-випадкове число у інтервалі від 0 до RAND_MAX (32737). Прототип **int rand(void)** у `stdlib.h`;

random - повертає ціле псевдо-випадкове число в інтервалі від 0 до N-1.

Прототип **int random(int N)** у `stdlib.h`;

randomize - макрос, що ініціалізує генератор випадкових чисел. Потрібен, щоб послідовності не повторювались. Прототип **void randomize(void)** у `stdlib.h`,

викликає `time` з `time.h`. Також генератор можна ініціалізувати викликом наступної функції: `void srand(unsigned seed);`

Приклади використання

ціле випадкове число з інтервалу [0, N-1]	<code>random(N);</code> або <code>rand()%N;</code>
ціле випадкове число з інтервалу [a, b]	<code>a+random(b-a+1);</code>
дійсне випадкове число з інтервалу [0, 1]	<code>(double) rand()/RAND MAX;</code> або <code>1.0*rand()/RAND MAX;</code>
дійсне випадкове число з інтервалу [a, b]	<code>a+(double) rand()/RAND MAX*(b-a);</code>
ініціалізація випадкової послідовності від таймера	<code>randomize();</code> або <code>srand(time(NULL));</code>

Приклад роботи з масивом - вектором

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int main() {
int n=15,a ,b, i, S;
int mas[15]; /* виділення пам'яті під масив * /
/* Можна визначати масив і відразу задати значення
double mas[5]={12, 10, 11, 8, 17};
*/
/* Можна ввести значення з клавіатури
for(i=0; i<N; i++) {
printf("Введіть мас[%d] елемент ", i);
scanf("%d",&mas[i]);
}
*/
/* Заповнюємо масив випадковими цілими числами
в інтервалі [a, b] * /
a= -5; b= 24;
/* randomize();
якщо хочемо, щоб послідовність
випадкових чисел не повторювалася
*/
for(i=0;i<n;i++) mas[i]=a+random(b-a+1);
clrscr(); // очистка екрана
/* Друкуємо значення масиву (вектора)
* /
printf("Исходный массив\n");
for (i=0;i<n;i++) printf("%2d ", mas[i]);
printf("\n");
/* Обчислення суми * /
S=0;
for (i=0;i<n;i++)
S=S+mas[i];
```



```

/ * Обчислення суми. Фрагмент "переписаний на C"
for (S=i=0;i<n;S+=mas[i],i++) ;
*/
printf("сумма = %d\n", S);
getch();
return 0;}

```

Індивідуальні завдання до ЛР

Завдання 1

1	Обчислити добуток елементів масиву С, що належать інтервалу [А,В].
2	Знайти номери елементів масиву М, рівних заданому числу А
3	Обчислити добуток перших семи елементів масиву А.
4	Знайти номери елементів масиву D, значення яких непарне.
5	Знайти мінімальне число серед елементів масиву Е, індекс яких кратний трьом.
6	Підрахувати кількість не додатних елементів масиву В, значення яких по модулю перевищує індекс елемента.
7	Обчислити суму парних елементів масиву F, що перевищують задане число А.
8	Обчислити добуток елементів масиву S, значення яких не менше заданого позитивного числа А, а індекси кратні трьом.
9	Обчислити суму квадратів елементів масиву В, менших заданого числа А
10	Обчислити суму елементів масиву С, значення яких парне і не перевищує задане число А.
11	Обчислити добуток елементів масиву Y, значення яких не перевищує 3*i, де i – індекс елемента.
12	Підрахувати кількість додатних елементів масиву X, значення яких в три рази більше індексу.
13	Порівняти суму і добуток парних за індексом елементів масиву В, виключаючи нульові елементи.
14	Вивести на друк елементи масиву Z, квадрат значення яких перевищує значення індексу в кубі.
15	Обчислити квадрат суми елементів, які більше заданого числа А і індекси яких кратні 3.
16	Обчислити суму всіх елементів масиву V, при обчисленні не додатні значення елементів замінювати їх абсолютним значенням.
17	Обчислити суму індексів елементів масиву W, у яких значення елементів більше А.
18	Обчислити суму творів значень елементів масиву G на значення їх індексів.
19	Знайти середнє геометричне елементів R, значення яких більше А.
20	Знайти середнє арифметичне елементів Q, значення яких менше А.
21	Підрахувати кількість не додатних, додатних, нульових елементів масиву D.
22	Обчислити суму елементів масиву H, що не належать інтервалу [А, В].
23	Обчислити суму останніх десяти елементів масиву В.
24	Знайти номери елементів масиву S, значення яких парне.
25	Знайти максимальне число серед елементів масиву U, значення яких непарне.

26	Обчислити добуток елементів масиву S, значення яких не більше заданого негативного числа A, а індекси непарні.
27	Обчислити суму елементів масиву B, значення яких кратне п'яти і не перевищує задане число A..
28	Обчислити суму елементів масиву M, значення яких перевищує індекс елемента.
29	Обчислити добуток ненульових елементів масиву F, індекси яких непарні.
30	Підрахувати кількість елементів масиву E, рівних нулю, індекси яких парні.

Числові дані до першого індивідуального завдання

№	Тип елементів	Кількість елементів - N	Елементи		Додатково
			від	до	
1	дійсні	12	-4	15	A=0,05; B = 4,4
2	цілі	15	-10	10	A=5
3	дійсні	17	1	7	
4	цілі	17	3	30	
5	дійсні	21	-12	14	
6	дійсні	15	-23	20	
7	цілі	12	2	25	A=8
8	дійсні	24	-18	12	A=3
9	дійсні	18	-10	21	A=5
10	цілі	15	2	18	A=10
11	дійсні	17	3	60	
12	цілі	20	1	70	
13	дійсні	19	-13	11	
14	дійсні	15	7	60	
15	дійсні	16	-5	15	A=9
16	дійсні	17	-20	30	
17	дійсні	20	-10	22	A=7
18	дійсні	18	-8	14	
19	дійсні	14	-12	10	A=5
20	дійсні	19	-15	21	A=11
21	дійсні	25	-50	50	
22	дійсні	17	-20	32	A=-4; B= 12
23	дійсні	23	-14	42	
24	цілі	19	4	45	
25	цілі	15	2	38	
26	дійсні	16	-9	4	A=-4
27	цілі	14	1	90	A=75
28	цілі	18	2	25	
29	дійсні	15	-8	13	
30	цілі	22	-4	4	

Лабораторна робота №8

ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ.

РОБОТА З БАГАТОВИМІРНИМИ МАСИВАМИ

Мета: Навчитися обробляти багатовимірні масиви.

Завдання

Виконати індивідуальні завдання згідно свого варіанту.

Кожна з програм повинна виводити початкову матрицю, результати обробки або модифіковану матрицю.

Кожна з програм повинна бути універсальною, тобто треба програмувати, наприклад, максимальні значення індексів матриці у вигляді змінних.

Значення елементів матриць необхідно отримати використовуючи генератор псевдо-випадкових чисел, згідно з умовами завдання.

Приклад:

```
/* Приклад роботи з масивом - матрицею (динамічне виділення
пам'яті) */
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int main(){
int n, m, a, b, i, j, S;
/* dmas2 - покажчик на покажчик (покажчик на масив). */
int ** dmas2;
// Запитуємо розміри масиву
printf(" n m =? ");
scanf("%d%d",&n,&m);
/* Виділяємо пам'ять для масиву покажчиків. */
dmas2=(int **)calloc(n,sizeof(int *));
/* Виділяємо пам'ять для масивів-рядків (построчно) та їх
початкові адреси запам'ятовуємо в масиві покажчиків. */
for (i=0; i<n;i++)
dmas2[i]=(int *)calloc(m,sizeof( int ));
/* Можна ввести значення з клавіатури
for(i=0; i<n; i++)
for(j=0; j<m; j++) {
printf("Введіть mas2[%d][%d] елемент ", i,j);
scanf("%d",&mas2[i][j]);
} */ /* Заповнюємо масив випадковими цілими числами
в інтервалі [a, b] */
a= -5; b= 24;
/* randomize();
якщо хочемо, щоб послідовність
випадкових чисел не повторювалася */for(i=0; i<n; i++)
for(j=0; j<m; j++) dmas2[i][j]=a+random(b-a+1);
/* Друкуємо значення масиву (матриці)
*/
printf("Исходный массив\n");
for (i=0;i<n;i++) {
for(j=0; j<m; j++)
printf("%2d ",dmas2[i][j]);
printf("\n");
}
/* Обчислення суми */
S=0;
for (i=0;i<n;i++)
for(j=0; j<m; j++)
S=S+dmas2[i][j];
/* Обчислення суми. Фрагмент "переписаний на C"
```

```

for (S=i=0;i<n; i++)
for(j=0; j<m;S+=dmas2[i][j], j++); */ printf("сумма = %d\n", S);
/ Звільнення пам'яті, виділеної для масиву
for (i=0; i<n;i++) free(dmas2[i]);
free(dmas2);
getch();
return 0;
}

```

Індивідуальні завдання до ЛР

1	Розташувати елементи кожного стовпця матриці $A(N, M)$ за збільшенням.
2	Поміняти місцями рядок, що містить максимальний елемент, з рядком, що містить мінімальний елемент матриці $E(N, M)$.
3	Знайти і надрукувати значення і індекси максимального елемента в кожному стовпці матриці $C(N, M)$.
4	Помножити значення кожного елемента рядка матриці $D(N, M)$ на значення останнього елемента відповідного рядка.
5	Розташувати елементи всієї матриці $B(N, M)$ за збільшенням (зліва направо, зверху вниз).
6	Обчислити суму стовпців матриці $F(N, N)$ з позитивними елементами на головній діагоналі.
7	Дана матриця $A(N, M)$. Визначити кількість рядків матриці A , що містять хоча б одну нульову компоненту.
8	Обчислити число непарних елементів в кожному рядку матриці $W(N, M)$
9	Замінити перший рядок матриці $X(N, N)$ максимальним елементом головної діагоналі.
10	Знайти кількість додатних елементів в кожному стовпці матриці $Z(N, M)$.
11	Замінити всі елементи матриці $Y(N, N)$, розташовані вище головної діагоналі, максимальним елементом, розташованим на головній діагоналі.
12	Отримати нову матрицю $C(N, M)$ шляхом віднімання всіх її елементів з максимального елемента матриці $A(N, M)$.
13	В заданій матриці $T(N, M)$, яка містить тільки цілі числа, замінити перший не додатний елемент максимальним елементом матриці. Якщо не додатних елементів немає, то вивести відповідний текст. Елементи матриць проглядати зліва направо і зверху вниз.
14	В заданій матриці $G(N, M)$ знайти індекси першого елемента, що перевершує середнє арифметичне всіх елементів. Елементи матриць проглядати зліва направо і зверху вниз.
15	Із заданої матриці $H(N, M)$ видалити рядок, в якому знаходиться перший не додатний елемент. Елементи матриць проглядати зліва направо і зверху вниз
16	Визначити, скільки рядків заданої матриці $U(N, M)$ містять хоча б один елемент із діапазону $[A, B]$.
17	Поміняти місцями рядок матриці $P(N, N)$, що містить максимальний елемент головної діагоналі, із стовпцем, що містить мінімальний елемент головної діагоналі.
18	Отримати нову матрицю $F(N, M)$ шляхом складання всіх елементів заданої

	матриці $B(N, M)$ з її найменшим за модулем елементом.
19	Розташувати елементи кожного рядка матриці $A(N, M)$ за спаданням.
20	Обчислити суму парних елементів в кожному стовпці матриці $A(N, M)$
21	Знайти і надрукувати значення і індекси мінімального елемента в кожному рядку матриці $W(N, M)$.
22	Збільшити значення кожного елемента стовпця матриці $G(N, M)$ на значення першого елемента відповідного стовпця.
23	Обчислити суму рядків матриці $Z(N, N)$ з нульовими елементами на побічній діагоналі.
24	Розташувати елементи всієї матриці $C(N, M)$ за спаданням (зверху вниз, зліва направо).
25	Замінити останній стовпець матриці $H(N, N)$ мінімальним елементом головної діагоналі.
26	Знайти і надрукувати, скільки не додатних елементів містяться в кожному рядку матриці матриці $F(N, M)$.
27	Замінити мінімальний елемент головної діагоналі матриці $D(N, N)$ сумою елементів першого рядка.
28	Отримати нову матрицю $S(N, M)$ шляхом складання всіх елементів завданої матриці $C(N, M)$ з її найбільшим за модулем елементом.
29	Із заданої матриці $A(N, M)$ видалити рядок і стовпець, в яких знаходиться перший елемент, рівний нулю. Отриману матрицю ущільнити. Елементи матриць проглядати зліва направо і зверху вниз.
30	В заданій матриці $A(N, M)$ знайти рядки, що не містять не додатних елементів.

Числові дані до другого індивідуального завдання

№	Тип елементів	Розміри матриці		Елементи		Додатково
		N-рядки	M-стовпчики	від	до	
1	дійсні	5	6	-9	25	
2	цілі	4	7	-12	30	
3	цілі	6	5	-20	18	
4	дійсні	4	6	-9	12	
5	цілі	7	4	-40	50	
6	цілі	6		-10	24	
7	цілі	5	4	-5	6	
8	цілі	5	7	1	45	
9	цілі	7		-9	37	
10	дійсні	5	6	-14	26	
11	цілі	5		-17	48	
12	цілі	6	5	-15	45	
13	цілі	7	5	-12	23	
14	цілі	6	4	-14	16	
15	цілі	7	6	-20	30	
16	дійсні	4	6	-19	22	A=5; B=12
17	цілі	5		-8	19	
18	цілі	7	6	-15	25	
19	цілі	5	7	-31	12	

20	цілі	7	6	2	51	
21	дійсні	6	5	-13	36	
22	дійсні	7	5	-8	16	
23	цілі	7		-4	7	
24	дійсні	5	4	-54	61	
25	цілі	6		-24	43	
26	цілі	6	7	-10	37	
27	цілі	5		-7	14	
28	цілі	6	4	-18	18	
29	цілі	6	7	-9	9	
30	дійсні	8	5	-12	20	

Лабораторна робота №9 РОБОТА ЗІ СТРУКТУРАМИ

Мета: Навчитися описувати, визначати структури та застосовувати їх для обробки даних.

Завдання

Виконати індивідуальне завдання. Описати структуру, ввести початкові данні та надрукувати їх на екрані, виконати запити. При сортуванні, рядки можна порівнювати один з одним за допомогою функції `int strcmp(const char *s1, const char*s2);` яка повертає: число < 0 , якщо $s1 < s2$; 0 , якщо $s1 == s2$ і число > 0 , якщо $s1 > s2$. (прототип функції знаходиться у заготовочному файлі `string.h`)

Приклад

Опишемо структуру літак: назва – рядок до 10 символів, число пасажирів – ціле число, дальність польоту – ціле число.

Завдамо початкові дані та виведемо їх на екран.

Лістинг програми:

```
#include <stdio.h>
int main() {
    typedef struct { // структура самолет, описание типа
        char name[10]; // марка
        int k_pas; // число пассажиров
        int distance; // дальность полета
    } airplane;
    /* определение структуры air, массива из 4 структур типа
    airplane
    и инициализация начальными данными */
    airplane air[4]={{"Fly-2", 20, 1000},
                    {"Fly-10", 50, 2500},
                    {"Fly-50", 180, 600},
                    {"SFly-100", 380, 10000}};

    int i;
    printf("          марка          вместительность
дальность\n");
    for ( i=0; i<4; i++) {
        printf(" %15s\t%6d\t\t\t%8d",
            air[i].name, air[i].k_pas, air[i].distance);
```

```

    printf("\n"); }
return 0;
}

```

Результати:

марка	вместительность	дальность
Fly-2	20	1000
Fly-10	50	2500
Fly-50	180	600
SFly-100	380	10000

Індивідуальні завдання до ЛР

- Скласти програму, що обробляє наступні дані про студентів: П.І.Б. студента, група, середній бал у сесію. Вивести:
 - списки студентів по групах;
 - середній бал групи.
- Скласти програму, що обробляє наступні дані про склад: код товару; назва товару; ціна за одиницю товару; кількість. Вивести:
 - список усіх товарів за алфавітом;
 - загальну кількість і ціну кожного товару.
- Скласти програму, що обробляє наступні дані про склад: код складу; код деталі; загальна кількість деталей; кількість забракованих.. Вивести:
 - списки деталей на кожному складі;
 - процент браку по кожній деталі.
- Скласти програму, що обробляє наступні дані про робітника: П.І.Б.; шифр (табельний номер); відділ, заробітна плата; термін роботи. Вивести:
 - списки робітників по відділах;
 - порахувати середню заробітну плату на підприємстві.
- Скласти програму, що обробляє наступні дані про кафедру: назва кафедри, факультет, кількість викладачів, кількість лаборантів. Вивести:
 - списки кафедр по факультетам;
 - порахувати кількість викладачів і лаборантів на кафедрі по факультетах.
- Скласти програму, що обробляє наступні дані про гуртожиток : П.І.Б. студенту, назва гуртожитку, номер кімнати. Вивести:
 - списки студентів по гуртожитках;
 - кількість студентів по гуртожитках.
- Скласти програму, що обробляє наступні дані про робітника: П.І.Б робітника, код цеху, норма випуску за планом, кількість фактично виробленої продукції. Вивести:
 - Вивести фактичне виконання плану;
 - кращого робітника;
- Скласти програму, що обробляє наступні дані про ремонтну бригаду: назва бригади; найменування ремонту; кількість запчастин, що потрібні; кількість запчастин, що є у наявності. Вивести:
 - Загальна кількість деталей, що потребує бригада;
 - список бригад, що мають простоювати
- Скласти програму, що обробляє наступні дані про поїзди: шифр вагону; шифр составу; вага вагону; місце призначення. Вивести:
 - списки вагонів за місцем призначення;

- вагу для кожного составу.
10. Скласти програму, що обробляє наступні дані про банки: П.І.Б клієнта, назва банку; назва вкладу; сума вкладу. Вивести:
 - списки клієнтів по банках;
 - загальна кількість коштів, що розміщені у кожному із банків.
 11. Скласти програму, що обробляє наступні дані про пошту: індекс поштового відділення; назва видання; ціна видання; кількість підписаних видань. Вивести:
 - список видань по поштовим відділенням;
 - загальну вартість підписки для поштового відділення.
 12. Скласти програму, що обробляє наступні дані про рекламне агенство: назва видання, назва фірми, що рекламується, вартість реклами. Вивести:
 - список фірм, що рекламуються, для кожного видання;
 - вартість реклами для фірми.
 13. Скласти програму, що обробляє наступні дані про автомобілі: марка авто; колір авто; рік випуску; місце реєстрації П.І.Б. Вивести:
 - список авто за роком випуску;
 - кількість автомобілів однакового кольору.
 14. Скласти програму, що обробляє наступні дані про автомобілі: марка авто; колір авто; рік випуску; пробіг. Вивести:
 - список авто по зростанню пробігу;
 - загальний пробіг авто однієї марки
 15. Скласти програму, що обробляє наступні дані про ресторан: номер столу, назва блюда, ціна блюда, прізвище кухаря. Вивести:
 - списки блюд для кожного столу;
 - загальну вартість блюд, що виготовлені одним кухарем.
 16. Скласти програму, що обробляє наступні дані про школу: П.І.Б учня, назва класу, оцінка. Вивести:
 - список учнів у яких є хоча б одна двійка;
 - кількість відмінних оцінок для кожного класу.
 17. Скласти програму, що обробляє наступні дані про літак: П.І.Б пасажиру, код рейсу, вага пасажиру; вага багажу. Вивести:
 - вагу кожного пасажиру разом з багажем;
 - вагу кожного рейсу
 18. Скласти програму, що обробляє наступні дані про літак: П.І.Б пасажиру, код рейсу, вага багажу, кількість речей. Вивести:
 - список пасажирів для рейсу;
 - середню вагу однієї речі пасажиру.
 19. Скласти програму, що обробляє наступні дані про чемпіонат: прізвище спортсмена, назва команди, кількість балів, кількість виступів. Вивести:
 - список спортсменів для кожної команди;
 - загальну кількість очок, що набрала команда.
 20. Скласти програму, що обробляє наступні дані про: П.І.Б робочого, назва підрозділу, час запізнення. Вивести:
 - списки робочих за підрозділами;
 - загальний час запізнення для кожного підрозділу.
 21. Скласти програму, що обробляє наступні дані про деканат: прізвище викладача, назва предмету, назва групи, кількість не склавших МСК. Вивести:
 - список предметів що веде кожен викладач;
 - кількість невстигаючих студентів для кожної групи.

22. Скласти програму, що обробляє наступні дані про виробництво: П.І.Б робочого, назва підрозділу, стать, сімейний стан. Вивести:
- список одружених виробників;
 - загальна кількість жінок і чоловіків.
23. Скласти програму, що обробляє наступні дані про бригаду: П.І.Б робочого, назва підрозділу, стать, стаж роботи. Вивести:
- список робітників, вік яких менший за 27 років;
 - середній стаж робітників бригади;
24. Скласти програму, що обробляє наступні дані про АТС: прізвище абоненту, назва АТС, тривалість дзвінка, тип дзвінка (міжміський/місцевий). Вивести:
- кількість дзвінків у абонента;
 - загальна тривалість дзвінків по АТС.
25. Скласти програму, що обробляє наступні дані про АТС: прізвище абоненту, назва АТС, тривалість дзвінка, місто, з яким з'єднують. Вивести:
- список міст, у які телефонував абонент;
 - загальна тривалість розмов з містом для кожної АТС.
26. Скласти програму, що обробляє наступні дані про диск: назва файлу, диск, каталог розміщення, розмір файлу. Вивести:
- списки файлів на різних дисках;
 - об'єм кожного каталогу.
27. Скласти програму, що обробляє наступні дані про заробітну плату: П.І.Б робітника, відділ, нарахована заробітна плата, сума вирахувань з робітника. Вивести:
- заробітну плату, що одержить кожен робітник на руки;
 - суму податку, що сплачує кожен відділ.
28. Скласти програму, що обробляє наступні дані про наявність обчислювальної техніки: назва підрозділу, назва відділу, чисельність персоналу, кількість комп'ютерів. Вивести:
- загальну кількість комп'ютерів у підрозділах;
 - список підрозділів, що мають потребу у обчислювальній техніці
29. Скласти програму, що обробляє наступні дані про ДТП: прізвище інспектора, що зафіксував ДТП; прізвище порушника; місце події; сума штрафу. Вивести:
- списки зафіксованих подій по кожному інспектору;
 - загальна сума штрафу для правопорушника.
30. Скласти програму, що обробляє наступні дані про працевлаштування: назва фірми; посада; заробітна плата, досвід роботи. Вивести:
- список вакансій для кожної фірми.
 - список посад, що не потребують досвіду роботи;

ЛІТЕРАТУРА

1. Б. Керниган, Д. Ритчи. Язык программирования Си.: Пер. с англ. - М.: Финансы и статистика, 1992.-272
2. Х. Дейтл, П. Дейтл. Как программировать на С(С++): Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 2000. – 1008с.
3. Б. Страуструп. Язык программирования СИ++, спец изд./Пер. с англ. –М.; СПб.: «Издательство БИНОМ»-«Невский Диалект», 2001. –1099с., ил.
4. С.С++ Программирование на языке высокого уровня./Т.А. Павловская. – СПб:Питер, 2002. –464с;ил.

5. Подбельский В.В., Фомин С.С. Программирование на языке Си: Учеб. пособие.- 2-е доп. изд.- М.: Финансы и статистика, 2001.- 600 с.: ил.
6. Подбельский В.В. Язык Си++: Учеб. пособие.- 2-е изд., перераб. и доп.- М.: Финансы и статистика, 1996.- 560 с.: ил.
7. Стивен Прата. Язык программирования С++. Лекции и упражнения. Учебник: Пер. с англ./Стивен Прата – К.: Издательство «ДиаСофт», 2001. –656с.
8. Borland C++ 3.0. Programmer's Guide. – Scotts Valley, USA: Borland International, Inc. –1991. –467 p.
9. Borland C++ 3.0. User's Guide. – Scotts Valley, USA: Borland International, Inc. – 1991. –229 p.
10. Borland C++ 3.0. Library Reference. – Scotts Valley, USA: Borland International, Inc. –1991. –655 p.

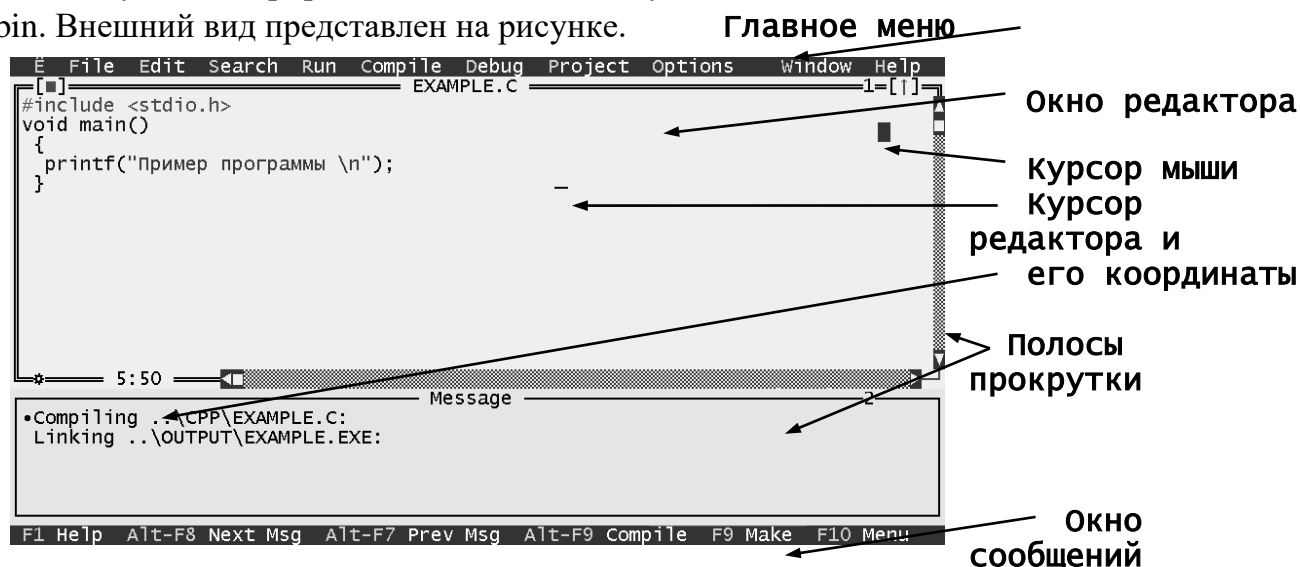
ДОДАТОК 1

ИНТЕГРОВАННЕ СЕРЕДОВИЩЕ BORLAND C/C++ (російською мовою)

ИНТЕГРИРОВАННАЯ СРЕДА BORLAND C/C++

Borland C++ - это профессиональный оптимизирующий компилятор для разработчиков, работающих с языками C++ и Си. Он поддерживает язык C++, соответствующий версии 3.0 стандарта AT&T, и Си стандарта ANSI. Это мощный, быстрый и эффективный компилятор, позволяющий быстро и удобно разрабатывать и отлаживать программное обеспечение.

Запуск интегрированной оболочки осуществляется командой `bc.exe` из каталога `bin`. Внешний вид представлен на рисунке.



Экран системы может состоять из нескольких окон. Обычно это следующие типы окон: окна редактирования, окно сообщений, окна отладки.

Управление системой осуществляется с помощью клавиатуры и мыши, используя меню и специальные комбинации клавиш. Для перехода к главному меню, необходимо

нажать F10 или клавишу Alt в совокупности с выделенной буквой необходимого пункта меню.

Кратко рассмотрим основные пункты меню.

Меню Ё (Системное, Alt + Space).

При вызове этого меню на экран выводится команда Repaint Desktop и имена тех программ, которые были установлены вами с помощью команды, Transfer(пункт Options).

Меню File (Работа с файлами, Alt + F)

Команда New (Новый файл)!

Команда New(пункт File) позволяет открыть новое окно редактирования с именем NONAMExx.C, заданным по умолчанию (вместо xx стоит номер от 00 до 31).

Команда Open (Открыть, F3)

Отображает диалоговое окно для выбора файлов, позволяющее выбирать программный файл, который будет открыт в окне редактирования.

Команда Save (Сохранить, F2)

Записывает на диск файл из активного окна редактирования.

Команда Save As (Сохранить под именем)

Позволяет сохранить файл, находящийся в активном окне редактирования, под другим именем, в другом каталоге или на другом дисковом диске.

Команда Save All (Сохранить все)

Аналогична команде Save, но сохраняет на диске содержимое всех измененных файлов, а не только файла активного окна редактирования.

Команда Change Dir(Сменить каталог)

позволяет задавать идентификатор дискового диска и имя каталога, которые следует сделать текущими.

Команда Print (Печатать)

Позволяет выводить на печать содержимое активного окна редактирования или окна сообщений.

Команда DOS Shell (Выход в оболочку DOS)

Позволяет временно выйти из системы Borland C, чтобы выполнить команду DOS или запустить какую-либо программу. Для того, чтобы возвратиться в систему Borland C, необходимо ввести с клавиатуры EXIT и нажать клавишу Enter.

Команда Exit (Выход, Alt + X)

Выполняет выход из интегрированной интерактивной среды.

Меню Edit (Редактирование, Alt + E)

Меню Edit (Редактирование) позволяет выполнять удаление, копирование и вставку текста в окнах редактирования.

Команда Undo (Отменить изменения, Alt + Backspace)

Восстанавливает файл в текущем окне в тот вид, который он имел перед самым последним действием по редактированию или перемещению курсора

Команда Redo (Отменить отмену, Alt + Shift + Backspace)

Отменяет действие последней команды Undo.

Команда Cut (Вырезать, Shift + Del)

Удаляет из документа выделенный фрагмент текста и заносит его в "карман" (текстовый буфер). Затем этот текст можно извлечь из "кармана" (вклеить) в любой документ, выбрав команду извлечения (вклейки) Paste.

Команда Copy (Копировать, Ctrl + Ins)

Оставляет выделенный текст нетронутым, но заносит в текстовый буфер точную копию этого текста. Затем, выбрав команду вклейки Paste, можно вклеить текст в любой документ.

Команда Paste (Извлечь – Вклеить, Shift + Ins)

Вставляет текст, находящийся в текстовом буфере, в текущее окно по месту курсора.

Команда Clear (Стереть, Ctrl + Del)

Удаляет выбранный фрагмент текста без занесения его в текстовый буфер.

Команда Copy Example (Скопировать пример)

Копирует предварительно выбранный в окне подсказки Help текст примера в текстовый буфер.

Команда Show Clipboard (Показать содержимое текстового буфера)

Открывает окно текстового буфера, в котором хранятся фрагменты текста, вырезанного или скопированного из других окон.

Меню Search (Поиск, Вызов, Alt + S)

Позволяет производить поиск и замену текста, поиск описаний функций, а также ошибок в файлах.

Команда Find (Найти, Ctrl + Q+F)

Отображает диалоговое окно Find Text, позволяющее ввести искомый текст и задать параметры, оказывающие влияние на процесс поиска.

Опции:

Case sensitive - чувствительность к регистру(различать символы верхнего и нижнего регистров).

Whole words only - только целые слова. Поиск только целых слов, то есть искомая строка должна быть с обеих сторон ограничена символами пунктуации или пробелами.

Direction – направление поиска: forward(вперед) или backward(назад), указание направления, в котором необходимо выполнить поиск с места, которое определяется с помощью опции "Начало поиска".

Score - область поиска: указание, в какой части файла нужно вести поиск: во всем файле (Global) или только в выбранном тексте (Selected text).

Origin - начало поиска: From cursor(От курсора) или Entire score(Весь диапазон).

Команда Replace (Заменить, Ctrl + Q + A)

Позволяет производить поиск с заменой, большинство опций идентичны имеющимся в диалоговом окне Find Text.

Text to find (искомый текст), Replace Text (Заменить текст), Prompt on Replace (Подсказка при замене: определяет, будет ли выдаваться запрос на подтверждение каждого изменения).

Для запуска процесса поиска, нужно ввести искомую строку и строку замены и выбрать кнопку ОК или Change All (Заменить все). Выбор командной кнопки Cancel приводит к отмене поиска.

Команда Search Again (Повторить поиск, Ctrl + L или F3)

Повторяет действие последней введенной команды Find или Replace.

Команда Go to Line Number (Перейти к строке с номером)

Запрашивает номер той строки, на которую требуется перейти.

Команда Previous Error (Предыдущая ошибка, Alt + F7)

Перемещает курсор в позицию, вызвавшую предыдущее сообщение об ошибке или предупреждении.

Команда Next Error (Следующая ошибка, Alt + F8)

Перемещает курсор в позицию, вызвавшую следующее сообщение об ошибке или предупреждении.

Команда Locate Function (Определить местоположение функции)

Эта команда доступна только в период проведения сеанса отладки.

Меню Run (Выполнение, Alt +R)

Содержит команды, которые запускают программы, а также иницируют и завершают сеанс отладки.

Команда Run (Выполнение, Ctrl + F9)

Запускает программу, используя аргументы, переданные программе с помощью команды Arguments. Если с момента последней компиляции исходный текст изменялся, то для перекompиляции и перекompоновки программы эта команда вызовет также программу администратора проектов.

Команда Program Reset (Сброс программы, Ctrl + F2)

Прекращает текущий сеанс отладки, освобождает память, занимаемую вашей программой, и закрывает все открытые файлы, которые использовались в вашей программе.

Команда Go to Cursor (Выполнить до курсора, F4)

Выполняет программу от начала (или последней выполненной строки) до строки, в которой установлен курсор.

Команда Trace Into (Выполняет трассирующее вхождение, F7)

Пошагово выполняет программу. Когда при этом достигается вызов какой-либо функции, то вместо выполнения этой функции за один шаг будет выполняться каждый оператор этой функции. Команда останавливается на следующем выполняемом операторе.

Команда Step Over (Одношаговое выполнение функции, F8)

Выполняет следующий оператор в текущей функции. Трассирующие вхождения в вызовы функций более низкого уровня не выполняются даже в том случае, если они доступны отладчику.

Команда Arguments (Аргументы)

Позволяет задавать выполняемой вами программе аргументы командной строки.

Меню Compile (Компиляция, Alt + C)

Используется для компиляции программы, полной или избирательной перекompиляции проекта.

Команда Compile (Компилировать, Alt + F9)

Компилирует файл, находящийся в активном окне редактора.

Команда Make (Сборка, F9)

Вызывает администратор проектов для создания файла EXE или библиотеки. Перекомпилирует только те файлы, которые изменялись с момента последней компиляции.

Команда Link (Компоновка)

Берет текущие файлы, задаваемые либо в файле проекта, либо по умолчанию, и компоует их.

Команда Build (Полная перекомпиляция и перекомпоновка)

Команда аналогична команде Make, но перекомпиляция производится безотносительно к датам исходных файлов.

Команда Information (Получить информацию)

Выводит на экран информацию о текущем файле или проекте.

Команда Remove Messages (Удаляет все сообщения из окна Message).

Меню Debug (Отладка, Alt + D)

Управляют всеми возможностями интегрированного отладчика. Информация о необходимости отладочной информации задается с помощью диалогового окна Options\Debugger

Команда Inspect (Проверка, Alt + F4)

Открывает окно проверки Inspector, позволяющее анализировать и изменять значения элемента данных.

Команда Evaluate/Modify (Вычислить/Изменить, Ctrl + F4)

Вычисляет значение переменной или выражения, отображает их значение и, если возможно, позволяет изменить это значение.

Команда Call Stack (Стек вызовов, Ctrl + F3)

Показывает стек вызовов, отображает последовательность функций, которые вызывались вашей программой перед достижением функции, выполняемой в настоящий момент. У основания стека располагается функция main; у вершины стека находится та функция, которая выполняется в настоящий момент.

Команда Watches (Просмотр выражений)

Открывает всплывающее меню, содержащее команды, которые управляют использованием точек просмотра.

Команда Add Watch (Добавить выражение просмотра, Ctrl + F7)

Команда Delete Watch (Удалить выражение для просмотра)

Команда Edit Watch (Редактировать выражение для просмотра)

Команда Remove All Watches (Удалить все выражения для просмотра)

Команда Toggle Breakpoint (Включить/Выключить точку останова, Ctrl + F8)

Позволяет устанавливать или отменять безусловную точку останова на той строке, в которой установлен курсор. Когда точка останова установлена, она отмечается маркером точки останова.

Команда Breakpoints (Точки останова)

Открывает диалоговое окно, позволяющее управлять использованием как условных, так и безусловных точек останова

Меню Project (Проект, Alt + P)

Содержит команды управления проектом.

Команда Open Project (Открыть проект)

Команда Close Project (Заккрыть проект)

Команда Add Item (Добавить элемент)

Используется, когда вы хотите добавить к списку проекта какой-либо файл.

Команда Delete Item (Удалить элемент)

Команда Local Options (Локальные параметры)

Позволяет включить параметры переопределения командной строки для конкретного модуля, входящего в состав файла проекта.

Команда Include Files (Включаемые файлы)

Отображает список файлов, включенных в проект.

Меню Options (Опции, Alt + O)

Диалоговое окно Application (Установить параметры для прикладной программы)

Команда Compiler (Компилятор)

Задания параметров, оказывающих влияние на процесс компиляции программы

Команда Transfer (Переход к): позволяет добавлять или удалять программы в меню

Команда Make (Сборка или избирательная перекомпиляция): условия работы с проектом.

Команда Linker (Компоновщик): параметры компоновщика.

Команда Librarian (Библиотекарь): параметры встроенного библиотекаря.

Команда Debugger (Отладчик): параметры отладчика интегрированной среды.

Команда Directories (Каталоги)

указывать системе Borland C, где она должна вести поиск файлов, необходимых для компиляции и компоновки, и куда должны направляться двоичные файлы и файлы карты памяти.

*Include Directories задает каталог, в котором содержатся используемые включаемые файлы. (имена которых заданы в угловых скобках (<>) или заключены в кавычки в операторе #include).

* Library Directories задает каталоги, в которых содержатся стартовые объектные файлы (с именами C0?.OBJ), файлы библиотек .LIB и другие библиотеки, которые могут использоваться в проекте.

*Output Directory задает тот каталог, в который помещаются файлы с расширениями .OBJ, .EXE и .MAP. При сборке или выполнении программы этот каталог используется системой Borland C для поиска файлов. Кроме того, он используется для проверки даты и времени создания файлов с расширениями .OBJ и .EXE. Если эта строка пуста, то файлы заносятся в текущий каталог.

* Source Directories задает тот каталог, в котором встроенный отладчик будет искать исходные тексты библиотек, не входящих в открытый проект (например, библиотек контейнерного класса).

Команда Environment (Операционная среда)

Позволяет задать параметры, которые будут относиться ко всей среде: Preferences (Приоритеты), Editor (Редактор), Mouse (Мышь) и Desktop (Рабочая область).

Команда Save (Сохранить)

Позволяет осуществить сохранение установок, сделанных как в диалоговых окнах Find и Replace (меню Search), так и в меню Options.

Меню Window (Окно, Alt + W)

Содержит команды управления окнами, в том числе и окно просмотра результатов работы программы user screen (окно пользовательских данных, Alt + F5)

Меню Help (Помощь)

Позволяет вам обратиться к оперативной подсказке

Команда Contents (Содержание)

Открывает окно Help, в котором отображается перечень тем (содержание).

Команда Index (Индекс, Тематический указатель, Shift + F1)

Выводит полный перечень ключевых слов, по которым может быть получена подсказка (выделенный световым атрибутом в экранных кадрах подсказки текст, позволяющий быстро перейти к связанной с данным ключевым словом теме).

Команда Topic Search (Тематический поиск, Ctrl + F1)

Отображает справку по выбранному элементу синтаксиса языка. Для того, чтобы получить справку по языку, необходимо переместить курсор на какой-либо элемент в окне редактирования и выбрать команду Topic Search.

Команда Previous Topic (Предыдущая тема, Alt + F1)

Открывает окно Help, в котором отображается тот текст, который вы просматривали в последний раз при обращении к системе подсказки.

Команда Help on Help (Информация по системе подсказки)

Команда Active File (Активный файл)

Выводит диалоговое окно, позволяющее выбрать файл помощи, который интегрированная интерактивная среда будет использовать.

Команда About (Информация о системе Borland C++)

Краткая справка по часто используемым клавишам

Клавиши режима редактирования			
Команда	Элемент меню	Функция	
Ctrl+Del	Edit Clear(Редактирование Очистка)	Удаляет выбранный текст из окна, но не помещает его в текстовый буфер.	
Ctrl+Ins	Edit Copy (Редактирование Копирование)	Копирует выбранный текст в текстовый буфер	
Shift+Del	Edit Cut (Редактирование Вырезание)	Заносит выбранный текст в текстовый буфер и удаляет выбранный текст.	
Shift+Ins	Edit Paste(Редактирование Вставка)	Вклеивает текст, находящийся в текстовом буфере, в активное окно.	
Alt+Bkspc	Edit Undo (Редактирование Откат)	Восстанавливает текст в активном окне в предыдущее состояние	
Alt+Shift+Bkspc	Edit Redo(Редактирование Отмена)	Восстанавливает текст в активном окне в предыдущее состояние.	
Команды перемещения курсора			
Команда	Оба реж	CUA	Альтерн
Символ влево	←		Ctrl+S

Символ вправо	→		Ctrl+D
Слово влево	Ctrl+ ←		Ctrl+A
Слово вправо	Ctrl+ →		Ctrl+F
Строка вверх	↑		Ctrl+E
Строка вниз	↓		Ctrl+X
Страница вверх	PgUp		Ctrl+R
Страница вниз	PgDn		Ctrl+C
К началу строки	Home		
К концу строки	End		
К верхнему краю окна	Ctrl+Q E	Ctrl+E	Ctrl+Home
К нижнему краю окна	Ctrl+Q X	Ctrl+X	Ctrl+End
К началу файла	Ctrl+Q R	Ctrl+Home	Ctrl+PgUp
К концу файла	Ctrl+Q C	Ctrl+End	Ctrl+PgDn
К последней позиции курсора	Ctrl+P		
Команды вставки и удаления			
Удалить символ по месту курсора	Del		Ctrl+G
Удалить символ слева от курсора	Backspace		Ctrl+H
Удалить строку	Ctrl+Y		
Удалить символы до конца строки	Ctrl+Q Y		
Удалить слово справа от курсора	Ctrl+T		
Вставить строку	Ctrl+N		
Команды работы с блоками			
К началу блока	Ctrl+Q B		
К концу блока	Ctrl+Q K		
Начало отметки блока	Ctrl+K B		
Конец отметки блока	Ctrl+K K		
Выход в меню	Ctrl+K D		
Спрятать/показать блок	Ctrl+K H		
Отметить строку	Ctrl+K L		
Удалить блок	Ctrl+K Y		
Скопировать блок	Ctrl+K C		
Переместить блок	Ctrl+K V		
Считать блок с диска	Ctrl+K R	Shift+Ctrl+R	
Записать блок на диск	Ctrl+K W	Shift+Ctrl+W	

Команды управления окнами			
Выйти из интегрированной среды		Alt+F4	Alt+X
Открыть файл			F3
Сохранить файл	Ctrl+K S		F2
Заккрыть окно			Alt+F3
Отобразить экран пользователя			Alt+F5
Распахнуть окно на весь экран/ вернуть к прежнему состоянию			F5
Изменить размер или положение окна			Ctrl+F5
Переключить активное окно (переход к следующему)			F6

Переключить активное окно (возврат к предыдущему)			Shift+F6
Команды выполнения и отладки программы			
Собрать (скомпилировать) программу			F9
Выполнить программу			Ctrl+F9
Реинициализировать программу			Ctrl+F2
Выполнить программу до строки, на которой установлен курсор.			F4
Выполнить программу в режиме отладки, осуществляя трассирующее вхождение в функции.			F7
Выполнить программу в режиме отладки, не выполняя вызовы функций.			F7
Открыть окно проверки Inspect			Alt+F4
Добавить просматриваемое выражение (Add watch)			Ctrl+F7
Вычислить выражение (Evaluate/Modify)			Ctrl+F4
Установить или отменить точку прерывания программы			Ctrl+F8
Перейти к следующей ошибке			Alt+F8
Перейти к предыдущей ошибке			Alt+F7
Прервать выполнение программы			Ctrl+Break
Команды поиска и замены			
Найти парный символ	Ctrl+Q [Alt+[
	Ctrl+Q]	Alt+]	
Поиск	Ctrl+Q F		
Поиск с заменой	Ctrl+Q A		
Повторить последний поиск		F3	Ctrl+L
Вставить управляющий символ	Ctrl+P (*)		

ДОДАТОК 2

ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Уважно прочитати і зрозуміти умову задачі, яку належить вирішити.
2. Ознайомитися з необхідним теоретичним матеріалом.
3. Вивчити приклади, додатки до відповідної лабораторної роботи, особливу увагу надати лістингу програми (для повного розуміння, можна виконати програму на комп'ютері).
4. Виконати загальні приклади.
5. Підготувати свій варіант програми і вирішити його за допомогою комп'ютера. При введенні і виведенні даних програма повинна видавати запрошення, коментарі і підказки. Програма повинна бути написана у загальному вигляді, тобто початкові дані, розміри масивів і т.п. треба задавати змінними, значення яких можна перевизначити або ввести з клавіатури (файлу).
6. Зробити звіт.
7. Захистити виконану роботу, показавши знання теоретичного матеріалу по пройденій темі і продемонструвавши виконання завдання на комп'ютері.

Кожний звіт оформляється у вигляді записки пояснення згідно ГОСТ 19.001-77 — ЕДИНАЯ СИСТЕМА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ і ГОСТ 2.105- 95 - ЕДИНАЯ СИСТЕМА КОНСТРУКТОРСКОЙ ДОКУМЕНТАЦИИ (на окремих листах формату А4 або в зошиті для лабораторних робіт) і повинен містити наступні елементи:

- титульний лист (при оформленні на окремих листах). При оформленні в зошиті необхідно вказати номер роботи, номер варіанту, прізвище і групу студента, що виконав роботу;
- текст записки пояснення в машинописному або рукописному вигляді;
- список використаної літератури;
- лістинг програми на мові Borland C: роздрукований на принтері або переписаний від руки і електронний варіант – файл з програмою.

Зміст записки пояснення

1. Постановка задачі.
2. Короткі теоретичні відомості про особливості вживання операторів і методів (теоретичне введення).
3. Опис програми:
 - загальні відомості (мова програмування, операційна система, тип процесора);
 - опис логічної структури програми;
 - опис алгоритму рішення задачі (у вигляді блок-схеми згідно ГОСТ 19.002-80 і ГОСТ 19.003-80);
 - опис початкових і вихідних даних програми;
 - опис функцій;
 - перелік аномалій і допустимих значень початкових даних (тестові приклади).
4. Висновки.