



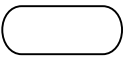
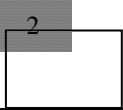
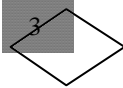
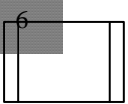
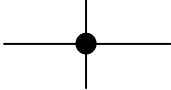
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Донецький національний технічний університет

МЕТОДИЧНИЙ ПОСІБНИК

до лабораторних робіт по дисципліні
«Введення в інформатику»

Донецьк, ДонНТУ, 2009

№ п.п.	Умовне позначення	Найменування	Коментар
1		Пуск, зупинка	Початок та кінець алгоритму
2		Процес	Обчислювальна дія, сукупність обчислювальних дій
3		Розв'язання	Перевірка умови галуження, вибір подальшого спрямування розв'язання залежно від умови
4		Модифікація	Заголовок циклу, перевірка умов циклу
5		Введення, виведення даних	Введення початкових даних, виведення результатів
6		Зумовлений процес	Використання раніше створених алгоритмів (підпрограм)
7		Документ	Вивід даних на друк
8		Сторінковий з'єднувач	Розрив ліній потоку в межах однієї сторінки
9		Міжсторінковий з'єднувач	Перенесення ліній потоку на іншу сторінку
10		Вузол	Злиття ліній потоку

У міру розвитку обчислювальної техніки, ускладнення вирішуваних завдань удосконалюються й алгоритмічні мови – на зміну одним приходять інші, більш досконалі.

В даний час великого поширення набула об'єктно-орієнтована алгоритмічна мова C++. Саме на цю мову зорієнтовані всі завдання до лабораторних робіт по дисципліні, що вивчається.

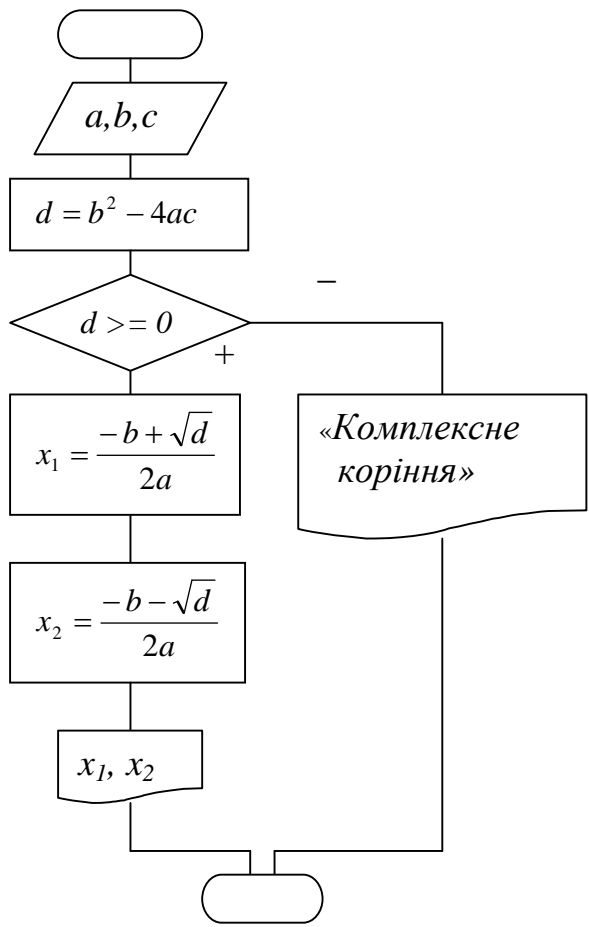


Рис. 2.1 Блок-схема обчислення коріння квадратного рівняння

2.2. Типові алгоритми

Алгоритми більшості завдань будуються на базі простіших, відпрацьованих рішень. Часто використовувані алгоритми називаються *типовими*. Таких алгоритмів не так вже і багато, але їх знання допомагає полегшити розробку більш складних алгоритмів, складовими яких можуть бути типові. Це дозволяє заощадити час розробки і зробити алгоритми більш оптимальними. Нижче розглядаються *типові* алгоритми.

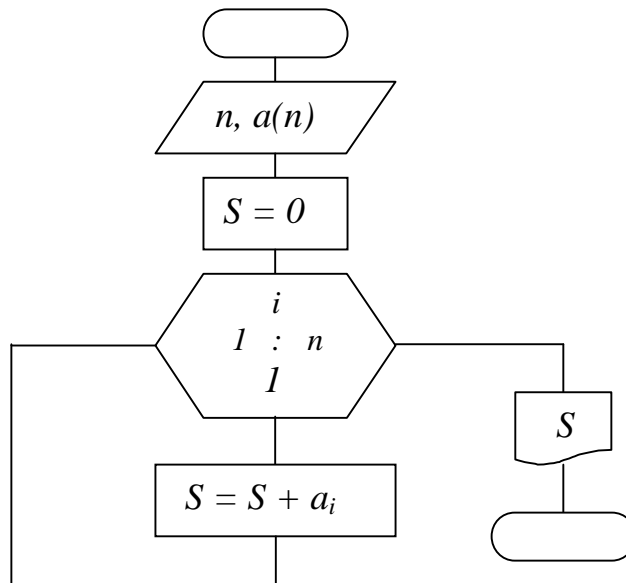
2.2.1. Визначення суми чисел довільного ряду

Даний: ряд довільних чисел $a_1, a_2, a_3, \dots, a_n$, зазвичай званий масивом чисел і що зображується як $a[n]$. Тут a – ім'я масиву, а n – кількість його елементів.

Визначити: суму цих чисел
$$S = \sum_{i=1}^n a_i = a_1 + a_2 + a_3 + \dots + a_n.$$



Алгоритм рішення цієї задачі приведений на мал.2.2.



Мал. 2.2 Блок - схема обчислення суми ряду чисел

Алгоритм передбачає використання циклічної операції, у котрій параметром циклу є індекс елементу ряду i (його порядковий номер). Тіло циклу представлено рекурентною формулою $S = S + a_i$, за допомогою якої подальше значення суми S обчислюється з використанням її попереднього значення. Важливою операцією в цьому алгоритмі є операція $S = 0$, яка передбачає очищення елементу оперативної пам'яті, виділеної для розміщення значень змінної S . Така операція обов'язкова і виконується до початку роботи циклу.

Річ у тому, що при рішенні якої-небудь задачі для всіх даних, приймаючих участь в операціях обчислювального процесу (початкових, проміжних, кінцевих), виділяються елементи оперативної пам'яті. Після завершення виконання програми дані у використовуваних елементах автоматично не видаляються (не анулюються). Якщо після цього на комп'ютері вирішуватиметься інша задача, то для її змінних можуть бути виділені елементи пам'яті, які використовувались у попередньому завданні, зі збереженими в них даними. У випадку, коли вирішується завдання по представленому алгоритму, змінній S може бути виділений елемент з "залишками" від попереднього завдання (їх ще називають "сміття"), і це може спотворити результат рішення. Щоб уникнути цього елемент пам'яті для змінної S спочатку очищують від "сміття", встановлюючи значення 0. Можна замість нуля встановлювати значення першого доданку (першого числа ряду), але тоді параметр циклу i повинний змінитися не від одиниці, а від 2.

Чищення елементів пам'яті передбачається тільки для тих змінних, значення яких обчислюються по рекурентних формулах. При використанні інших формул обчислення така операція не обов'язкова.

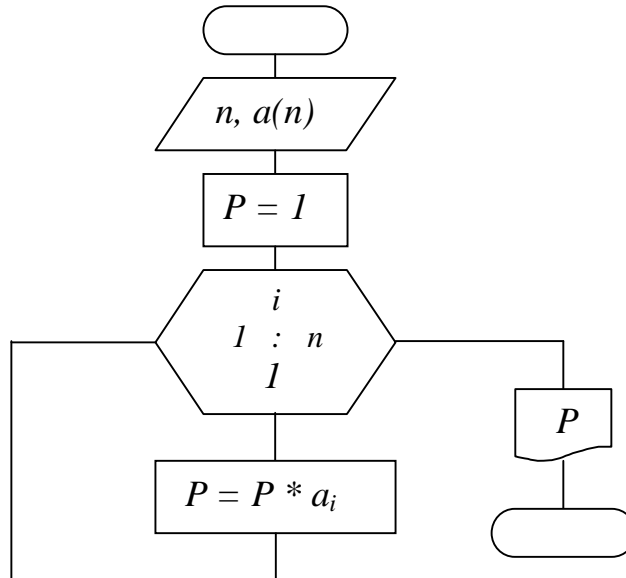
2.2.2. Визначення твору чисел довільного ряду



Даний: ряд довільних чисел $a_1, a_2, a_3, \dots, a_n$.

Визначити: твір цих чисел $P = \prod_{i=1}^n a_i = a_1 * a_2 * a_3 * \dots * a_n$.

Блок-схема обчислювального процесу приведена на мал.2.3.



Мал. 2.3 Блок-схема обчислення твору ряду чисел.

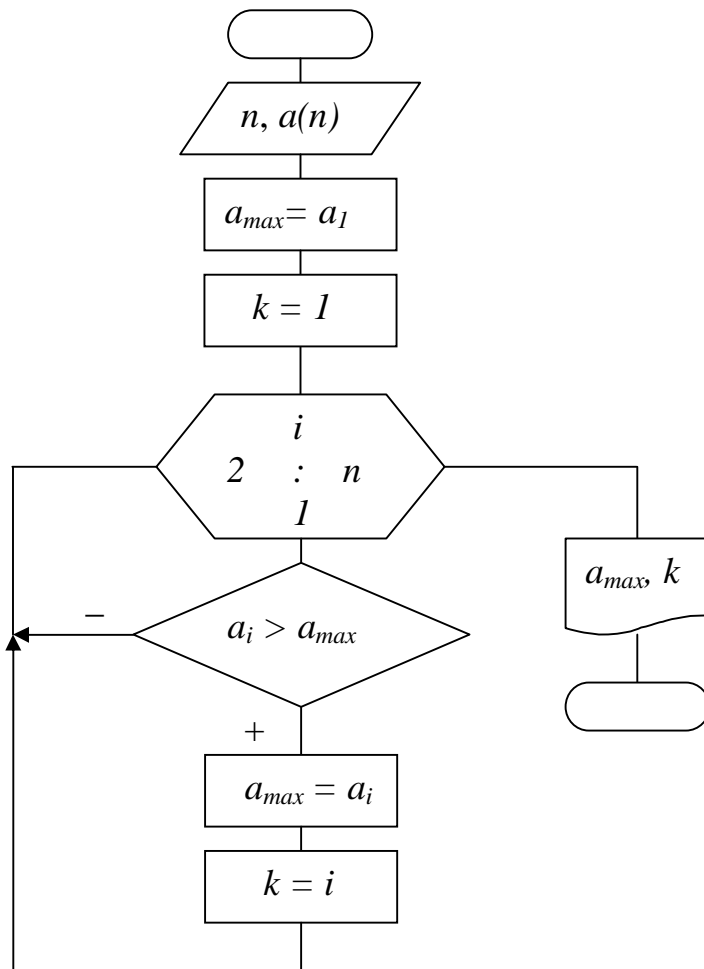
Цей алгоритм подібний до попереднього. Відмінністю є тільки те, що обчислення твору здійснюється по рекурентній формулі, а очищення елементу пам'яті, виділеної під змінну, здійснюється не нулем, а одиницею із зрозумілих причин.

2.2.3. Алгоритм визначення найбільшого (найменшого) числа із заданого ряду чисел

Даний: ряд довільних чисел $a_1, a_2, a_3, \dots, a_n$.

Визначити: найбільше число a_{max} та його місце у рядку k .

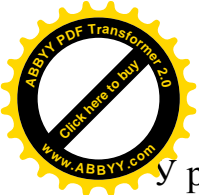
Типовий алгоритм для вирішення таких завдань приведений на мал.2.4



Мал. 2.4 Блок - схема визначення найбільшого числа ряду

Алгоритм заснований на використанні циклічної операції, в якій по черзі проводиться порівняння всіх чисел ряду з числом a_{max} , що грає роль еталону для порівняння. Як початкове значення a_{max} вибирається перший елемент ряду a_1 незалежно від його дійсного значення (найбільше, найменше серед чисел ряду або яке-небудь інше).

Змінна k призначена для зберігання поточної адреси максимального числа, на старті ця адреса - одиниця. Параметр циклу i , як і в попередніх алгоритмах, визначає адресу (порядковий номер) числа ряду і має початкове значення 2, так, як перше число вже використане у ролі a_{max} . Далі виконується порівняння значення поточного числа ряду, що має адресу, з числом, що виконує на даний момент роль максимального a_{max} . Якщо в результаті порівняння поточне число a_i виявиться більшим максимального a_{max} , то з цієї миті роль максимального виконуватиме поточне число a_i , а його адреса буде зафіксована як значення змінної k . Якщо результат порівняння є іншим, то ніякі зміни не відбуваються, а відразу виконується просування циклу, тобто встановлюється адреса i наступного числа ряду і проводиться перехід до нового виконання тіла циклу.



У результаті перебору всіх чисел ряду, визначається найбільше число a_{max} , його адреса k і їх значення виводяться на екран.

Визначення найменшого числа здійснюється аналогічно. Для цього в представленому алгоритмі (мал.2.4) ідентифікатор a_{max} слід замінити a_{min} (оскільки цей ідентифікатор відповідатиме призначенню алгоритму) і операцію порівняння $a_i > a_{max}$ замінити операцією $a_i < a_{min}$.

2.2.4. Алгоритм визначення найбільшого (найменшого) значення обчислюваної функції

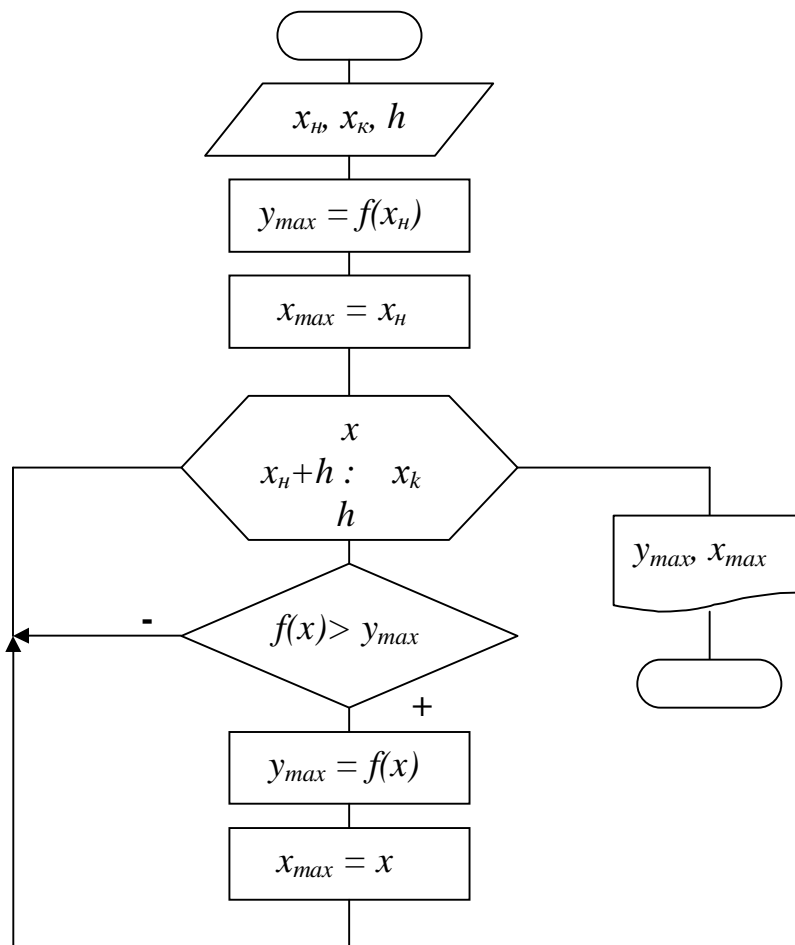
Дана: функція $y = f(x)$, область зміни аргументу x : від x_H (початкове значення), до x_K (кінцевого значення), з кроком зміни h .

Визначити: найбільше значення обчислюваної функції y_{max} на даному інтервалі та значення аргументу x_{max} , при якому функція приймає максимальне значення.

Алгоритм рішення цієї задачі (мал.2.5) аналогічний алгоритму, наведеному на мал.2.4. Відмінність полягає в тому, що в попередньому завданні значення чисел ряду вже були відомі, а в цій їх необхідно заздалегідь визначати

У заголовку циклу, на відміну від попереднього алгоритму, у якості параметру циклу використовується не адреса числа i , а значення аргументу x . Параметр циклу x змінюється не від початкового значення, а від значення $x_H + h$, оскільки значення x_H вже було використано в операції визначення початкового значення функції $y_{max} = f(x_H)$. Змінна x_{max} зберігає поточне значення аргументу, при якому функція приймала максимальне значення. Після завершення роботи циклу змінна y_{max} дорівнюватиметься значенню функції на даному інтервалі при значенні аргументу $x = x_{max}$.

Визначення найменшого значення обчислюваної функції проводиться аналогічно. Для цього в приведеній на мал.2.5 блок - схемі ідентифікатори y_{max} необхідно замінити y_{min} , x_{max} на x_{min} , а операцію порівняння $y > y_{max}$ – на операцію $y < y_{min}$.



Мал. 2.5 Блок - схема обчислення найбільшого значення функції.

2.2.5 Сортування ряду чисел

Сортування це впорядкування (розміщення) чисел за збільшенням або по збуванню їх значень.

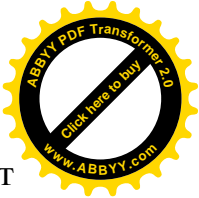
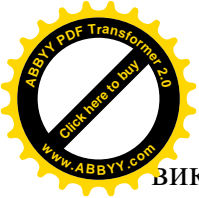
Існує декілька різних способів рішення цієї задачі, проте найчастіше використовуються:

- метод суміжних пар (метод "бульбашки"),
- метод пошуку найменшого (найбільшого) числа.

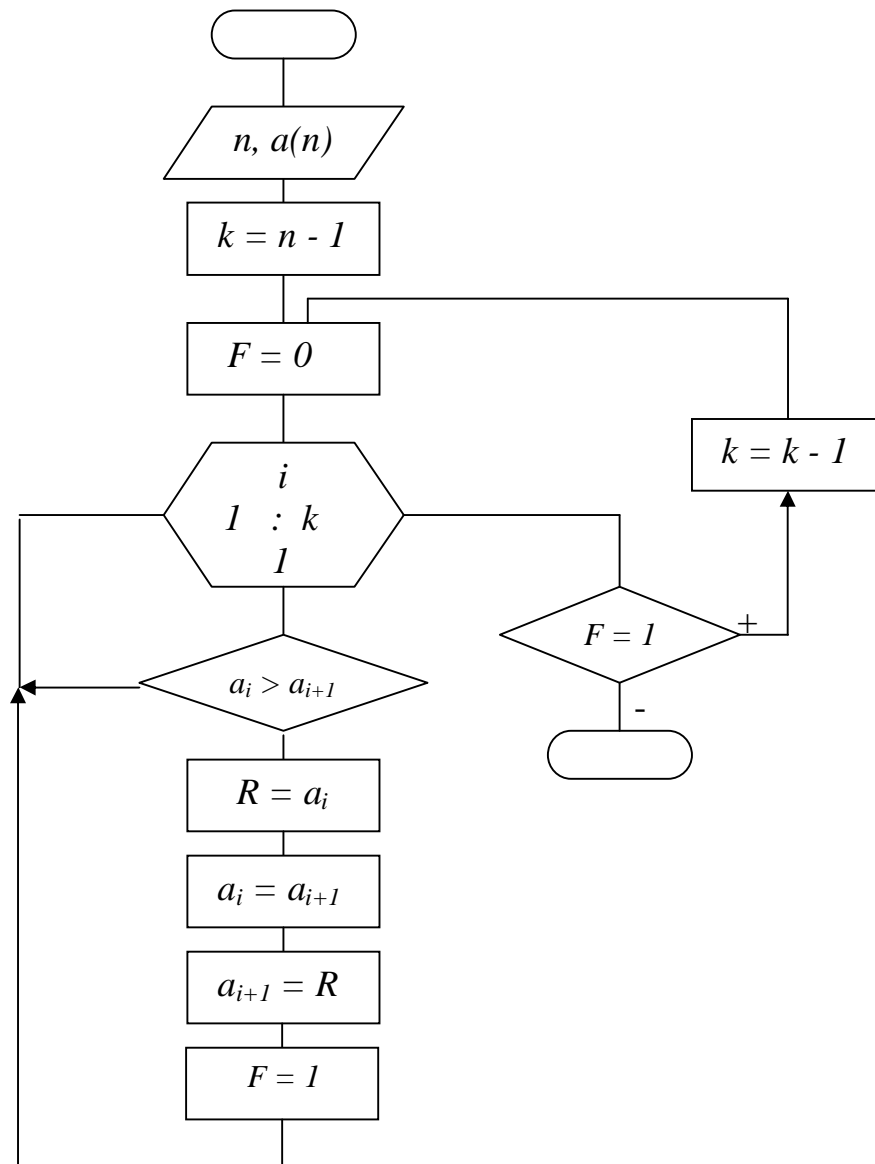
2.2.5.1 Метод суміжних пар

Алгоритм, що реалізує метод суміжних пар, заснований на порівнянні двох сусідніх чисел з подальшою їх перестановкою у випадку, якщо вони розташовані не у потрібній черговості.

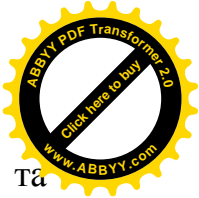
На мал.2.6 приведений алгоритм сортування заданого ряду чисел $a_1, a_2, a_3, \dots, a_n$ за збільшенням. Для регулярного порівняння двох сусідніх чисел передбачається циклічний процес. У заголовку циклу, як параметр циклу,



використовується адреса числа i з його областю зміни від l до $k = n - 1$. Тут слід звернути увагу на те, що права межа області зміни параметра циклу i рівна не n , а $n-1$. Це пояснюється тим, що при порівнянні двох сусідніх чисел адреса першого числа визначається значенням параметра циклу i , а адреса другого числа обчислюється додаванням одиниці ($i+1$). Якби параметр циклу мав би своє останнє значення n , а не $n-1$, то при $i = n$ адреса другого числа визначалася б як $i = n+1$, а це вже виходить за межі даного ряду, і числа з такою адресою немає. Тому, щоб виключити таку ситуацію, останнє значення параметра циклу приймається на одиницю менше.



Мал. 2.6 Блок - схема сортування чисел за збільшенням методом суміжних пар



У тілі циклу відбувається порівняння значень двох сусідніх чисел a_i та a_{i+1} , і у разі їх неправильного розташування здійснюється їх взаємна перестановка за допомогою проміжної змінної R . Змінна R використовується з метою виключення втрати значення одного з чисел при перестановці. Факт перестановки фіксується за допомогою змінної F , яка грає роль індикатора, шляхом установки її значення рівного одиниці. Якщо ж числа розташовані в необхідній черговості, то їх перестановка не відбувається. Після завершення кожного циклу, на останнє місце переміщується саме велике число з чисел, що розглядаються в цьому циклі. Окрім цього, перевіряється стан індикатора F . Якщо його значення рівне одиниці, то це значить, що хоч би одна перестановка в циклі, що завершився, була і необхідно зробити ще один цикл для порівняння сусідніх чисел, але без останнього числа, яке і так вже розташовано на потрібному місці. Виключення останнього числа з розгляду, після завершення чергового циклу, здійснюється операцією $k = k - 1$. Це дозволяє економити машинний час, оскільки немає ніякого сенсу перевіряти розміщення вже розташованого на потрібному місці числа.

Після цього все починається спочатку. Індикатор F встановлюється в стартовий стан із значенням нуль, і в циклі проводиться по чергове порівняння двох сусідніх чисел. Такі цикли повторюватимуться доти, доки не виявиться, що після завершення чергового циклу індикатор F не змінив свого початкового значення 0 на 1. І це означатиме, що у процесі циклічної операції не було жодної перестановки, а, отже, числа розташувалися в необхідній послідовності, тобто ряд відсортовано. Таким чином, алгоритм дозволяє повторювати циклічні операції стільки разів, скільки буде необхідно для завершення процесу сортування.

Сортування ряду чисел по убутанню виконується аналогічно. В цьому випадку необхідно в операції порівняння сусідніх чисел знак " $>$ " замінити знаком " $<$ ".

Метод суміжних пар економічний з погляду витрати машинного часу. Якщо початковий ряд вже сортований в необхідній послідовності, то досить виконати тільки один перевірочний цикл, щоб переконатися, що індикатор F не змінив свого початкового значення 0.

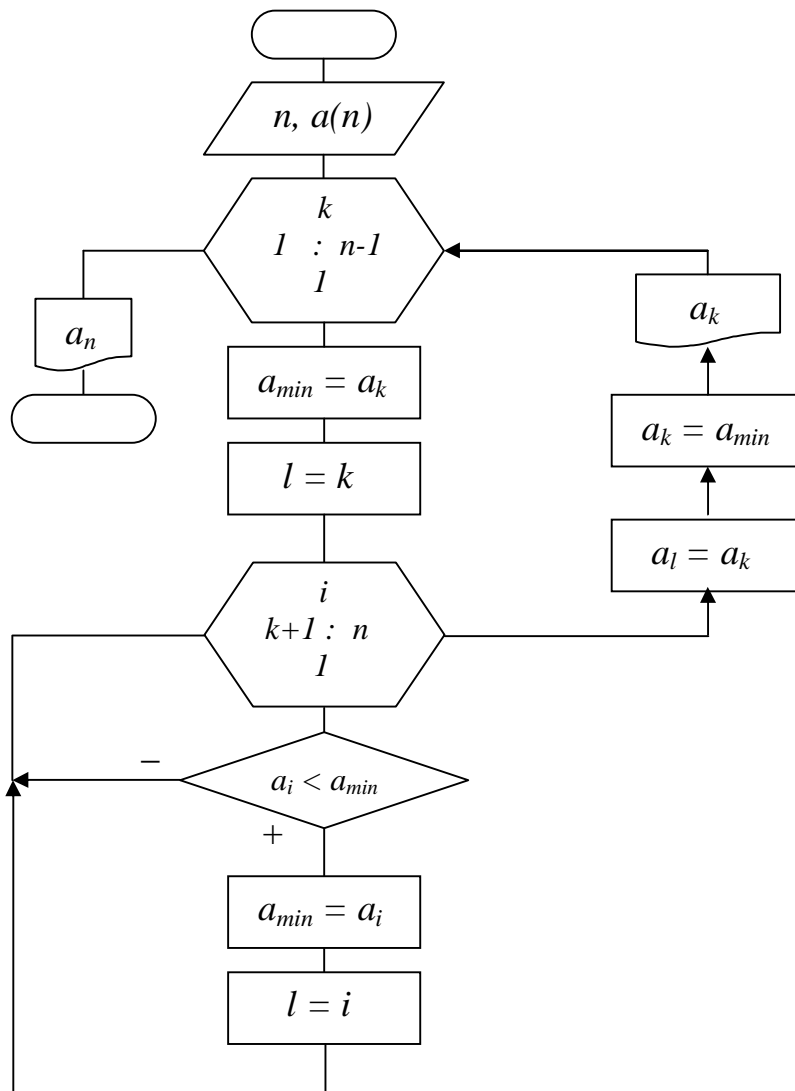
2.2.5.2 Метод пошуку найменшого (найбільшого)

Ідея методу полягає в тому, що визначається місце найменшого числа при сортуванні ряду за збільшенням (найбільшого числа при сортуванні по убутанню), яке потім міняється місцями з першим числом ряду. Після цього ряд чисел розглядається вже без першого числа. Знайдене найменше (найбільше) число виставляється аналогічним чином на друге місце і надалі ряд чисел розглядається вже без двох перших чисел, які виставлені в потрібному порядку. Це повторюється багато разів до тих пір, поки залишиться одне число, після чого процес сортування припиняється.

На мал.2.7 приведений алгоритм сортування чисел за збільшенням. Блок - схема представлена двома вкладеними циклами. Зовнішній цикл з пара-

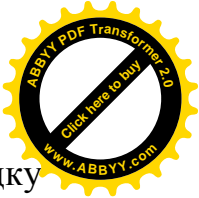
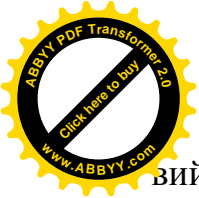
метром циклу k здійснює призначення кожного разу числа на роль мінімального a_{min} в даному ряду чисел і одночасно відсікає вже виставлене уперед число після попереднього циклу.

Вкладений цикл з параметром циклу i є типовим алгоритмом визначення найменшого числа з ряду чисел. Тут роль найменшого числа a_{min} грає число a_k , яке змінюється послідовно від a_1 до a_{n-1} . Вкладений цикл розглядає ряд чисел, починаючи з сусіднього з a_k , тобто з числа a_{k+1} . За межами вкладеного циклу проводиться обмін числами, найменшого і першого в даному ряду. Тут роль змінної R , використаної в методі суміжних пар, грає змінна a_{min} .



Мал. 2.7 Блок - схема сортування чисел за збільшенням методом визначення найменшого числа

На відміну від попереднього методу, цей алгоритм здійснює сортування чисел за «жорсткою» програмою. Тобто, незалежно від того, чи сортований початко-



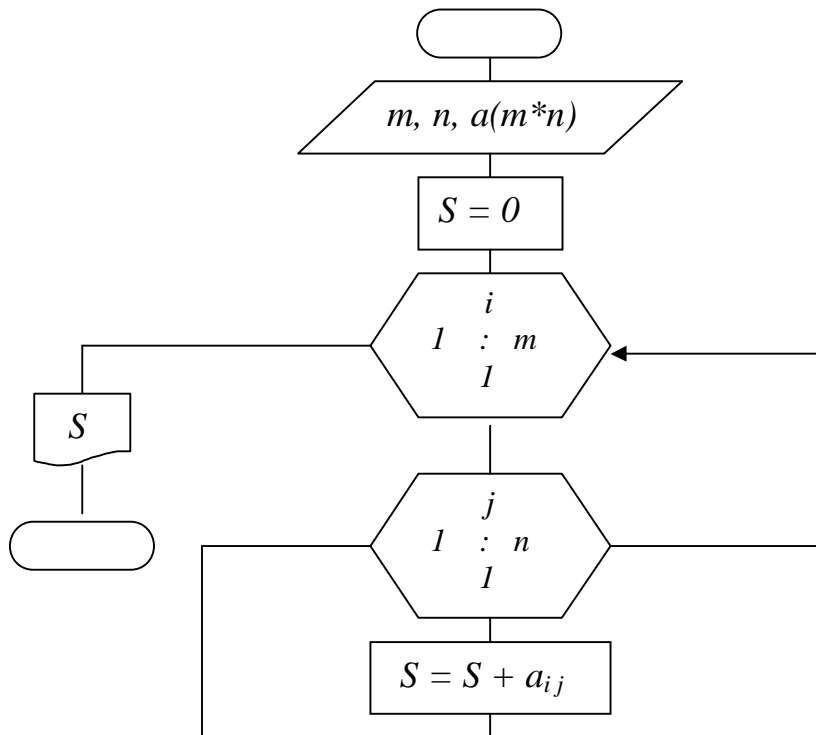
вий ряд чи ні, тіло зовнішнього циклу буде виконано $n - 1$ раз В цьому випадку непродуктивно витрачається машинний час, що визначає недолік методу.

Сортування по убутанню здійснюється аналогічно. При цьому в розглянутій блок-схемі ідентифікатор a_{min} необхідно замінити ідентифікатором a_{max} , що викликає правильні асоціації, а операцію порівняння " $<$ " – на операцію " $>$ ".

2.2.6. Типові алгоритми вирішення завдань з використанням матриць

Відмітною особливістю алгоритмів вирішення завдань з використанням матриць (двовірних масивів) від алгоритмів, орієнтованих на роботу з рядами чисел (векторами), є наявність двох вкладених циклів.

На мал.2.8 приведений алгоритм обчислення суми елементів матриці $a(m*n)$. Зовнішній цикл призначений для встановлення поточної адреси i строки матриці, а внутрішній (вкладений) - поточної адреси j її стовпця.

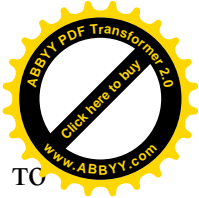
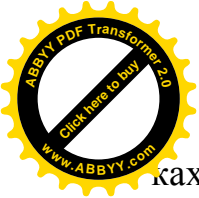


Мал. 2.8 Блок - схема обчислення суми елементів матриці

Обчислення твору P елементів матриці здійснюється по аналогічній блок - схемі, в якій слід замінити символ S на P , операцію $S = 0$ на операцію $P = 1$, а операцію $S = S + a_{ij}$ - на операцію $P = P * a_{ij}$.

В деяких випадках виникає питання, який цикл повинен бути зовнішнім, а який - внутрішнім. Відповідь на нього наступна.

Якщо умовою завдання передбачаються які-небудь операції у строках, то зовнішній цикл повинен визначати поточну адресу рядка, якщо ж передбачуються операції в стовпцях, то зовнішній цикл повинен визначати поточну адресу стовпця. Якщо ж операції над елементами матриці не акцентуються на ряд-

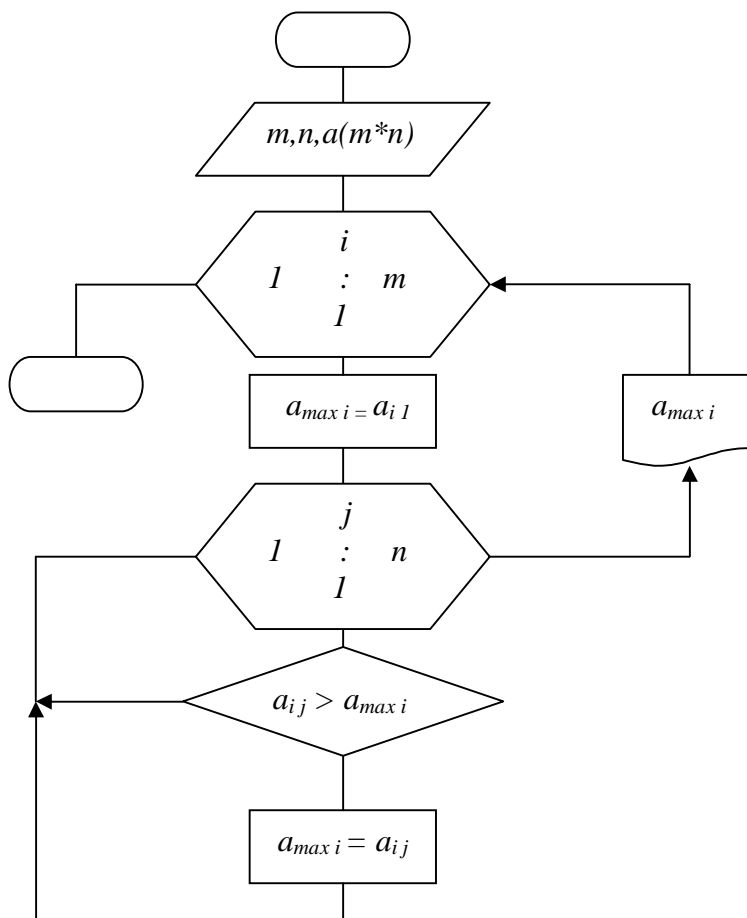


ках або на стовпцях, як у разі обчислення суми або твору елементів матриці, то взаємне розташування циклів може бути будь-яким.

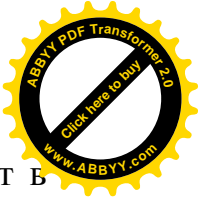
Приклад 1. У поданій матриці $a(m*n)$ визначити найбільші елементи рядків. Алгоритм рішення цієї задачі представлений на мал.2.9.

Тут зовнішній цикл визначає поточна адреса рядка. Отримавши адресу строки i , перший її елемент a_{i1} (елемент першого стовпця) береться у якості початкового значення змінної $a_{max i}$. Вкладений цикл передбачає визначення поточного порядкового номера елементу в даному рядку (номер стовпця), починаючи з другого (перший вже використаний) і є типовим алгоритмом визначення найбільшого числа у ряді чисел (у рядку матриці).

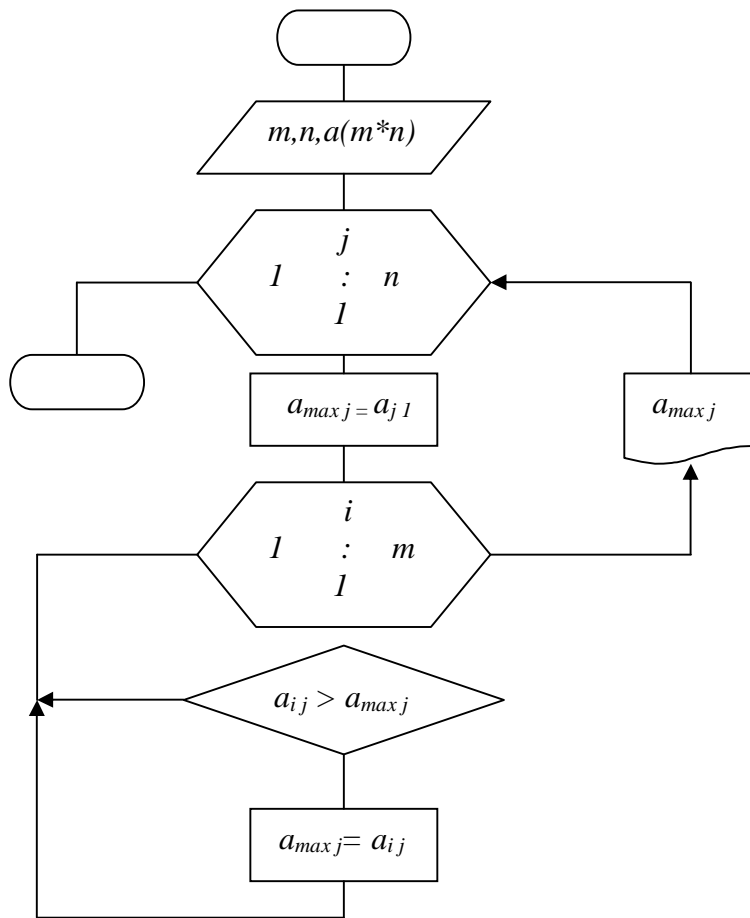
Слід підкреслити, що в завданні потрібно визначити максимальні елементи в рядках, тому зовнішній цикл алгоритму встановлює поточна адреса *рядка*.



Мал. 2.9 Блок-схема визначення максимального елементу у рядку матриці



Приклад 2. У поданій матриці $a(m*n)$ визначити найбільший елемент в кожному стовпці. Алгоритм рішення цієї задачі приведений на мал.2.10.



Мал. 2.10 Блок-схема визначення максимального елементу у стовпці матриці

На відміну від попереднього прикладу в цьому алгоритмі зовнішній цикл визначає поточна адреса *стовпця*.

У разі прикладу з обчисленням суми елементів матриці (мал.2.8) зовнішній цикл може визначати як поточну адресу рядка, так і поточну адресу стовпця, оскільки тут не акцентується місце операцій (рядки або стовпці). Областю операцій є всі елементи матриці без уточнення, в яких рядках або стовпцях вони розташовані.

Приведені алгоритми складають далеко не повний перелік алгоритмів з операціями над елементами матриць. Проте вони можуть бути використані як фрагменти при розробці складніших алгоритмів.

Слід добре розуміти логіку типових алгоритмів, добре *запам'ятати* їх і уміло застосовувати при вирішенні різних завдань. Це дозволить економити час на розробку алгоритмів інших завдань, в яких використовуються типові алгоритми.



1. КОРОТКІ ВІДОМОСТІ ПРО АЛГОРИТМІЧНУ МОВУ C++

Алгоритмічна мова C++ - об'єктно-орієнтована мова, яка успадкувала всі основні властивості структурно-орієнтованої мови C (Cі) і доповнила його новим виглядом змінних - об'єктами (змінними типу клас). Це доповнення додало мові нові якості, істотно розширюючи можливості мови.

Мова C++ заснована на трьох основних концепціях:

- *інкапсуляції* (закритості);
- *поліморфізмі* (використання єдиного інтерфейсу в схожих завданнях);
- *спадкоємстві* (використання наявних розробок).

Автором мови C++ є Бьярн Страуструп, співробітник науково-дослідного центру AT & T Bell Laboratories (Нью-Джерсі, США). Роком створення мови вважається 1979 рік.

Мова C++ зазнала дві істотні модернізації в 1985 і в 1990 рр. Остання, третя модернізація пов'язана з процесом стандартизації мови C++. В даний час існує стандарт мови, розроблений комітетом із стандартизації ANSI (American National Standards Institute) та міжнародною організацією по стандартах ISO (International Standard Organization).

C++ став універсальною мовою для програмістів всього світу. Це єдина мова, яка просто зобов'язана знати будь-який програміст-професіонал, що поважає себе. Багато авторів книг по C++ вважають його мовою XXI століття.

Внаслідок того, що з'являється все більше пакетів прикладних програм для вирішення найрізноманітніших завдань, частка «ручного» програмування для рядового користувача помітно скорочується. Про це мовилося раніше. Тому в дисципліні «Введення в інформатику» висловлюються тільки основні відомості про мову C++, що дозволяють складати прості програми обчислень.

У справжньому методичному посібнику висловлюються рекомендації по виконанню лабораторних робіт відповідно до структури компілятора мови версії Borland C++ .



2. ЗАГАЛЬНІ МЕТОДИЧНІ ВКАЗІВКИ ПО ВИКОНАННЮ ЛАБОРАТОРНИХ РОБІТ

4.1. Порядок проведення лабораторних робіт

Виконанню кожної лабораторної роботи повинна передувати домашня підготовка. Це самостійна робота студента, яка повинна включати:

- повторення теоретичного матеріалу, викладеного під час лекційних занять, що відносяться до теми лабораторної роботи;
- уважне ознайомлення із завданням і методичними рекомендаціями по виконанню роботи;
- розробку алгоритму обчислень (блок-схеми);
- написання програми на алгоритмічній мові;
- підготовку відповідей на питання, приведені в розділі «Контрольні запитання і завдання» в кінці кожної лабораторної роботи.

Перед початком виконання підготовчої частини лабораторної роботи слід уважно вивчити завдання, зрозуміти його суть, а потім приступати до його виконання. При складанні програми особливу увагу слід звернути на потрібну в завданні форму представлення результатів обчислення і строго її дотримувати.

Перед початком виконання практичної частини роботи студент зобов'язаний показати викладачеві складену блок-схему і програму обчислень, відповісти на питання викладача по суті виконуваної роботи. Отримавши дозвіл, студент приступає до її виконання.

Лабораторні роботи виконуються в обладнаному комп'ютерами дисплейному класі. В процесі виконання кожної роботи студент формує файли, що містять тексти програм, підготовлених в процесі самостійної роботи, здійснює відладку програм та їх рішення. Отримані результати необхідно проаналізувати і показати викладачеві. Після отримання позитивної оцінки викладачем отриманих результатів практична частина лабораторної роботи вважається виконаною.

Наступним етапом є підготовка звіту про виконану лабораторну роботу з подальшою його захистом. Звіт складається в години самостійної роботи студента. Форма звіту повинна задовольняти вимогам, викладеним нижче. На наступному після виконання лабораторної роботи занятті звіт повинен бути захищений. Захистом звіту є кроткий виклад студентом суті виконаної лабораторної роботи та відповіді на питання викладача по темі роботи.

Дозвіл на виконання наступної лабораторної роботи студент може отримати тоді, коли захищений звіт по попередній роботі та виконана підготовча частина до нової роботи. У випадку, якщо студент не встигає виконати лабораторну роботу у відведений годинник, він повинен виконати її в години самостійної роботи і представити звіт.

Кожна лабораторна робота містить декілька варіантів завдань. Номером варіанту студента є порядковий номер його прізвища в журналі викладача, що здійснює керівництво лабораторними заняттями. Номер варіанту студента повідомляє викладач.



4.2. Вимоги до оформлення звіту по виконаній лабораторній роботі

1. Текст звіту повинен бути представлений на листах паперу формату А4 з одної боку.
2. Перша сторінка звіту повинна бути титульним листом, виконаний формою, зображеною на рис.4.1.
3. На подальших сторінках в приведеній послідовності повинні бути представлені:
 - мета роботи;
 - завдання;
 - блок-схема алгоритму обчислень;
 - програма (роздрук);
 - результати обчислень (роздрук);
 - аналіз отриманих результатів і виводи.
4. Першим рядком програми повинен бути коментар, представлений по наступній формі:

// ст. гр. ЕАПУ-00А Іванов К.Л., л.р. №1, варіант 10

Текст звіту, блок - схема алгоритму обчислень і математичні вирази слід створювати за допомогою засобів текстового редактора MS Word, розмір шрифту повинен бути рівним 14.

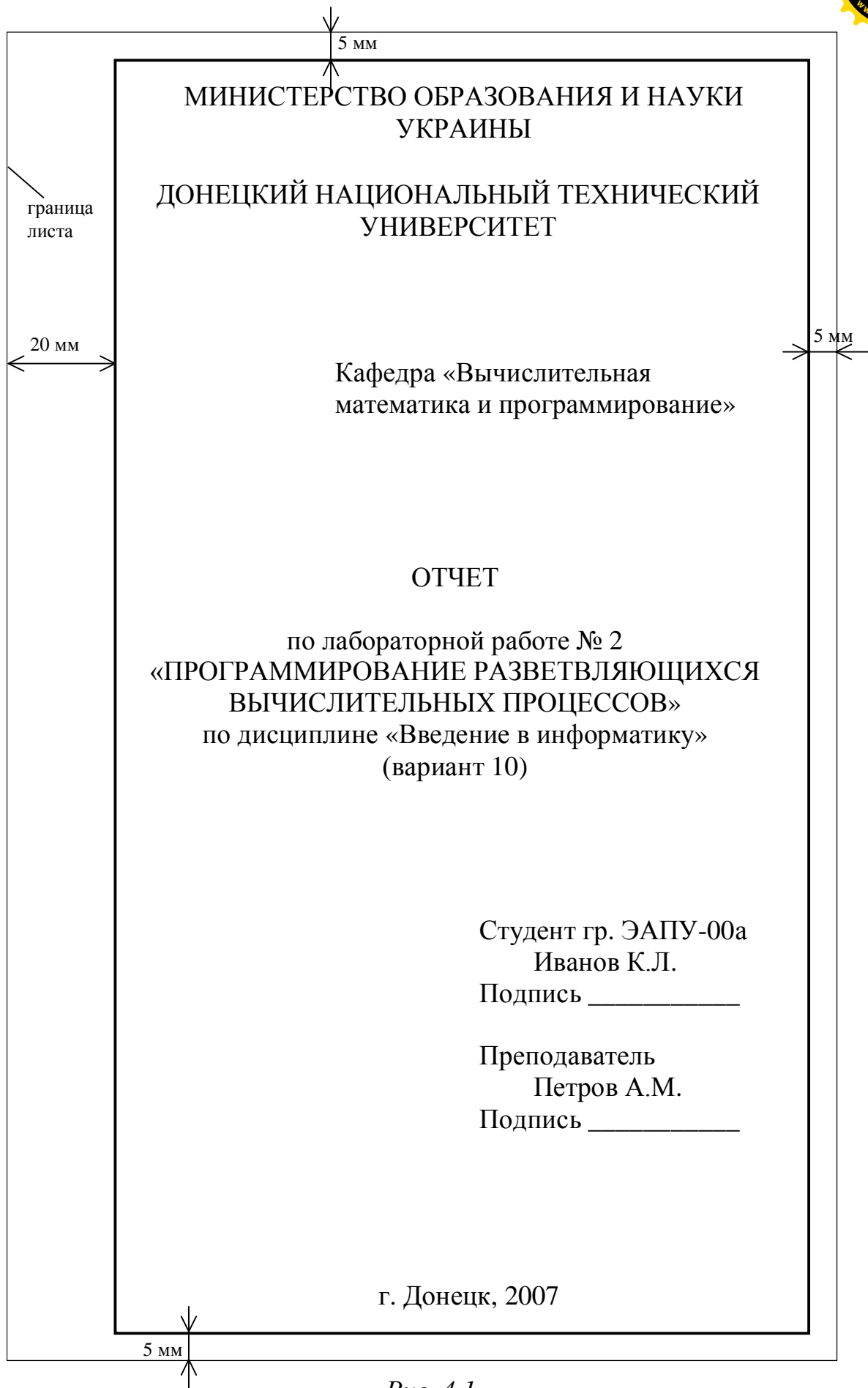
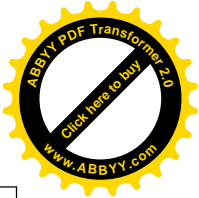


Рис. 4.1.



5. ЦІЛІ ТА ЗМІСТ ЛАБОРАТОРНИХ РОБІТ

5.1. Лабораторна робота № 1

Тема: обчислення виразів з використанням бібліотечних функцій мови C++.

Мета роботи: відпрацювання навиків програмування при обчисленні математичних виразів з використанням бібліотечних функцій мови C++.

5.1.1. Завдання

Розробити алгоритм і скласти програму обчислення функції по формулі, приведеній в табл.5.1. Отриманий результат вивести на екран в наступному вигляді:

РЕЗУЛЬТАТ ПІДРАХУНКІВ
F = ...

5.1.2. Загальні відомості та рекомендації по виконанню роботи

Програма повинна складатися на підставі раніше розробленого лінійного алгоритму. Послідовність всіх операцій над даними у програмі повинна суворо відповідати послідовності їх розташування в блок-схемі. Практика розробки програми, а потім складання по ній блок-схеми недопустима і шкідлива.

При записі математичних виразів слід пам'ятати, що в C++ вони записуються в рядок. Якщо вираз не поміщається в один рядок, його можна переносити на наступний рядок. Розривати вираз рекомендується на знаках операцій, причому при перенесенні знак операції *не повторюється*. Він може поміщатися як на попередньому, так і на подальшому рядку. При записі довгих рядків програми слід залишати справа поля розміром близько 10 - 15% від ширини екрану.

Інший спосіб уникнути довгих рядків - використання проміжних змінних. При цьому слід мати на увазі, що оскільки під кожен змінну відводиться певна частина оперативної пам'яті, то при використанні великої кількості проміжних змінних як програма, так і оперативна пам'ять можуть бути "засмічені".

Не слід забувати, що в алфавіті мови C++ немає грецьких букв (наприклад, α , β , γ та ін.). Замість таких імен змінних можна використовувати ідентифікатори *alfa*, *beta*, *gamma* і т.д. Для позначення, наприклад, змінних з індексами a_n , a_k можна використовувати ідентифікатори *an*, *ak* або *An*, *Ak*. Необхідно пам'ятати, що компілятор мови C++ розрізняє рядкові і прописні букви (тобто, ідентифікатори *A* і *a* сприйматимуться як імена двох різних даних).

Правилом хорошого стилю програмування є позначення імен констант рядковими буквами, а змінних - прописними. Наприклад:

```
const float A = 15.2;    // Константа  
float x = -0.85, y;     // Змінна
```

Особливу увагу слід звертати на розстановку дужок. Від цього може залежати результат обчислень.



Часто викликає утруднення обчислення коріння ступеня вище два (наприклад, $\sqrt[3]{b}$, $\sqrt[5]{a^2}$ та ін.). В цьому випадку доцільно корінь перетворити в статичну функцію вигляду x^y (наприклад, $b^{1/3}$, $a^{2/5}$) та потім скористатися відомими бібліотечними функціями:

$$F = \text{pow}(y,x);$$

або

$$F = \text{exp}(y*\text{log}(x));$$

Аналогічним чином зводиться будь-яке число в дійсну ступінь. Проте при невеликому цілому показнику ступеня (до чотирьох) краще удаватися до заміни операції піднесення до ступеня множенням, де це можливо. Наприклад, замість

$$F = \text{pow}(3,a);$$

Простіше записати

$$F = a*a*a;$$

Слід пам'ятати, що при діленні цілих чисел результат також представлятиметься цілим числом (за наявності дробової частини вона буде відкинута). Таким чином, наприклад, результатом операції $1/3$ буде число 0. Щоб цього уникнути, необхідно хоч би один операнд представляти дробовим числом (наприклад, $1/3.$, $1./3$ або $1./3.$). Після крапки нуль можна не писати. У таких випадках результат доводитиметься до дробового типу числа, і дробова частина не втрачається.

Перед виведенням результату доцільно провести очищення екрану за допомогою стандартної функції

```
clrscr( );
```

Для цього до програми слід приєднати заголовний файл `<conio.h>` за допомогою директиви для препроцесора.

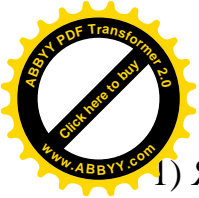
Після завершення виконання програми відбувається повернення в середу компілятора. При цьому результати з екрану дисплея зникають. Для того, щоб проглянути результат слід скористатися комбінацією клавіш `[Alt+F5]`. Зручнішим варіантом буде запис перед останньою дужкою програми `}` оператора виклику стандартної функції очікування натиснення клавіші

```
getch( );
```

із заголовного файлу `<conio.h>`. В цьому випадку результат на екрані буде знаходитися до тих пір, поки не буде натиснута будь-яка клавіша.

Не обов'язково всі рядки програми починати з першої позиції. Рекомендується мати в своєму розпорядженні текст програми уступами, відокремлюючи один логічний фрагмент від іншого. При цьому пари фігурних дужок краще розміщувати на одній вертикалі, а різні пари - на різних вертикалях. Це покращує "читання" програми і вважається хорошим стилем її написання. Першу частину програми (директиви для препроцесора) краще відокремлювати від другої частини (функції) порожнім рядком.

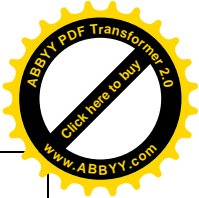
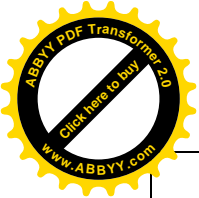
5.1.3. Контрольні запитання та завдання



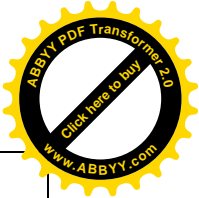
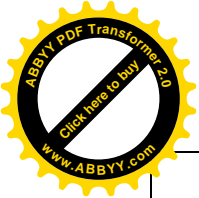
- 1) Яка структура C++ - програми?
- 2) У яких місцях програми можуть розташовуватися оператори опису констант і змінних?
- 3) У чому відмінність операторів опису констант від операторів опису змінних?
- 4) Запишіть вираз $y = \sqrt[3]{(a^2 + e^x)^2}$ з використанням функції pow().
- 5) Запишіть число $23,15 \cdot 10^{-6}$ у формі з фіксованою крапкою і у формі з плаваючою крапкою.
- 6) Чим відрізняється оператор привласнення від операції привласнення?

Таблиця 5.1- Варіанти завдань для лабораторної роботи 1

№ варіанту	Вираз	Початкові дані
1	$Y = \frac{\sin \gamma + \sqrt{\gamma \cdot x + b}}{b \cdot x + \operatorname{tg} \gamma} \cdot \ln \left \frac{\cos x + \sqrt{e^{\gamma \cdot b}}}{b^2 + \operatorname{tg}(\gamma - b)} \right $	$\gamma = 0.8; b = 0.6;$ $x = \gamma + b$
2	$Z = \frac{a^2 + \ln \sqrt{ r \cdot \operatorname{tg} \varphi - c }}{e^\varphi \cdot \operatorname{arctg} \frac{a}{c}}$	$a = 15.8; \varphi = 2.3;$ $r = 3.7; c = -1.2$
3	$\omega = \operatorname{arctg} \left(e^{\frac{\alpha \cdot \beta}{2}} \cdot \sqrt{\frac{\alpha}{\beta}} \right) \cdot \sin \frac{4 \cdot (\beta - \alpha)}{3 \cdot \ln \frac{\beta}{2 \cdot \alpha}}$	$\alpha = 0.37;$ $\beta = \alpha^2 + \cos^2 \alpha$
4	$R = \sin \frac{\sqrt{\ln \operatorname{tga} - e^3 \cos \varphi }}{\sqrt{8 \left \cos^2 a + \sin \frac{\varphi}{2} \right }}$	$a = 2.3; \varphi = \sqrt[3]{a} + 0.5a$
5	$F = \lg \left(e^{\sqrt[4]{a}} \cdot B \right) - \sin^3 \sqrt{\left \operatorname{arctg} \frac{b}{c} \right }$	$a = 13.4; b = 19.3;$ $c = 0.37; B = 2a + b$
6	$Q = \frac{\sin \sqrt{R + \lg^2(4 + 2e^{\sqrt{R}})}}{1 - \ln \frac{a_n + a_k}{a_n a_k}}$	$a_n = 0.04; a_k = 0.76;$ $R = \frac{5}{ \cos a_n a_k }$
7	$S = \frac{e^{\sqrt{\ln(\sin^2 \alpha) + 3 \cos \beta^2}}}{2 \operatorname{arctg} \frac{\sin \alpha}{\cos(\alpha + \beta)}}$	$\alpha = 1.2; \beta = \sqrt[3]{\alpha} - 2\alpha$
8	$\gamma = \frac{0.8 \cdot a^3 + \frac{b - \sin(\lg a^2)}{a + c}}{\sqrt{1 - \cos^2 e^{a+b}}}$	$a = 3.6; b = 4.3;$ $c = \frac{b}{a} + ab$
9	$\psi = 3.7 \cos^2 \sqrt{\left e^{3(a-b^2)} - \frac{\operatorname{tg}(a \cdot b)}{2 \lg(b + 100)} \right }$	$a = 1.74; b = \sqrt[3]{a}$



№ варіанту	Вираз	Початкові дані
10	$H = \frac{4.6 \sqrt{\sin(3 - C_y^3) + \ln \cos^{-2} C_x }}{e^{\cos C_y^2}}$	$C_x = 0.483;$ $C_y = 0.146 \cdot e^{C_x}$
11	$X = \omega \cdot \operatorname{tg} \left(e^{\sin(\omega^2 - 3c)} - \frac{\ln^2 \left \frac{c}{\omega \cdot d} \right }{1 + \sqrt{c^2 + d^2}} \right)$	$c = 12.37; d = -9.04;$ $\omega = c - d $
12	$P = \operatorname{tg} \frac{\sqrt{\sin^3(\gamma \cdot e^{x-1} + b^4) - \ln \frac{c^2}{\gamma - b}}}{\gamma(1 - bc)}$	$\gamma = 3; b = 2.73;$ $x = 0.12; c = \frac{2}{\gamma^b}$
13	$\Phi = \ln^2 \left(1 - \frac{\sqrt{\operatorname{tg}(g - h)^3 + \operatorname{lg}(g^2 + h^2)}}{\sin\left(0.2 + \frac{1}{e^c}\right)} \right)$	$g = 8.36; h = 2.7;$ $c = \frac{g + h}{g(g - h)}$
14	$W = \left \frac{e^{\sin^3 \delta} + \operatorname{lg}(\operatorname{arctg} \delta)}{\sqrt{a \cdot \sin \delta^3}} \right \cdot \sqrt{\frac{a}{2\delta}}$	$a = 0.0138; \delta = 0.9284$
15	$d = \frac{\left X_a - \operatorname{arctg} \frac{X_a}{X_b} \right - \ln^2 \sqrt{X_b - X_a}}{e^{2X_a} \cos^3 \sqrt{X_b}}$	$X_a = -0.84; X_b = 0.538$
16	$y = \frac{e^x + \sqrt[3]{a} - \sin x}{\ln 5x \cdot \left(1 - \frac{a}{b}\right)}$	$x = 1.2; a = 0.8;$ $b = -0.15$
17	$z = \frac{\sqrt[3]{\cos(\ln a)} \cdot \sqrt{5 - x}}{\sin 15b}$	$a = 1.2; x = 4.9;$ $b = 0.018$
18	$\alpha = \frac{1}{3.8 - \frac{\sqrt{ e^x - 1 }}{a^3}}$	$x = -0.8; a = 1.87$
19	$\Gamma = \cos \sqrt{\frac{x^a + a^x}{\operatorname{tg}(b - x^2)}}$	$a = -0.197; x = 2.89;$ $b = 18.36$
20	$K = e^{\operatorname{lg} \sqrt{\sin \alpha - \frac{1}{x^3}}}$	$\alpha = \frac{\pi}{1.7}; x = 2.08$



№ варіанту	Вираз	Початкові дані
21	$M = \frac{\sin \sqrt{\alpha + \lg^2 \left(a + 2e^{\lg x } \right)}}{4 \cos \alpha \cdot \operatorname{arctg} \left(e^{a+b} \right)}$	$a = 0.96; b = 2.07;$ $x = \sin(a^2 - b^2); \alpha = \frac{\pi}{3.6}$
22	$N = \frac{e^x - x e^x}{\cos x} + \sqrt[3]{xy}$	$x = 3.28; y = 0.15$
23	$V = \sin(\omega t) \cdot \frac{U - I \cdot R}{k \cdot \Phi}$	$U = 220; \omega = 1.8;$ $I = 0.5; R = 0.18;$ $t = 2.6; k = 1.2; \Phi = 3.6$
24	$R = \frac{\sqrt{x^2 + y^2} - e^{\sin x}}{\operatorname{arctg} \frac{c}{\cos \alpha} \cdot \ln(c^2 - 1)}$	$x = 1.8; y = 2.3; \alpha = \frac{\pi}{8};$ $c = \sqrt{y^2 - x^2}$
25	$B = \frac{e^{\sqrt{\beta^3 - m} \sqrt{\frac{\ln k}{1.27 - x}}}}{\operatorname{arcsin} x}$	$x = 0.38; k = 4.1;$ $\beta = 18.6; m = x^{\beta - k}$
26	$A = \arccos \left(\sin \alpha - \ln \sqrt{\frac{\cos \alpha}{\alpha^2 - \beta^2}} \right)$	$\alpha = \frac{\pi}{6}; \beta = \frac{\pi}{8}$

5.2. Лабораторна робота № 2

Тема: програмування обчислювальних процесів, що розгалужуються.

Мета роботи: відробіток практичних навиків програмування обчислювальних процесів, що розгалужуються за допомогою умовного оператора.

5.2.1.Завдання

Розробити алгоритм і скласти програму обчислення функції, приведений у табл.5.2 (колонки 1 - 3), для декількох значень аргументу, що вводяться з клавіатури в діалоговому режимі з використанням оператора goto. Результати обчислень представити у вигляді:

РЕЗУЛЬТАТИ ОБЧИСЛЕНЬ

X	Y
...	...
...	...



5.2.2. Загальні відомості та рекомендації по виконанню роботи

Кількість значень аргументу, що вводяться, повинна визначатися, виходячи з наступних міркувань:

- Значення аргументу повинне знаходитися в діапазоні $X_H \dots X_K$ (у останньому ці значення до даної роботи не відносяться);
- На один інтервал обчислення функції повинно доводитися як мінімум два значення аргументу.

Наприклад, якщо функція обчислюється по двох виразах (при яких аргумент знаходиться в різних областях), то слід вводити не менше чотирьох значень, якщо по трьом - не менше шести і так далі

Перед введенням чергового значення необхідно передбачити в програмі вивід на екран повідомлення:

Введіть значення x

Введення числа слід проводити шляхом набору його значення з клавіатури з подальшим натисненням клавіші [Enter].

Рекомендується всі вкладені умови брати у фігурні дужки щоб уникнути логічних помилок.

Для того, щоб при кожному новому значенні x не відправляти програму на компіляцію, слід скористатися оператором безумовного переходу goto, що здійснює повернення до оператора, що виводить на екран повідомлення "*Введіть значення x*". Для програмного контролю кількості вводимих значень аргументу x слід передбачити лічильник вводимих значень і встановити їх граничну кількість. При цьому слід передбачити умову, де повинна порівнюватися поточна кількість вводимих значень x із заданою кількістю. Цим реалізується група операцій, що циклічно повторюється, по обчисленню функції. Такий алгоритм простіше реалізувати за допомогою операторів циклу, але цій темі присвячена наступна лабораторна робота № 4.

Слід пам'ятати, що логічна операція (умова), записувана після ключового слова if, *завжди* повинна полягати в круглі дужки.

Не слід забувати, що в C++ операція порівняння на рівність представляється знаком "=", а не "=". Наприклад, умова:

if (x == 0) ...

записано правильно, а умова

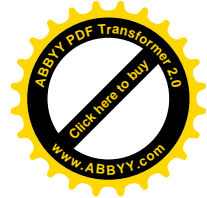
if (x = 0) ...

невірно.

Будь-яка гілка оператора умови повинна містити тільки одного оператора (простий або складений). Отже, якщо в гілці передбачено виконання групи операторів, то вони повинні полягати у фігурні дужки.

Для зручності читання програми слід кожну пару фігурних дужок розташовувати на одній вертикалі, а різні пари - на різних вертикалях, наприклад,

```
void main( )  
{  
...  
оператор 1;  
}
```

```

{
    оператор 2;
    оператор 3;
}
...
}

```

Умови виду $a \leq x < b$ слід записувати одним з приведених способів:

```

if (x >= a && x < b) ...
if ((x >= a) && (x < b)) ...

```

Умовою завершення роботи програми можна узяти перевищення кількості значень аргументу, що вводяться, раннє встановленою межею.

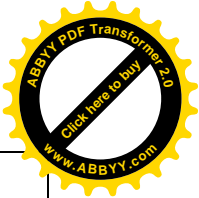
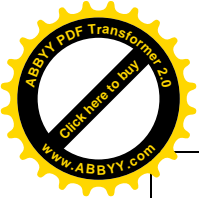
5.2.3. Контрольні запитання та завдання.

- 1) Скільки гілок може містити умовного оператора?
- 2) Скільки операторів може бути в окремій гілці умовного оператора?
- 3) Яка кількість вкладень інших умов може бути в складеному операторі умови?
- 4) Які логічні помилки можуть виникати у разі використання складених (вкладених) операторів умови? Як їх уникати?
- 5) Запишіть оператора для обчислення функції:

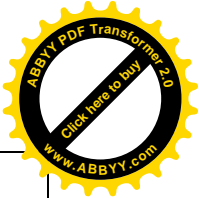
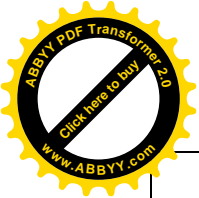
$$y = \begin{cases} -a, & x < -c \\ b \cdot x/a, & -c \leq x \leq d \\ k, & x > d \end{cases}$$

Таблиця 5.2 – Варіанти завдань для лабораторних робіт 2, 3, 4, 11

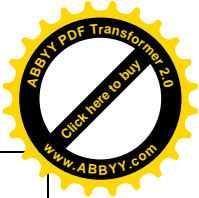
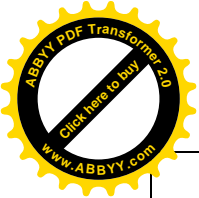
№ варіанту	Вираз	Початкові дані	Межі зміни аргументу
1	$Z = \begin{cases} \frac{b \ln x}{x} + m \cos x, & x \leq 6.5 \\ \frac{2cx + \sin(mx)}{bx + 1}, & x > 6.5 \end{cases}$	$c = 2.7;$ $b = \sqrt[3]{x^2};$ $m = 2.4$	$X_H = 2;$ $X_K = 8.5;$ $H_x = 0.35$
2	$Y = \begin{cases} \frac{a \cdot \sin\left(\frac{\pi}{2}x\right) + b \cdot \cos\left(\frac{\pi}{2}x\right)}{d \cdot \sin(\pi x) + \cos(\pi x)}, & x < 2.4 \\ \frac{\lg x}{d(x-6)}, & x \geq 2.4 \end{cases}$	$a = 2.1;$ $b = 3.2;$ $d = \sqrt{ab}$	$X_H = 0;$ $X_K = 3.8;$ $H_x = 0.2$
3	$g = \begin{cases} \frac{\ln(x+c)}{k(2x+7)}, & x \leq 5 \\ \arctg x + x \cdot \frac{\sin(\pi x)}{2 \cos(\pi x)}, & x > 5 \end{cases}$	$k = 0.2;$ $c = \sin x $	$X_H = -10$ $X_K = 10$ $H_x = 1$



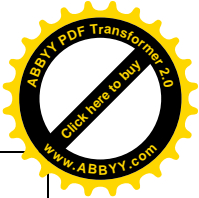
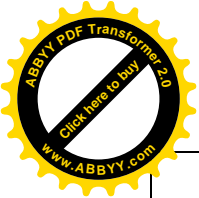
№ варіанту	Вираз	Початкові дані	Межі зміни аргументу
4	$L = \begin{cases} \frac{0.3e^x \sin^3 x}{b(x+2)}, & x \leq \frac{\pi}{2} \\ a\sqrt{e^x \sin^2(x/2)}, & x > \frac{\pi}{2} \end{cases}$	$a = 2.4;$ $b = \ln a$	$X_H = 0;$ $X_K = \pi;$ $H_x = \pi/20$
5	$C = \begin{cases} \frac{x}{\pi m}, & x^2 + y^2 \geq m^2 \\ \frac{\sqrt{x^2 + e^{2x}}}{\sqrt[3]{m}}, & x^2 + y^2 < m^2 \end{cases}$	$m = 5.4;$ $y = 3.2 \operatorname{tg} x$	$X_H = -5;$ $X_K = 5;$ $H_x = 0.5$
6	$F = \begin{cases} \frac{ax^3}{96} + \frac{5x-3}{\sqrt{ 2x-7 }}, & x \leq 4 \\ (x^2+3) \cdot e^{-x} - \frac{\ln x}{kx}, & x > 4 \end{cases}$	$a = 2.2;$ $k = 0.9$	$X_H = 0;$ $X_K = 7;$ $H_x = 0.4$
7	$M = \begin{cases} \frac{\lg^2(x+c)}{a(x^2+b^2)}, & x \leq 6 \\ \frac{x + \sin 2x}{\operatorname{tg} x}, & x > 6 \end{cases}$	$a = 4;$ $b = 1.3;$ $c = \sqrt{\frac{ x }{ab}}$	$X_H = -10;$ $X_K = 10;$ $H_x = 1$
8	$\gamma = \begin{cases} y - \frac{ay^3}{2 \ln y}, & y > 3 \\ \cos^2 ky, & 2 \leq y \leq 3 \\ e^{-y}, & y < 2 \end{cases}$	$a = 0.3$ $k = 3;$ $y = \sqrt{2x}$	$X_H = 1;$ $X_K = 9;$ $H_x = 0.5$
9	$\Phi = \begin{cases} \ln \left \operatorname{tg} \frac{xy}{c} \right , & \sin y \geq \cos x \\ \frac{y \sin x - x \cos(cy)}{5c}, & \sin y < \cos x \end{cases}$	$c = 2.375;$ $y = \sqrt{ x }$	$X_H = -5;$ $X_K = 33;$ $H_x = 2$
10	$R = \begin{cases} a \left(1 + \frac{a^b}{bx^2} \right), & x \leq -5 \\ b(x-1), & -5 < x \leq 1 \\ (x-a)^2, & 1 < x \leq 6 \\ \lg^2(2x^3), & x > 6 \end{cases}$	$a = 8.73;$ $b = -3.4$	$X_H = -10;$ $X_K = 10;$ $H_x = 1$



№ варіанту	Вираз	Початкові дані	Межі зміни аргументу
11	$\sigma = \begin{cases} \frac{\operatorname{tg}(\cos^2(y + 7.2))}{\pi d}, & \pi d \leq y \\ \ln\left(\operatorname{arctg} \frac{y}{d}\right), & \pi d > y \end{cases}$	$d = 5.2;$ $y = \sqrt{x e^x}$	$X_H = 5;$ $X_K = 30;$ $H_x = 2$
12	$\varphi = y + \frac{mx}{1 + \cos^2 x}, \text{ где}$ $y = \begin{cases} 1 + e^{-z}, & z > 3 \\ \cos z + z^2, & -1 \leq z \leq 3 \\ z + \sqrt{ z }, & z < -1 \end{cases}$	$m = 0.48;$ $z = \frac{1.5x}{1-x} + \frac{x}{x^2 - 1}$	$X_H = -1.5;$ $X_K = 5;$ $H_x = 0.35$
13	$U = \begin{cases} \frac{e^x}{\sqrt{x^2 + \pi a^2}}, & (x+a)^2 < 10 \\ 1 + a e^x, & 10 \leq (x+a)^2 < 50 \\ e^{\sqrt[3]{ x }}, & (x+a)^2 \geq 50 \end{cases}$	$a = 2.5$	$X_H = 0;$ $X_K = 20;$ $H_x = 1$
14	$\beta = \begin{cases} 4x^2 - k \sin^2(x/2), & x < 5 \\ 2.5x^2 - 10\sqrt{\operatorname{tg} x}, & 5 \leq x < 10 \\ \frac{5x + c e^{x/2}}{k \lg x^2}, & x \geq 10 \end{cases}$	$k = 0.75;$ $c = 1.38$	$X_H = 0;$ $X_K = 10;$ $H_x = 0.5$
15	$\xi = \frac{3 \sin(\omega \cdot t + x)}{2 \cos(x - \omega \cdot t)}, \text{ где}$ $\omega = \begin{cases} \pi/2 - kx, & x < -\pi/4 \\ 0, & -\pi/4 \leq x \leq \pi/4 \\ \pi - kx, & x > \pi/4 \end{cases}$	$t = 4;$ $k = 0.38$	$X_H = -\pi/2;$ $X_K = \pi/2;$ $H_x = \pi/20$
16	$H = \begin{cases} \sqrt[3]{ y+z }, & x < 6 \\ \operatorname{arctg} \frac{y}{z}, & x = 6 \\ \frac{1}{y e^z}, & x > 6 \end{cases}$	$y = x \operatorname{tg} \frac{x}{4};$ $z = \ln^2 2x $	$X_H = 0;$ $X_K = 8;$ $H_x = 0.4$



№ варіанту	Вираз	Початкові дані	Межі зміни аргументу
17	$C = \begin{cases} \frac{z^3}{2(e^{1.4x} + 1)}, & z < 0 \\ 10z + \cos(\pi\omega), & 0 \leq z \leq 5 \\ \sin(\alpha \cdot z) - \alpha\sqrt{z} , & z > 5 \end{cases}$	$\begin{aligned} \omega &= 0.3; \\ \alpha &= \frac{\pi}{3}; \\ z &= \frac{x^2}{2} - 6 \end{aligned}$	$\begin{aligned} X_H &= 0; \\ X_K &= 5; \\ H_x &= 0.25 \end{aligned}$
18	$W = \begin{cases} \sin(x + y), & x < 0 \\ \arctg y, & 0 \leq x \leq 4, \\ e^{-y} + \sqrt{y}, & x > 4 \end{cases}$ <p>где $y = \frac{(\alpha \cdot x - z)^2}{n}$</p>	$\begin{aligned} z &= 0.3; \\ \alpha &= 1.5; \\ n &= 3.7 \end{aligned}$	$\begin{aligned} X_H &= -2; \\ X_K &= 8.5; \\ H_x &= 0.75 \end{aligned}$
19	$S = x^3 \ln^2 Z + 1, \text{ где}$ $Z = \begin{cases} \sqrt{x + 2}, & x < 0 \\ \frac{1.5x^2 - 1}{2k}, & 0 \leq x \leq 3 \\ 6.5 \sin x^2 + b, & x > 3 \end{cases}$	$\begin{aligned} k &= 3.2; \\ b &= 2 \end{aligned}$	$\begin{aligned} X_H &= -3; \\ X_K &= 5; \\ H_x &= 0.4 \end{aligned}$
20	$Y = \begin{cases} e^{\lg x} + 2.3x^2, & x < 5 \\ \frac{ax - ax^2}{\ln x}, & x \geq 5 \end{cases}$	$a = 2.7$	$\begin{aligned} X_H &= 0.5; \\ X_K &= 7; \\ H_x &= 0.5 \end{aligned}$
21	$V = \begin{cases} \frac{\lg^2(x + a) - e^x}{2(b - x)}, & x > 3 \\ \frac{\sqrt{x^2 + 2 \sin^2(ax)}}{x^b}, & x \leq 3 \end{cases}$	$\begin{aligned} a &= 0.36; \\ b &= 5.7 \end{aligned}$	$\begin{aligned} X_H &= -1.6; \\ X_K &= 5; \\ H_x &= 0.4 \end{aligned}$
22	$L = \begin{cases} ax^2 + bx + c, & x < 0.5 \\ \frac{\ln x}{e^{-x}}, & 0.5 \leq x \leq 2.7 \\ \sqrt{\frac{ax}{b}} - c, & x > 2.7 \end{cases}$	$\begin{aligned} a &= 0.3; \\ b &= 4.2; \\ c &= -3.7 \end{aligned}$	$\begin{aligned} X_H &= -3; \\ X_K &= 5; \\ H_x &= 0.4 \end{aligned}$
23	$N = \begin{cases} \ln \sqrt{\ln x + a}, & x < 0 \\ \frac{\sin(a + x)}{e^{\sqrt{\cos(a+x)}}}, & x \geq 0 \end{cases}$	$a = 4.7$	$\begin{aligned} X_H &= -3; \\ X_K &= 4.5; \\ H_x &= 0.5 \end{aligned}$



№ варіанту	Вираз	Початкові дані	Межі зміни аргументу
24	$M = \begin{cases} \alpha x^2 + \beta \sin x, & x < 0 \\ (\alpha^2 x + \beta) \cdot \operatorname{tg} x, & 0 \leq x \leq 1.5 \\ \alpha^{x+\beta}, & x > 1.5 \end{cases}$	$\alpha = 0.95;$ $\beta = 3.9$	$X_H = -2;$ $X_K = 4;$ $H_x = 0.3$
25	$\theta = \begin{cases} \frac{\sqrt{ x^5 } - b}{2x}, & x \leq -1 \\ \frac{\sin x}{1 + \operatorname{tg}^2 x}, & x > -1 \end{cases}$	$b = 0.35$	$X_H = -4;$ $X_K = 3;$ $H_x = 0.35$
26	$\alpha = \begin{cases} \frac{a+x}{a^3-x} + e^x, & x > 0 \\ \frac{\cos(x + \ln x)}{\sqrt{b-3.7x}}, & x \leq 0 \end{cases}$	$a = 1.9;$ $b = 3.15$	$X_H = -3.2;$ $X_K = 4.1;$ $H_x = 0.4$



5.3. Лабораторна робота № 3

Тема: програмування розгалужених обчислювальних процесів з використанням оператора множинного вибору.

Мета роботи: відробіток навиків програмування обчислювальних процесів із застосуванням оператора множинного вибору.

5.3.1. Завдання

Розробити алгоритм і скласти програму обчислення функції, приведений в табл.5.2 (колонки 1 - 3), для декількох значень аргументу, що вводяться з клавіатури. При програмуванні обчислень використовувати оператора множинного вибору *switch*, що дозволяє проводити обчислення значень функції по будь-якому з приведених виразів на вибір.

5.3.2. Загальні відомості та рекомендації по виконанню роботи

Оператор множинного вибору *switch*, або, як його ще називають, перемикач, дозволяє представляти в програмі в компактнішій формі і в більш читабельному вигляді складені оператори умови з великою кількістю вкладень (зазвичай його застосовують, коли кількість вкладень більше трьох).

При розробці алгоритму слід передбачити можливість вибору за встановленим кодом того або іншого виразу для обчислення функції залежно від введенного з клавіатури значення x . Крім того, слід передбачити можливість завершення програми за певним кодом.

Вимоги до кількості і діапазону значень, що вводяться, ті ж, що і в лабораторній роботі № 2.

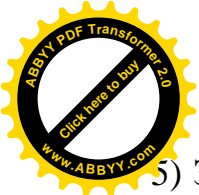
Після завершення чергового обчислення функції слід передбачити вивід на екран повідомлення «Введіть нове значення x або код завершення програми».

При введенні неправильного коду програмою повинен передбачатися вивід на екран повідомлення «Код введений неправильно. Повторіть введення коду».

Кодами можуть служити мітки операторів перемикача *switch*. Для повторення введення нових значень x слід використовувати оператор *goto*.

5.3.3. Контрольні питання та завдання

- 1) У яких випадках використовується оператор *switch*?
- 2) Яка роль операторів *break* в операторові *switch*?
- 3) Чи обов'язкова присутність в операторові *switch* гілки *default*?
- 4) Що відбудеться, якщо який-небудь помічений міткою *case* оператор не завершуватиметься оператором *break*?



5) Запишіть оператора *switch*, що дозволяє по вибору виводити на екран повідомлення про вибір якого-небудь кольору з декількох можливих (червоний, синій, жовтий, зелений, чорний). Наприклад, «*Ви вибрали червоний колір*».



5.4. Лабораторна робота № 4

Тема: програмування з використанням простих циклів.

Мета роботи: відробіток навиків програмування циклічних обчислювальних процесів.

5.4.1. Завдання

Розробити алгоритм і скласти програму обчислення функції, наведеної в табл.5.2, при зміні значень аргументу x від X_H до X_K з кроком H_x . Визначити суму позитивних і твір негативних значень функції.

На екран вивести поточні значення аргументу та функції, остаточне значення суми позитивних і твору негативних значень функції. Виведення результатів на екран здійснити в наступному вигляді:

РЕЗУЛЬТАТИ ОБЧИСЛЕНЬ	
X	Y
...	...
...	...
Сума позитивних значень функції	S =...
Твір негативних значень функції	P =...

За відсутності якого-небудь виду значень функції (позитивних або негативних) замість повідомлення про суму або твір на екран вивести повідомлення «*Позитивних (негативних) значень немає*».

5.4.2. Загальні відомості та рекомендації по програмуванню

У C++ існує три види операторів циклу:

- while (з передумовою);
- for (з передумовою);
- do...while (з післяумовою).

Найбільш універсальним з них є оператор циклу **for**, який найчастіше і використовують програмісти.

У специфікаціях циклу **for** слід використовувати не чисельні значення X_H , X_K та H_x , а їх ідентифікатори.

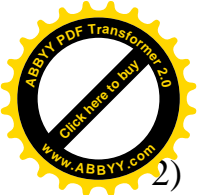
Слід пам'ятати, що тіло циклу може представлятися *тільки одним оператором* (простим або складеним). Недотримання цього правила може привести до спотворення результату обчислень.

При виведенні результатів на екран слід використовувати маніпулятори `setw()`, `setprecision()`, `setiosflags()` и др.

Хорошим стилем програмування вважається запис операторів, що використовують рекурентні формули, з використанням модифікацій оператора приписування (наприклад, $x += h$ замість $x = x + h$, $x++$ замість $x = x + 1$ і т.д.)

5.4.3. Контрольні питання та завдання

1) Які Ви знаєте види циклів? Які правила їх запису (синтаксис)?



- 2) Перерахуйте можливості (модифікації) заголовка циклу **for**?
- 3) Чи може цикл **for** використовуватись замість **while** та **do...while**?
- 4) Як уникнути переривання обчислювального процесу у разі виникнення ситуації ділення на нуль, витягання квадратного кореня з негативного числа, обчислення логарифма нуля або негативного числа?
- 5) Запишіть оператора циклу **for** для обчислення функції

$$y = \frac{kx}{a - x},$$

якщо $a = 2.9$, $k = 1.6$, а x змінюється від $X_H = -4.8$ до $X_K = 8.2$ з кроком $H_x = 0.1$. Виключення переривання обчислювального процесу передбачити за допомогою оператора *continue*.

- 6) Напишіть фрагмент програми для обчислення функції п.5 за допомогою операторів циклу **while** та **do...while**.



5.5. Лабораторна робота № 5

Тема: програмування з використанням ітераційних циклів.

Мета роботи: закріплення навиків програмування ітераційних циклів.

5.5.1. Завдання

Розробити алгоритм і скласти програму обчислення виразу, представленого в табл.5.3.

На екран виводити номер ітерації та значення змінної y , а після закінчення циклу - значення результату циклічного процесу (суму значень y), кількість ітерацій, значення перевіркової функції та відносної погрішності. Результати обчислень представити в наступному вигляді:

ІТЕРАЦІЙНИЙ ЦИКЛ	
РЕЗУЛЬТАТИ ОБЧИСЛЕНЬ	
I	Y
...	...
...	...
РЕЗУЛЬТАТ ЦИКЛІЧНОГО ПРОЦЕСУ	
S = ...	
ПЕРЕВІРОЧНИЙ РЕЗУЛЬТАТ	
Sp=...	
ВІДНОСНА ПОГРІШНІСТЬ, %	
d = ...	

5.5.2. Загальні відомості та поради по програмуванню

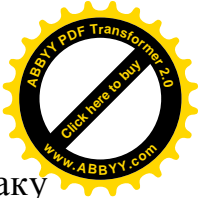
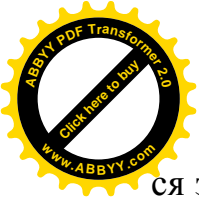
Ітерація - це послідовне наближення до чого-небудь.

Цикли, що використовують ітерації, називають *ітераційними*. Закінченням ітераційного циклу є досягнення якої-небудь певної умови (найчастіше - умови необхідної точності). Кількість кроків в ітераційних циклах, на відміну від простих, встановити наперед практично неможливо. Воно стає відомим після завершення роботи циклу.

У цій роботі рекомендується використовувати оператори цикли `while` и `do...while`, хоча приведені завдання можна вирішувати і за допомогою циклу `for`, оскільки мова C++ це дозволяє.

При розробці алгоритму слід мати на увазі, що використання масиву для зберігання значень y_i в даній роботі недоцільно, оскільки, по суті роботи, при кожному виконанні тіла циклу нас цікавить лише поточне значення змінної y та (у деяких варіантах завдань) її попереднє значення. Тому, можна ввести ідентифікатор для зберігання поточного значення (наприклад, Y) та, у разі потреби, ідентифікатор для зберігання попереднього або подальшого значення (наприклад, $Y1$). Це дозволить використовувати мінімум оперативної пам'яті і зробити програму раціональнішою.

У приведених прикладах часто зустрічаються випадки обчислення ступінчатих функцій, у яких показник ступеня, будучи параметром циклу, змінюється



ся з певним кроком. Наприклад, x^{5i} , де i – параметр циклу. Обчислювати таку функцію при кожному виконанні циклу за допомогою бібліотечної функції `pow()` або `exp()` нераціонально і не завжди зручно. Простішим і раціональнішим способом є використання рекурентної формули, у котрій подальше значення статечної функції обчислюється шляхом творення його попереднього значення на деякий коефіцієнт. Як це робити видно на загальному прикладі.

Хай попереднє значення статечної функції рівне x^{5i} , тоді наступне її значення рівне $x^{5(i+h)}$, де h – крок зміни параметра циклу i . Після перетворення

$$x^{5(i+h)} = x^{5i} \cdot x^{5h}$$

видно, що подальше значення функції обчислюється шляхом множення її попереднього значення $b = x^{5i}$ на постійний коефіцієнт $k = x^{5h}$. В цьому випадку можна використовувати рекурентну формулу $b = b * k$ або у C++ як

`b *= k;`

Аналогічно можна організувати поточне значення факторіалу

$$i! = \begin{cases} 1 \cdot 2 \cdot 3 \cdot \dots \cdot i, & \text{якщо } i \in \mathbb{N}, i \geq 1 \\ 1, & \text{якщо } i = 0 \end{cases}$$

по рекурентній формулі.

При записі умови виконання циклу часто допускається характерна помилка, коли як він використовується вказане в табл.5.3 умова завершення циклу. В цьому випадку, оскільки в C++ в заголовку будь-якого циклу вказується умова його виконання, цикл завершує свою роботу, навіть не почавши її (у разі циклу з передумовою), або ледве почавши (у разі циклу з післяумовою). Наприклад, якщо умова завершення циклічного процесу має вигляд $y < \varepsilon$ (тобто досягти цього обчислення слід припинити), то в заголовку циклу як умова виконання процесу слід вказати зворотну умову $y \geq \varepsilon$, а не наведене у табл.5.3.

Відносну погрішність обчислень (у відсотках) слід обчислювати по формулі:

$$d = \left| \frac{S - Sp}{Sp} \right| \cdot 100.$$

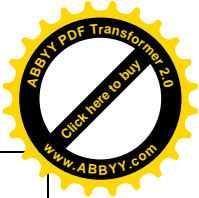
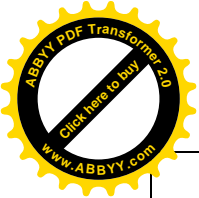
5.5.3. Контрольні питання та завдання

- 1) У чому відмінність циклу `while` від циклу `do...while`?
- 2) Чи можливе у C++ використання циклу `for` замість циклів `while` та `do...while`?
- 3) Складіть програму для обчислення суми ряду чисел

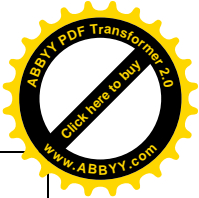
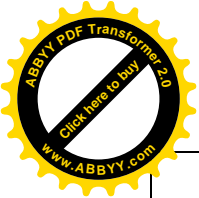
$$S = y_1 + y_2 + \dots + y_n = \sum_{i=1}^n y_i, \text{ де } y_1 = 1, x = 2.7, \text{ а } y_{i+1} = \frac{1}{2} \left(y_i + \frac{x}{y_i} \right).$$

Обчислення припинити при виконанні умови $|y_i - y_{i-1}| < \varepsilon$ при $\varepsilon = 1.5E - 05$.

Таблиця 5.3 – Варіанти завдань до лабораторної роботи 5



№ вар.	Вираз	Результат цикл. процесу	Умова закінчення процесу, перевірна формула	Початкові дані
1	$y_{i+1} = y_i(2 - xy_i)$	$Z = \sum_{i=0}^{n+1} y_i$	$ y_{n+1} - y_n < \varepsilon$ $y_{n+1} \approx \frac{1}{x}$	$x = 3;$ $y_0 = 0.25;$ $\varepsilon = 1.0e - 05$
2	$y_i = \frac{x^i}{i!}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx e^x$	$x = 0.5;$ $\varepsilon = 1.0e - 06$
3	$y_{i+1} = \frac{1}{2} \left(y_i + \frac{x}{y_i} \right)$	$Z = 3 \sqrt[n]{\prod_{i=0}^n y_i}$	$ y_{n+1} - y_n < \varepsilon$ $y_{n+1} \approx \sqrt{x}$	$x = 5;$ $y_0 = x;$ $\varepsilon = 1.0e - 04$
4	$y_i = (-1)^i \frac{x^{2i+1}}{(2i+1)!}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx \sin x$	$x = \pi/8;$ $\varepsilon = 1.0e - 05$
5	$y_{i+1} = \frac{1}{3} \left(2y_i + \frac{x}{y_i^2} \right)$	$Z = \ln \left(\prod_{i=0}^{n+1} y_i \right)$	$ y_{n+1} - y_n \leq \varepsilon$ $y_{n+1} \approx \sqrt[3]{x}$	$x = 10;$ $y_0 = \frac{x}{3};$ $\varepsilon = 1.0e - 08$
6	$y_i = \frac{x^{2i}}{(2i)!}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx \cos x$	$x = 2;$ $\varepsilon = 3.0e - 06$
7	$y_{i+1} = y_i + \frac{1}{3} \left(\frac{x}{y_i^2} - y_i \right)$	$Z = \lg \left(\prod_{i=0}^{n+1} y_i \right)$	$ y_{n+1} - y_n < \varepsilon$ $y_{n+1} \approx \sqrt[3]{x}$	$x = 5;$ $y_0 = \frac{x}{2};$ $\varepsilon = 2.0e - 05$
8	$y_i = (-1)^{i+1} \frac{x^i}{i!}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx \ln(1+x)$	$x = 0.7;$ $\varepsilon = 1.0e - 04$
9	$y_{i+1} = 0.5y_i(3 - xy_i)$	$Z = \cos \left(\sum_{i=0}^{n+1} y_i \right)$	$ y_{n+1} - y_n < \varepsilon$ $y_{n+1} \approx \frac{1}{x}$	$x = 3;$ $y_0 = 0.5;$ $\varepsilon = 1.0e - 05$
10	$y_i = \frac{x^{2i+1}}{(2i+1)!}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx \sin x$	$x = 2.5;$ $\varepsilon = 1.0e - 08$
11	$y_i = (-1)^{i-1} \frac{2i+1}{(2i)!} x^{2i}$	$S = \sum_{i=1}^n y_i$	$ y_n < \varepsilon$ $S \approx x \sin x$	$x = 0.53;$ $\varepsilon = 1.0e - 07$
12	$y_i = (-1)^{i-1} \frac{2x^{2i}}{4i(2i-1)}$	$S = \sum_{i=1}^n y_i$	$ y_n < \varepsilon$ $S \approx \frac{x}{2} \arctg x$	$x = 0.2;$ $\varepsilon = 1.0e - 05$



13	$y_i = (2i - 1)x^{2i-1}$	$S = \sum_{i=1}^n y_i$	$ y_n < \varepsilon$ $S \approx \frac{1}{2} \ln \frac{1+x}{x}$	$x = 0.4;$ $\varepsilon = 1.0e - 04$
14	$y_i = \frac{2i+1}{i!} x^{2i}$	$S = \sum_{i=1}^n y_i$	$ y_n < \varepsilon$ $S \approx (1+2x^4) \cdot e^{x^2}$	$x = 0.7;$ $\varepsilon = 1.0e - 05$
15	$y_i = \frac{1}{2i+1} \left(\frac{x-1}{x+1} \right)^{2i+1}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx \frac{1}{2} \ln x$	$x = 0.85;$ $\varepsilon = 5.0e - 05$
16	$y_i = i(i+2)x^i$	$S = \sum_{i=1}^n y_i$	$ y_n < \varepsilon$ $S \approx \frac{x(2-x)}{1-x}$	$x = -0.2;$ $\varepsilon = 2.0e - 08$
17	$y_i = (-1)^{i-1} \frac{i-1}{i!} x^i$	$S \approx 1 + \sum_{i=2}^n y_i$	$ y_n < \varepsilon$ $S \approx (1+x)e^{-x}$	$x = -0.4;$ $\varepsilon = 1.0e - 04$
18	$y_i = (-1)^i \frac{x^{2i+1}}{(2i+1)i!}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx \frac{2}{\pi} e^x - 1$	$x = 1.5;$ $\varepsilon = 1.0e - 05$
19	$y_i = \frac{x^{2i}}{i!}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx e^{x^2} + 1$	$x = 0.25;$ $\varepsilon = 5.0e - 07$
20	$y_i = (-1)^i \frac{x^{2i+1}}{2i+1}$	$S = \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx \text{arctg} x$	$x = 0.4;$ $\varepsilon = 1.0e - 08$
21	$y_i = \frac{(-4)^i x^{4i}}{(4i)!}$	$S = \sum_{i=1}^n y_i$	$ y_n < \varepsilon$ $S \approx \text{ch} x \cos x - 1$	$x = \frac{\pi}{4};$ $\varepsilon = 1.0e - 05$
22	$y_i = (i+1)(i+2)(-x)^i$	$S = \frac{1}{2} \sum_{i=0}^n y_i$	$ y_n < \varepsilon$ $S \approx \frac{1}{4+x^3}$	$x = 0.6;$ $\varepsilon = 1.0e - 07$
23	$y_i = (-1)^{i+1} \frac{2^{4i-3} x^{2i}}{(2i)!}$	$S = \sum_{i=1}^n y_i$	$ y_n < \varepsilon$ $S \approx \sin^2 x \cdot \cos^2 x$	$x = \frac{\pi}{6};$ $\varepsilon = 5.0e - 05$
24	$y_i = (-1)^i \frac{x^i}{i(i-1)}$	$S = x + \sum_{i=2}^n y_i$	$ y_n < \varepsilon$ $S \approx (1+x) \ln(1+x)$	$x = -0.8;$ $\varepsilon = 2.0e - 06$
25	$y_i = (-1)^i \frac{(2x)^{2i}}{(2i)!}$	$S = \sum_{i=1}^n y_i$	$ y_n < \varepsilon$ $S \approx 2(\cos^2 x - 1)$	$x = 0.35;$ $\varepsilon = 5.0e - 08$
26	$y_i = \frac{2x^i}{i!}$	$S = \sum_{i=0}^n y_i$	$ y_n \leq \varepsilon$ $S \approx 2e^x$	$x = 0.9;$ $\varepsilon = 1.0e - 07$



5.6. Лабораторна робота № 6

Тема: розрахунок функцій з декількома одночасно змінюючимися аргументами.

Мета роботи: закріплення навиків програмування при розрахунку функцій з декількома одночасно змінюючимися аргументами.

5.6.1. Завдання

Розробити алгоритм і скласти програму обчислення функції, наведеної в табл.5.4, з декількома одночасно змінюючимися аргументами. Виведення результатів здійснити в наступному вигляді:

РЕЗУЛЬТАТИ ОБЧИСЛЕНЬ			
x	y	z	f
...
...

5.6.2. Загальні відомості по виконанню роботи

Обчислення значень функції можна виконувати за допомогою будь-якого з трьох операторів циклу. Проте в справжній роботі необхідно використовувати оператора циклу **for** з операцією "кома".

При виведенні результатів слід користуватися маніпуляторами `setw()`, `setprecision()`, `setiosflags()` та ін.

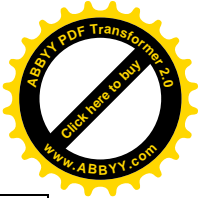
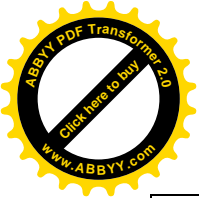
Як параметр циклу слід вибирати аргумент, область зміни якого визначена.

5.6.3. Контрольні питання та завдання

- 1) Які особливості обчислення функції, у якої одночасно змінюється декілька аргументів? Чому область зміни задається тільки для одного аргументу?
- 2) Яка суть операцій "кома" у циклі **for**?
- 3) Запишіть цикл для обчислення функції

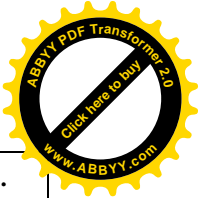
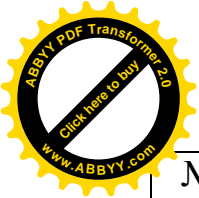
$$R = \frac{\sin x}{x} + \frac{\cos y}{y} + \frac{\operatorname{tg} z}{z}$$

за допомогою операторів **for**, **while** и **do...while**, якщо x , y та z змінюються одночасно, причому x змінюється від X_H до X_K з кроком H_x , y змінюється від Y_H з кроком H_y , z змінюється від Z_H з кроком H_z .

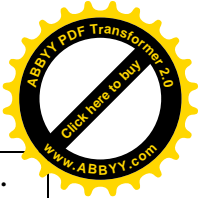
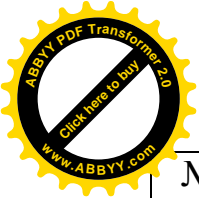


Таблиця 5.4 – Варіанти завдань до лабораторної роботи 6

№ вар.	Функція	x	y	z	Поч. дані
1	$f = \frac{x^2 \sin 2y}{k} - \sqrt{e^x}$	$X_H = 1.2$ $H_x = 0.1$	$Y_H = 5.7;$ $H_y = 0.8$ $Y_K = 12$	–	$k = 3.7$
2	$f = \frac{\operatorname{tg} x \cdot y^{\sqrt{x}} - (z^2 - 1)}{x + z}$	$X_H = 0.7$ $H_x = 0.1$ $X_K = 1.8$	$Y_H = 3.2$ $H_y = 1$	$Z_H = -0.5$ $H_z = 0.2$	
3	$f = \frac{\ln^2(x + y) - a(x^2 + b^2)}{e^z}$	$X_H = 4.8$ $H_x = 1$	$Y_H = -3.1$ $H_y = 0.2$	$Z_H = 1.2$ $H_z = 0.3$ $Z_K = 7.8$	$a = 3.2$ $b = 1.3$
4	$f = \frac{y \sin x - z \cos y}{\operatorname{tg} x} + \ln z$	$X_H = 0.3$ $H_x = 0.1$ $X_K = 3.1$	$Y_H = 1.8$ $H_y = 0.5$	$Z_H = 5.7$ $H_z = 0.4$	
5	$f = \frac{\cos(z - 1)}{z(e^{1.5x} + y^2)} - \sin x$	$X_H = 0.3$ $H_x = 0.2$	$Y_H = -1$ $H_y = 0.1$	$Z_H = -5$ $H_z = 0.4$ $Z_K = 1.2$	
6	$f = \sqrt{ x^2 + e^{2x} } - \sqrt{\sin^2(y - z)}$	$X_H = 1.5$ $H_x = 0.3$ $X_K = 7.2$	$Y_H = 0.2;$ $H_y = 1.2$	$Z_H = 4.2$ $H_z = 1$	
7	$f = \operatorname{arctg}(xy) + z \frac{\sin(\pi y)}{\cos(\pi z)}$	$X_H = 4.2$ $H_x = 1.3$	$Y_H = 2.5;$ $H_y = 0.5$ $Y_K = 10.5$	$Z_H = -7.2$ $H_z = 0.8$	
8	$f = \frac{0.3e^x \sin^3(y + z)}{\sqrt{ 2x - z }}$	$X_H = 1.2;$ $H_x = 0.5$	$Y_H = -2;$ $H_y = 1$	$Z_H = 3.8;$ $H_z = 0.7;$ $Z_K = 18.2$	
9	$f = \lg^2(2x^3) + \cos^2(y + z)$	$X_H = 1.2;$ $H_x = 0.3$	$Y_H = -2;$ $H_y = 0.75$ $Y_K = 13$	$Z_H = 5;$ $H_z = 0.3$	
10	$f = \frac{4x^2 + \sin^2\left(\frac{x}{y}\right)}{\sqrt[3]{(z - 1)^2}}$	$X_H = 3.8;$ $H_x = 0.4;$ $X_K = 9$	$Y_H = -1.5$ $H_y = 0.1$	$Z_H = 2.8;$ $H_z = 0.5$	
11	$f = (e^{-y} + \sqrt{x}) \cdot \operatorname{arctg}(x + y)$	$X_H = 1.8;$ $H_x = 0.2$	$Y_H = -3;$ $H_y = 0.5;$ $Y_K = 8$	–	



№ вар.	Функція	x	y	z	Поч. дані
12	$f = \frac{1}{ze^y} - \sin\left(\frac{\pi}{y}\right)\ln z$	-	$Y_H = 0.7;$ $H_y = 0.1$	$Z_H = 0.3;$ $H_z = 0.6;$ $Z_K = 12.2$	
13	$f = \frac{\cos(x + \ln y)}{\sqrt{(3 - 3.7z)^2}}$	$X_H = -3;$ $H_x = 1;$ $X_K = 18$	$Y_H = -1.5$ $H_y = 1.6$	$Z_H = 0.7;$ $H_z = 0.8$	
14	$f = \alpha x^2 + \beta \sin x - \gamma^{x+z}$	$X_H = 1.9;$ $H_x = 0.1$	-	$Z_H = -4.2$ $H_z = 0.6$ $Z_K = 12.2$	$\alpha = 1.2$ $\beta = 3.7$ $\gamma = -0.8$
15	$f = \frac{\ln x}{e^{-y}} + \frac{\sqrt[3]{\cos^2(\alpha + x)}}{\operatorname{tg} z}$	$X_H = 4;$ $H_x = 1;$ $X_K = 25$	$Y_H = -2;$ $H_y = 0.5$	$Z_H = 2.3$ $H_z = 0.5$	$a = 4.2$
16	$f = (x^2 + y^3 + z^2)\sqrt[5]{z^2}$	$X_H = -4.2$ $H_x = 0.8$	$Y_H = 0.5;$ $H_y = 0.2;$ $Y_K = 4.3$	$Z_H = -3;$ $H_z = 1.2$	
17	$f = x^3 \ln^2 z + 1 + e^z$	$X_H = 0.7;$ $H_x = 0.3;$ $X_K = 7.8$	-	$Z_H = 1.2;$ $H_z = 0.6$	
18	$f = \frac{\sqrt{y^2 + 2 \sin^2(xz)}}{x^y}$	$X_H = 1.2;$ $H_x = 0.3;$ $X_K = 8.9$	$Y_H = 2;$ $H_y = 1$	$Z_H = 0.7;$ $H_z = 0.5$	
19	$f = \frac{\sin x}{1 + \operatorname{tg}^2 y} - \sqrt{z^2 + b}$	$X_H = 5.3;$ $H_x = 1.2$	$Y_H = 0.5;$ $H_y = 0.1;$ $Y_K = 2.3$	$Z_H = 1.2;$ $H_z = 0.5$	$b = 12.6$
20	$f = \frac{a \ln x}{y} + b \cos(z - 1)$	$X_H = 5.2;$ $H_x = 1$	$Y_H = 1.5;$ $H_y = 0.4$	$Z_H = 2.8;$ $H_z = 0.6;$ $Z_K = 15.5$	$a = 3.2;$ $b = 4.8$
21	$f = \frac{\operatorname{lg} z + \sqrt{e^x}}{\cos x - \sin y}$	$X_H = -2;$ $H_x = 0.5;$ $X_K = 17$	$Y_H = 4.2;$ $H_y = 0.1$	$Z_H = 6.5;$ $H_z = 1.2$	
22	$f = \ln \operatorname{tg} \frac{x-y}{z} \cdot \sin \frac{x}{2} \cdot \cos \frac{y}{2}$	$X_H = 0.5;$ $H_x = 0.7$	$Y_H = 1.7;$ $H_y = -0.5$ $Y_K = -8.2$	$Z_H = 4.2;$ $H_z = 0.8$	



№ вар.	Функція	x	y	z	Поч. дані
23	$f = x^z - \frac{z^3}{2 \lg(z-x)}$	$X_H = 3.8;$ $H_x = 1.5$	-	$Z_H = 15.2$ $H_z = 2;$ $Z_K = 43$	
24	$f = \frac{z^{4-x} - \sin 2y}{\operatorname{tg} z}$	$X_H = -1.5$ $H_x = 0.8;$ $X_K = 17$	$Y_H = 2.7;$ $H_y = 0.4$	$Z_H = 3.6;$ $H_z = 1.2$	
25	$f = \frac{y \operatorname{ctg}(z-a)}{\operatorname{tg} z} - zy^{z+a}$	-	$Y_H = 1.1;$ $H_y = 0.1;$ $Y_K = 3.2$	$Z_H = 4;$ $H_z = 0.4$	$a = -2$
26	$f = \sqrt{\frac{x^2 + e^{2y}}{\sin(x+z)}}$	$X_H = 3.5;$ $H_x = 0.5$	$Y_H = 0.1;$ $H_y = 0.9;$ $Y_K = 19$	$Z_H = 4.9;$ $H_z = 1.3$	



5.7. Лабораторна робота № 7

Тема: програмування з використанням масивів і зовнішніх файлів.

Мета роботи: закріплення навиків програмування з використанням масивів і зовнішніх файлів.

5.7.1. Завдання

Розробити алгоритм і скласти програму утворення масиву $Z[n]$ ($n = 13$) за допомогою функції $z_i(x_i)$, наведеної в табл.5.5, де x_i – елементи заданого масиву $X[n]$:

i	1	2	3	4	5	6	7	8	9	10	11	12	13
x_i	22	-3.6	1.5	7.6	-0.2	25	4.8	0.15	-2.7	-12	3.4	0.28	5.2

Введення масиву $X[n]$ передбачити із зовнішнього файлу.

Обчислити середнє арифметичне

$$M = \frac{1}{n} \sum_{i=1}^n z_i$$

і середнє квадратичне

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (z_i - M)^2}$$

для елементів створеного масиву $Z[n]$.

Створити масив $A[n]$ з елементів, що обчислюються за формулою

$$a_i = x_i \cdot z_i$$

і визначити кількість k його позитивних елементів, а також твір P його елементів, значення яких розташовані в діапазоні 0.5...5.

Результати вивести в зовнішній файл в наступному вигляді:

ОПЕРАЦІЇ З МАСИВАМИ

ПОЧАТКОВІ ДАНІ

$$X[1] = 22$$

$$X[2] = -3.6$$

...

$$X[13] = 5.2$$

РЕЗУЛЬТАТИ ОБЧИСЛЕНЬ

$$1. X[1] = 22 \quad Z[1] = \dots$$

$$2. X[2] = -3.6 \quad Z[2] = \dots$$

.....

$$13. X[13] = 5.2 \quad Z[13] = \dots$$

СЕРЕДНЄ АРИФМЕТИЧНЕ

$$M = \dots$$

СЕРЕДНЄ КВАДРАТИЧНЕ

$$\sigma = \dots$$



МАСИВ А
 A[1] = ...
 A[2] = ...

 A[13] = ...

КІЛЬКІСТЬ ПОЗИТИВНИХ ЕЛЕМЕНТІВ МАСИВУ А
 k = ...
 ТВІР ЕЛЕМЕНТІВ МАСИВУ А В ДІАПАЗОНІ (0.5...5)
 P = ...

5.7.2. Загальні відомості та рекомендації по програмуванню

Масив – це складний (агрегатний) тип даних, що є сукупністю однотипних елементів. Масив є зручною формою зберігання та обробки даних.

При розробці алгоритму обчислень раціональнішим буде об'єднання в одному циклі **for** операторів введення даних, обчислення елементів масиву $Z[n]$ і знаходження їх суми, а також утворення масиву $A[n]$, визначення кількості k позитивних та негативних P елементів заданого діапазону.

Слід пам'ятати, що *індексація елементів масиву в C++ починається з нуля*, останній елемент має індекс $n - 1$ (n – загальна кількість елементів масиву). З урахуванням цього заголовок циклу слід писати в наступному вигляді:

```
for (i = 0; i < n; i++)
```

Тут i – поточне значення індексу елементу масиву.

У C++ операції здійснюються тільки над елементами масиву.

При великій кількості даних, що вводяться або виводяться, зручно їх розташовувати в зовнішніх файлах (у цій роботі наказує їх використання). У разі застосування зовнішніх файлів слід передбачити відповідну директиву препроцесору для під'єднання заголовного файлу `<fstream>`. У даній роботі слід застосовувати файли початкових даних та результатів обчислення текстового типу.

Файл початкових даних формується за такою ж технологією, що і файл, що містить текст програми. У даній роботі він повинен складатися з 13 чисел. Числа необхідно розташовувати, починаючи з першої позиції, відокремлюючи один від одного пропуском. Як і будь-які інші файли, цей файл повинен мати своє ім'я і розширення *.txt. Наприклад, `pd.txt` (ім'я можна розуміти як скорочення від словосполучення "початкові дані"). Призначення кожного файлу повинне бути строго визначене. Файл може бути використаний або тільки для читання (введення з файлу), або тільки для запису (вивід у файл) даних. Використання файлу одночасно в двох якостях неможливе. У справжній роботі повинні бути передбачені два зовнішні файли: один (з початковими даними) - для введення, інший - для виведення даних.

При написанні оператора виклику функції відкриття файлу для введення слід передбачити специфікацію *nocreate* (не створювати порожній файл, якщо не знайдений файл, з вказаним ім'ям):

```
fr.open("id.txt", ios::in || ios::nocreate);
```



Тут fr – ім'я задалегідь описаної файлової змінної.

Не слід забувати, що після завершення роботи з файлом його необхідно закрити, наприклад:

```
fr.close( );
```

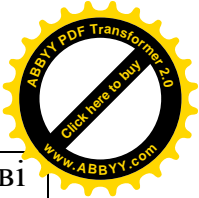
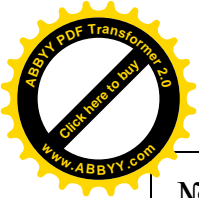
При виведенні даних їх слід форматувати відповідно до вимог завдання, використовуючи маніпулятори мови C++.

5.7.3. Контрольні питання та завдання

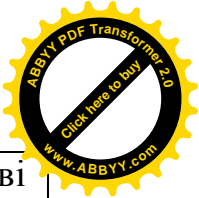
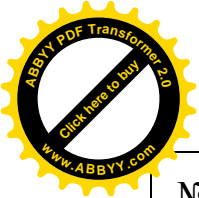
- 1) Як проводиться опис масиву?
- 2) Чим відрізняється ідентифікатор масиву від ідентифікатора його окремого елемента?
- 3) Чи може розмір масиву і індекс його елемента представлятися виразом?
- 4) Перерахуйте способи введення масиву. Запишіть оператора ініціалізації заданого масиву $X[n]$.
- 5) Яка роль змінної файлового типу в програмі?
- 6) Для чого необхідне закриття файлу? До яких наслідків може призвести відсутність в програмі оператора закриття файлу для запису?
- 7) Напишіть фрагмент програми, що передбачає закриття файлу для запису з подальшим його відкриттям для дозапису нових даних.

Таблиця 5.5 – Варіанти завдань до лабораторних робіт 7, 12

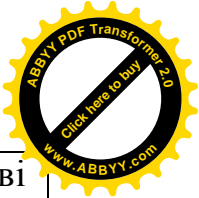
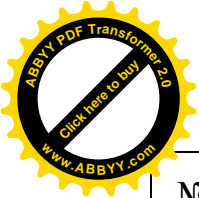
№ варіанту	Вираз	Початкові дані
1	$z_i = \begin{cases} \frac{\sqrt{x_i} \sin(\pi x_i)}{x_i + e^{x_i}}, & x_i > 1.5 \\ x_i + e^{x_i}, & x_i \leq 1.5 \end{cases}$	
2	$z_i = \begin{cases} y_i - 1.5ax_i \frac{y_i^2}{y_i + 1}, & y_i > 1 \\ 2.5a \cdot \cos y_i, & y_i \leq 1 \\ 2 \sin(\cos x_i), & y_i < -1 \end{cases}$ <p>де $y_i = \sqrt{ x_i } + ax_i - \frac{x_i^2}{x_i + 1}$</p>	$a = 0.2$
3	$z_i = \begin{cases} 1 - x_i \cos x_i , & y_i > 0.2 \\ \sqrt{1 + x_i^2}, & y_i \leq 0.2 \end{cases}$ <p>де $y_i = \sin^2(\pi x_i) - x_i$</p>	



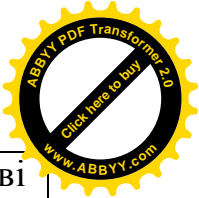
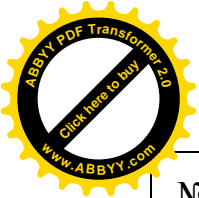
№ варіанту	Вираз	Початкові дані
4	$z_i = y_i + \frac{bx_i}{2 + \cos x_i},$ $\text{де } y_i = \begin{cases} 1 + e^{-t_i}, & t_i > 3 \\ t_i + \cos t_i, & -1 \leq t_i \leq 3 \\ t_i + \sqrt{ t_i }, & t_i < -3 \end{cases}$ $t_i = \frac{3b}{x_i} + \frac{x_i}{x_i^2 - 1}$	$b = 0.5$
5	$z_i = \begin{cases} \ln x_i + 1, & y_i > 0.25 \\ \sqrt{i} \cdot x_i / 2, & y_i \leq 0.25 \end{cases}$ $\text{де } y_i = \frac{1}{2} \sin(\pi x_i)$	
6	$z_i = \begin{cases} \frac{1}{\frac{c}{x_i}}, & x_i > 0.5 \\ \sqrt{x_i^5 e^{x_i} + c}, & 0 < x_i < 0.5 \\ e^{x_i} + c\sqrt{x_i^2 + 1}, & x_i \leq 0 \end{cases}$	$c = 1.5$
7	$z_i = \begin{cases} 0, & y_i < 0 \\ \frac{y_i + ax_i}{a(y_i - ax_i)}, & 0 \leq y_i \leq 5 \\ \arctg(2y_i), & y_i > 5 \end{cases}$ $\text{де } y_i = 2a - \frac{x_i^2}{3a + ax_i}$	$a = 0.1$
8	$z_i = \begin{cases} \frac{1.5x_i^2}{2a} + b \sin(\pi x_i) , & x_i \geq 0 \\ 6.5x_i - \frac{1}{2}\sqrt{x_i^2 + 1}, & x_i < 0 \end{cases}$	$a = -3.2;$ $b = 4.1$
9	$z_i = \frac{3 \sin(\omega_i t + x_i)}{2 + \cos(x_i - \omega_i t)}$ $\text{де } \omega_i = \begin{cases} \pi t - 4tx_i, & x_i < \pi/4 \\ \pi - 4tx_i, & x_i \geq \pi/4 \end{cases}$	$t = 0.5$



№ варіанту	Вираз	Початкові дані
10	$z_i = \begin{cases} c \sqrt{\frac{x_i}{x_i^2 + d}}, & x_i \geq 0 \\ \sqrt[3]{x_i^2 + d}, & x_i < 0 \end{cases}$ <p style="text-align: center;">де $c = \prod_{i=1}^n x_i$</p>	$d = 2$
11	$z_i = \begin{cases} \frac{cx_i}{10} + 1, & x_i < 0 \\ 4x_i^{0.6} + c, & 0 \leq x_i \leq 10 \\ \frac{2c}{x_i^2} - ce^{-x_i}, & x_i > 10 \end{cases}$	$c = 5$
12	$z_i = \begin{cases} y_i + x_i \sqrt{2 - \sin x_i}, & y_i > 0.5 \\ 3 \ln(1 + e^{y_i}), & y_i \leq 0.5 \end{cases}$ <p style="text-align: center;">де $y_i = \frac{\pi}{2} \sin\left(\frac{\pi}{2} x_i\right) - \frac{1}{2} \cos \frac{x_i}{2}$</p>	
13	$z_i = \begin{cases} k + kx_i \sin x_i, & x_i < 0 \\ \frac{k}{2} + \cos^2\left(\frac{\pi}{2} x_i\right), & 0 \leq x_i \leq 10 \\ x_i - \sqrt{\ln x_i }, & x_i > 10 \end{cases}$	$k = 2$
14	$z_i = \sigma_i \left(1 + \sqrt{1 + x_i^2}\right) + (1 - \sigma_i)x_i,$ <p style="text-align: center;">де $\sigma_i = \begin{cases} 1, & y_i \geq 0 \\ -1, & y_i < 0 \end{cases}$</p> <p style="text-align: center;">$y_i = \alpha \sin(\alpha x_i) + 1.5\alpha \cos^2 x_i$</p>	$\alpha = \frac{\pi}{3}$
15	$z_i = \begin{cases} x_i - y_i \sqrt{1 + b \cos y_i}, & y_i < 0 \\ 2\sqrt{1 + e^{by_i}}, & y_i \geq 0 \end{cases}$ <p style="text-align: center;">де $y_i = b \sin(bx_i) - c \cos(cx_i)$</p>	$b = 0.5;$ $c = \frac{1}{3}$
16	$z_i = \begin{cases} \max(R_i, Q_i), & x_i \geq 0 \\ \min(R_i, Q_i), & x_i < 0 \end{cases}$ <p style="text-align: center;">де $R_i = \sqrt{ \sin x_i - \cos x_i }; Q_i = \frac{e^{2x_i}}{1 + \ln 2x_i }$</p>	



№ варіанту	Вираз	Початкові дані
17	$z_i = \begin{cases} \cos(x_i/4), & y_i > x_i \\ 2\operatorname{tg}\sqrt{x_i^2 + 1}, & y_i \leq x_i \end{cases}$ <p>де $y_i = \sin^2 x_i + 5 \cos x_i$</p>	
18	$z_i = \begin{cases} e^{-y_i}, & y_i < -1 \\ \cos y_i, & y_i \leq 1 \\ y_i - \frac{y_i^2}{1.5a(y_i + 1)}, & y_i > 1 \end{cases}$ <p>де $y_i = \sqrt{x_i + 12} + a - \frac{x_i}{x_i + 1}$</p>	$a = 0.2$
19	$z_i = \sqrt{5 D_i + \frac{D_i^4}{2i}},$ <p>де $D_i = \begin{cases} x_i^{\sin^3 x_i} + \ln x_i^2, & 5 \leq x_i \leq 15 \\ e^{x_i} + \sqrt[3]{x_i}, & 0 < x_i < 5 \\ x_i, & x_i \leq 0 \text{ або } x_i > 15 \end{cases}$</p>	
20	$z_i = \begin{cases} x_i, & (x_i \geq 0 \text{ и } F_i \geq 0) \text{ або } (x_i < 0 \text{ и } F_i < 0) \\ F_i, & \text{у інших випадках} \end{cases}$ <p>де $F_i = \begin{cases} \sqrt[5]{\pi x_i + 1/b}, & x_i \geq 0 \\ \frac{e^{-x_i/2}}{3b}, & x_i < 0 \end{cases}$</p>	$b = 0.53$
21	$z_i = \begin{cases} kx_i + 1, & x_i \leq 0 \\ \frac{\sqrt{x_i} \sin(\pi x_i)}{x_i + e^{x_i}}, & 0 < x_i \leq 4.5 \\ 6.5x_i - k\sqrt{x_i + 1}, & x_i > 4.5 \end{cases}$	$k = 0.5$
22	$z_i = \begin{cases} 2 \sin(\cos y_i), & y_i \leq -1 \\ 1 - x_i \cos x_i, & -1 < y_i \leq 5 \\ \cos y_i, & y_i > 5 \end{cases}$ <p>де $y_i = c - \frac{x_i^2}{3c + 0.01x_i}$</p>	$c = 0.2$
23	$z_i = \begin{cases} x_i - \sqrt{\ln x_i}, & x_i > 0 \\ e^{-\frac{x_i}{2}} + 1, & x_i \leq 0 \end{cases}$	



№ варіанту	Вираз	Початкові дані
24	$z_i = \begin{cases} x_i - y_i \sqrt{1 + d \cos y_i}, & y_i < 0 \\ y_i - \frac{y_i^2}{d(y_i + 1)}, & 0 \leq y_i \leq 6.5 \\ y_i - d \cdot y_i^2 (y_i + 1) & y_i > 6.5 \end{cases}$ <p>де $y_i = \sin^2(\pi x_i) - x_i$</p>	$d = 0.3$
25	$z_i = \begin{cases} \frac{1.5x_i^2}{2a} + b \sin(\pi x_i) , & x_i \leq 0 \\ x_i + e^{\frac{1}{x_i}}, & x_i > 0 \end{cases}$	$a = 10;$ $b = 4.7$
26	$z_i = \begin{cases} c\sqrt{x_i^2 + 1} - e^{\frac{2x_i}{a}}, & x_i \leq 6 \\ x_i - \sqrt{\ln x_i}, & x_i > 6 \end{cases}$	$a = 1.7;$ $c = 10$



5.8 Лабораторна робота № 8

Тема: програмування з використанням функцій.

Мета роботи: набути практичних навичок програмування з використанням функцій.

5.8.1. Завдання

Розробити алгоритм і скласти програму виконання дій, вказаних в табл.5.6, з елементами заданої матриці $A[m*n]$ ($m = 4$; $n = 6$):

$$A = \begin{bmatrix} -2 & 1 & 3 & -1 & 4 & 8 \\ 0 & 5 & -8 & 7 & 6 & -3 \\ 5 & 3 & 14 & 0 & 4 & 1 \\ -15 & 12 & 0 & -9 & -8 & 4 \end{bmatrix},$$

використовуючи функцію, призначення якої вказане в табл.5.7. Додатково передбачити функції введення матриці із зовнішнього файлу і виведення результату на екран.

Виведення інформації на екран здійснити в наступному вигляді:

ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ ФУНКЦІЙ

Початкова матриця A

.....
.....

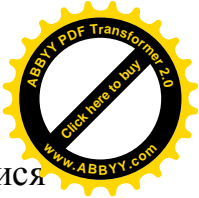
Результати обчислень

.....
.....

5.8.2. Загальні відомості та рекомендації по виконанню роботи

У мові C++ існують інструменти, що дозволяють розділяти програму обчислювального процесу на окремі, логічно обгрунтовані фрагменти. Це зручно робити в тих випадках, коли програма громіздка, має велику кількість операторів. Таку програму складно читати і сприймати. Кожен з фрагментів в цьому випадку отримує своє певне призначення. Такі фрагменти в C++ називають *функціями*, вони мають визначену форму запису (синтаксис).

Таким чином, друга частина програми (перша частина - директиви для препроцесора) може бути представлена в загальному вигляді як сукупність функцій різного призначення, серед яких обов'язково повинна бути головна функція *main()*. Головна функція виконує координуючу роль при реалізації обчислювального процесу. У потрібний момент виконання програми вона "привертає до роботи" ту або іншу функцію для реалізації конкретних дій, які "уміє" робити тільки та функція,що викликається. Така структурна організація програми робить її більш зрозумілою та читабельною.



У загальному випадку тексти створених функцій можуть розміщуватися як перед головною функцією *main()*, так і після неї. Ці функції можуть викликатися головною функцією для виконання фрагмента обчислень, по яких вони спеціалізуються, але також і викликати один одного.

Якщо вирішено розташовувати тексти створених функцій перед головною, то в тих випадках, коли передбачається виклик однієї функції інший, слідує текст функції, що *викликається*, розташовувати раніше тексту функції, що *викликає*. Якщо таких взаємних викликів функцій немає, то і взаємне розташування їх текстів може бути будь-яке. Ця форма розміщення функцій зручна тоді, коли створені функції викликаються тільки головною функцією *main()* і не викликають одна одну. Інакше слід напружувати увагу на їх правильне розміщення, що створює додаткові труднощі. При таких обставинах зручно користуватися іншим способом.

Суть його полягає в тому, що перед головною функцією *main()* записуються тільки заголовки створених функцій, що завершуються крапкою з комою (;). Такі заголовки називаються *прототипи*, а повні тексти функцій (разом із заголовками), звані *визначення*, записуються після головної функції в будь-якій послідовності. В цьому випадку не треба думати, як розташовувати тексти функцій.

Слід пам'ятати, що всі змінні типу масив у функціях оброблюються *по посиланню*, тобто функції використовують не копії змінних, а їх *оригінали*, змінюючи при цьому початкове значення елементів масивів.

Необхідно звертати особливу увагу на запис операторів виклику функцій в частині представлення списку аргументів, який повинен точно відповідати списку параметрів.

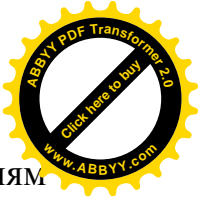
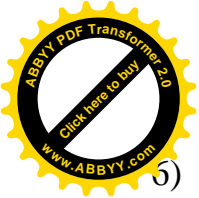
Масиви, використовувані як аргументи функцій, перед її викликом повинні бути заздалегідь описані у функції, що викликає.

Для зручності розміри матриці можна представити за допомогою директиви *define*, а саму матрицю описати як глобальну змінну, помістивши оператор її опису перед текстами функцій, відразу після директив.

При розробці програми особливу увагу слід звернути на суть операцій, що висловлюються у варіанті завдання. Від цього залежить вид функції (функція, що повертає результат по *посиланню* або за *значенням*), особливості її запису і виклику.

5.8.3. Контрольні питання та завдання

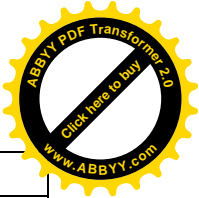
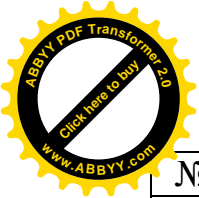
- 1) Яка доцільність використання функцій?
- 2) Визначте місце розташування функції, що викликається, по відношенню до головної (що викликає).
- 3) У чому відмінність функцій, що повертають результат за значенням, від функцій, що повертають результат по посиланню?
- 4) Яка особливість виклику функції, що повертає результат за значенням, і функції, що повертає результат по посиланню?
- 5) У яких випадках функції можуть представлятися без списку параметрів?



5) Складіть програму введення і виведення матриці $A[m*n]$ з використанням функції введення із зовнішнього файлу і функції виводу на екран.

Таблиця 5.6 – Варіанти завдань для лабораторних робіт 8, 15

№ варіанту	Завдання
1 – 4	Побудувати вектор B , елементи котрого рівні:
1	• максимальним елементам рядків,
2	• максимальним елементам стовпців,
3	• мінімальним елементам рядків,
4	• мінімальним елементам стовпців, початкової матриці A .
5 – 6	Поміняти місцями максимальні і мінімальні елементи:
5	• кожного рядка,
6	• кожного стовпця, початкової матриці A .
7 – 10	Упорядкувати в початковій матриці A елементи:
7	• рядків у порядку зростання,
8	• стовпців у порядку зростання,
9	• рядків к порядку убутання,
10	• стовпців у порядку убутання.
11 – 12	Визначити у початковій матриці A :
11	• максимальні елементи рядків та їх адреси,
12	• мінімальні елементи стовпців та їх адреси.
13 – 14	Замінити нульові елементи початкової матриці A максимальними елементами:
13	• рядків,
14	• стовпців, у яких ці елементи розташовані.
15 – 16	Замінити негативні елементи початкової матриці A мінімальними елементами:
15	• рядків,
16	• стовпців, у яких ці елементи розташовані.
17 – 18	Замінити в початковій матриці A нулями максимальні елементи:
17	• кожного рядка,
18	• кожного стовпця.
19 – 20	У початковій матриці A упорядкувати в порядку убутання елементи:
19	• рядків,
20	• стовпців, котрі починаються з позитивних елементів.
21 – 22	Упорядкувати у початковій матриці A у порядку зростання



№ варіанту	Завдання
21 22	елементи: <ul style="list-style-type: none">• парних рядків,• непарних стовпців.
23 – 24 23 24	У початковій матриці A замінити нулями мінімальні елементи: <ul style="list-style-type: none">• непарних рядків,• парних стовпців.
25 – 26 25 26	У початковій матриці A визначити суму: <ul style="list-style-type: none">• максимальних елементів стовпців,• мінімальних елементів стовпців.

Таблиця 5.7 – Призначення функції у лабораторних роботах 8, 15

№ варіанту	Призначення функції
1, 2, 13, 14 3, 4, 15, 16 5, 6 11, 25 12, 26 17, 18 23, 24	Визначити відповідно з варіантом: <ul style="list-style-type: none">• максимальний елемент,• мінімальний елемент,• індекс максимального та мінімального елементів,• максимальний елемент та його індекс,• мінімальний елемент та його індекс,• індекс максимального елемента,• індекс мінімального елемента.
7, 8, 21, 22 9, 10, 19, 20	Упорядкувати відповідно з варіантом елементи матриці: <ul style="list-style-type: none">• у порядку зростання,• у порядку убуття.



5.9. Лабораторна робота № 9

Тема: програмування з використанням операцій над комплексними числами.

Мета роботи набути практичних навичок програмування з використанням операцій над комплексними числами ті бібліотеки стандартних функцій обробки комплексних чисел.

5.9.1. Завдання

Обчислити коріння x_1 та x_2 квадратного рівняння

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

для заданих значень коефіцієнтів a , b і c , та визначити значення виразу, приведенного в табл.5.8. Правильність обчислення коріння рівняння перевірити по теоремі Вієта.

На екран вивести значення коріння рівняння і результат обчислення виразу в наступній формі:

ОПЕРАЦІЇ НАД КОМПЛЕКСНИМИ ЧИСЛАМИ

ПОЧАТКОВІ ДАНІ

a = ... b = ... c = ...

РЕЗУЛЬТАТИ ОБЧИСЛЕНЬ

Корені квадратного рівняння

$x_1 = 0.5 + 3i$ $x_2 = 0.5 - 3i$

Значення виразу

W = ...

Модуль W = ... Аргумент W = ...

Перевірка обчислення коренів

b = ... c = ...

5.9.2. Загальні відомості та рекомендації по виконанню роботи

Комплексне число – це число вигляду $\alpha + \beta i$, що зображується на числовій площині відрізком прямої, з'єднуючої початок координат з крапкою з координатами (α, β) (мал.5.1).

При цьому α називають *речовою* (дійсною) частиною і вважають вісь α віссю абсцис та позначають символом $+$.

βi – *уявна* частина комплексного числа, вісь β вважають віссю ординат та позначають символом i або j . Множник $i = \sqrt{-1}$

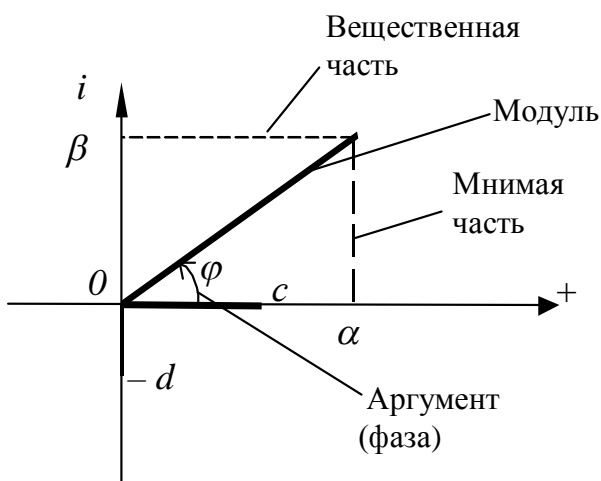
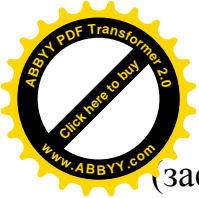


Рисунок 5.1.



(застосовують також символ j) називають уявною одиницею.

Дійсні числа є окремим випадком комплексних чисел з уявною частиною, рівною нулю, наприклад, в загальному випадку,

$$c + 0i.$$

Дійсні числа, як відомо, зображуються на числовій осі відрізком, проведеним з початку координат у бік позитивної або негативної нескінченності, і що має довжину, рівну c (мал.5.1).

Разом з дійсними (чисто речовими) числами, розрізняють і чисто уявні числа як інший окремий випадок комплексного числа з речовою частиною, рівною нулю, наприклад, $0 - di$ (мал.5.1).

Знак, записуваний між дійсною і уявною частинами комплексного числа, не є знаком операції віднімання, а є знаком уявної частини, записаної поряд з речовою частиною комплексного числа. Прикладом комплексних чисел можуть бути:

$$5 + 2.8i, 7.2 - 12i, -1 + 0i, 0 + 5.5i.$$

Останні два числа є відповідно чисто дійсне і чисто уявне число і можуть бути записані, як -1 та $5.5i$. Розглянута форма запису комплексного числа називається *алгебраїчною*.

У математиці і в теоретичній електротехніці також широко використовується *показова* форма запису комплексного числа у вигляді $Ae^{j\varphi}$ (мал.5.1). Тут $A = \sqrt{\alpha^2 + \beta^2}$ називають *модулем* комплексного числа (може застосовуватися також термін *амплітуда*), а кут $\varphi = \arctg \frac{\beta}{\alpha}$ називають *аргументом* або *фазою* комплексного числа.

Як правило, у формі алгебри зручно проводити складання і віднімання комплексних чисел, а в показовій формі - множення та ділення.

При негативному дискримінанті коріння квадратного рівняння буде комплексними числами, причому комплексно-зв'язаними (розрізняються тільки знаком уявної частини). Корінь з негативного числа можна визначити, як чисто уявне число, наприклад:

$$\sqrt{-d} = \sqrt{-1 \cdot d} = \sqrt{d} \cdot \sqrt{-1} = \sqrt{d} i.$$

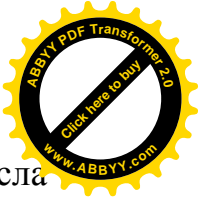
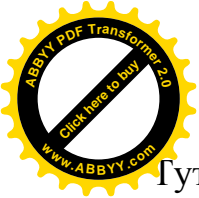
У мові C++ передбачена бібліотека стандартних функцій обробки комплексних чисел, яка міститься в заголовному файлі `complex.h` (додаток 2). Для її використання слід передбачити відповідну директиву препроцесору (підключення класу `complex`).

При написанні програми комплексну змінну, як і будь-яку іншу, слід описати. Тип комплексних даних має назву `complex`, а опис, наприклад, комплексної змінної h матиме вигляд:

```
complex h;
```

Привласнення комплексній змінній певного значення виконується таким чином:

```
h = (-1, 3);
```



Гут комплексній змінній h присвоєно значення $-1 + 3i$. Речова частина числа $Re[h] = -1$, уявна частина $Im[h] = 3$, модуль числа $|h| = \sqrt{10}$, аргумент числа $Arg[h] = \arctg(-3) = -1.249$ рад.

Над комплексними змінними можливі ті ж арифметичні операції, що і над дійсними числами. Наприклад, в результаті виконання фрагмента програми:

```
...
complex h1 = (2, -3.5), h2 = (-1, 5);
complex h = h1 + h2;
cout << h;
...
```

змінна h отримає значення $1 + 1.5i$, а на екрані з'явиться повідомлення $(1, 1.5)$

З урахуванням специфіки виведення комплексних чисел необхідно скласти власну функцію виводу на екран значення комплексного числа в алгебраїчній формі.

Основні функції бібліотеки обробки комплексних чисел приведені в додатку 2.

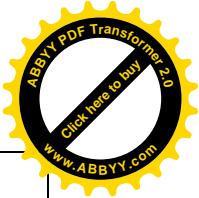
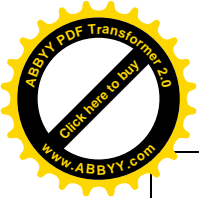
При написанні програми рекомендується комплексні числа, наприклад, $\sqrt{-3}$, $5 + 3i$ та ін., описати константами і при обчисленні виразу використовувати їх ідентифікатори.

5.9.3. Контрольні питання та завдання

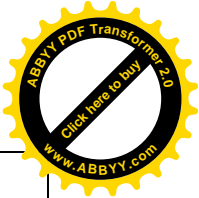
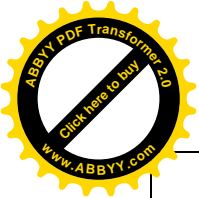
- 1) Запишіть число, речова частина якого рівна 3, а уявна -3 , в алгебраїчній та в показовій формі.
- 2) Як проводиться опис комплексної змінної, комплексної константи? Приведіть приклад.
- 3) Які математичні операції можуть проводитися над комплексними змінними?
- 4) Перерахуйте і опишіть стандартні функції бібліотеки обробки комплексних чисел, необхідні для написання програми згідно Вашому варіанту завдання.

Таблиця 5.8 – Варіанти завдань до лабораторної роботи 9

№ варіанту	a	b	c	Вираз
1	1	0.3	0.2	$P = \sqrt{-3} \frac{x_1 - x_2}{x_1 x_2}$
2	0.5	3	1	$q = e^2 + \left \frac{x_1}{2x_2} \right i$
3	1.2	1.8	1.5	$h = x_1(3.2 + 5i) - \frac{1}{x_2^2}$
4	15	2	3	$f = \frac{Re[x_2]}{x_1^3(1+i)}$



№ варіанту	a	b	c	Вираз
5	5.2	7.8	4.6	$Z = \lg x_1 + \frac{2-6i}{x_2}$
6	17	16	21	$y = \sqrt{-5 \cdot \operatorname{Im}[x_2]} - \frac{x_1}{5x_2}$
7	27	32	31	$W = \frac{5+3i}{x_1+2x_2} - \sqrt{ x_1x_2 }$
8	5.2	3.1	7.1	$S = \frac{(x_1+3x_2)(3x_1-x_2)}{(x_1-x_2)i}$
9	8	8	4	$P = \frac{\cos x_1 + i \sin x_2 }{x_1x_2}$
10	6.4	9.2	7.3	$Q = \frac{\sqrt{ x_1 } + x_2}{-3(x_1+x_2)}$
11	4.2	5.8	3.8	$R = \frac{i \cos x_1 + \sqrt{ x_2 }}{2x_1}$
12	3.2	12.4	21.8	$F = \frac{2x_1}{x_2} + \operatorname{Re}[x_1] - \operatorname{Im}[x_2]$
13	7.3	10.7	6.2	$Z = \sin\left \frac{3x_1 - \sqrt{ x_2 }}{2x_1}\right + ix_2$
14	4.2	11.4	11.2	$\varphi = \ln\left \frac{x_1}{x_2}\right + i\sqrt{ x_2 }$
15	6.7	8.9	4.9	$\mu = \sqrt{-5} \frac{2x_1+x_2}{i x_2 }$
16	18.3	12.6	11.2	$R = \frac{\operatorname{Re}[x_1] + i \operatorname{Re}[x_2]}{x_1x_2}$
17	5.7	3.2	11.3	$P = \frac{ix_1 + \sqrt{ x_2 }}{\operatorname{Re}[x_1]}$
18	31.8	22.7	42	$S = \frac{x_1+x_2}{x_1x_2} + \sqrt{ x_1 \cdot x_2 }$
19	11.89	9.3	12.9	$T = \sin(\operatorname{Re}[x_1]) + \frac{x_1}{ix_2}$
20	6.7	5.3	17.4	$g = \frac{x_1 + \sqrt{ x_2 }}{2 \sin(3 + \operatorname{Re}[x_1])}$



№ варіанту	<i>a</i>	<i>b</i>	<i>c</i>	Вираз
21	43.4	21.2	56	$h = \operatorname{Im}[3i + x_1] + \frac{\sqrt{-7}}{ x_2 }$
22	204	101	156	$f = \frac{\sin(\operatorname{Re}[x_1]) + i \cos(\operatorname{Im}[x_1])}{2x_2}$
23	94	72.3	82	$M = \frac{x_1 + x_2}{-5(x_1 - x_2)} + \sqrt{\frac{ x_1 }{ x_2 }}$
24	10.2	12.9	15.7	$L = \sqrt{-13} \frac{ x_1 + x_2 }{\operatorname{Im}[x_1 + x_2]}$
25	24.7	31.3	43.2	$H = i \operatorname{Re}\left[\frac{x_1 + x_2}{x_1 x_2}\right] + \operatorname{Im}\left[\frac{x_2}{x_1}\right]$
26	18.6	5.92	6.8	$Y = \frac{x_1 x_2}{x_1 + x_2} - \operatorname{Im}[x_1 x_2]$

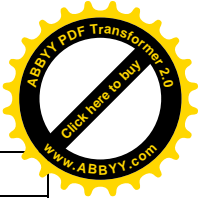
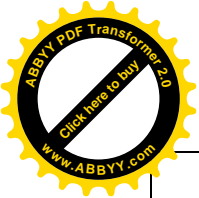


ДОДАТКИ

Додаток 1.

Основні функції стандартної бібліотеки математичних функцій для обробки дійсних чисел (заголовний файл math.h)

Функція	Призначення	Аргументи		Результат	
		Тип	Од. вим.	Тип	Од. вим.
abs(x)	$ x $	int		int	
labs(x)		long int		long int	
fabs(x)		double		double	
fabsl(x)		Long double		long double	
sin(x)	$\sin x$	double	радіани	double	
sinl(x)		Long double	радіани	long double	
cos(x)	$\cos x$	double	радіани	double	
cosl(x)		Long double	радіани	long double	
tan(x)	$\operatorname{tg} x$	double	радіани	double	
tanl(x)		Long double	радіани	long double	
asin(x)	$\arcsin x$	double		double	радіани
asinl(x)		Long double		long double	радіани
acos(x)	$\arccos x$	double		double	радіани
acosl(x)		Long double		long double	радіани
atan(x)	$\operatorname{arctg} x$	double		double	радіани
atanl(x)		Long double		long double	$-\frac{\pi}{2} \dots \frac{\pi}{2}$
atan2(x, y)	$\operatorname{arctg} \frac{x}{y}$	double, double		double	радіани $-\pi \dots \pi$
atan2l(x, y)		long double, long double		long double	
sinh(x)	$\operatorname{sh} x = \frac{e^x - e^{-x}}{2}$	double		double	
sinhl(x)		Long double		long double	
cosh(x)	$\operatorname{ch} x = \frac{e^x + e^{-x}}{2}$	double		double	
coshl(x)		long double		long double	
tanh(x)	$\operatorname{th} x = \frac{\operatorname{sh} x}{\operatorname{ch} x}$	double		double	
tanh1(x)		long double		long double	
ceil(x)	округлення до найближчого більшого	double		double	ціле число
ceil1(x)		long double		long double	
floor(x)	округлення до найближчого меншого	double		double	ціле число
floor1(x)		long double		long double	
log(x)	$\ln x$	double		double	
log1(x)		long double		long double	
log10(x)	$\lg x$	double		double	
log101(x)		long double		long double	
sqrt(x)	\sqrt{x}	double		double	



Функція	Призначення	Аргументи		Результат	
		Тип	Од. вим.	Тип	Од. вим.
sqrtl(x)		long double		long double	
exp(x)	e^x	double		double	
expl(x)		long double		long double	
pow10(x)	10^x	int		double	
pow10l(x)		int		long double	
pow(y, x)	x^y	double, double	ціле число у діапазоні $-\infty \dots +\infty$	double	
powl(y, x)		long double, long double		long double	
hypot(x, y)	$\sqrt{x^2 + y^2}$	double, double		double	
hypotl(x, y)		long double, long double		long double	
fmod(x, y)	залишок від ділення x на y	double, double		double	ціле число
fmodl(x, y)		long double, long double		long double	
modf(x, &xc)	повертає дробову частину числа x	double, double		double	значення цілої частини зберігається у змінній xc
modfl(x, &xc)		long double, long double		long double	
frexp(x, &n)	повертає таке $0.5 \leq m < 1$, що $x = m \cdot 2^n$	double, double		double	Значення показника ступеня зберігається у змінній n
frexpl(x, &n)		long double, long double		long double	
ldexp(x, n)	$x \cdot 2^n$	double, int		double	
ldexpl(x, n)		long double, int		long double	
poly(x, n, c)	$c_n x^n + c_{n-1} x^{n-1} + \dots + c_1 x + c_0$	double, int, double[]	c - масив розмірністю $[n+1]$	double	
polyl(x, n, c)		long double, int, long double		long double	
atof(&s[])	перетворює рядок в число	char	const	double	
_atold(&s[])		char		long double	



Додаток 2.

Основні функції стандартної бібліотеки математичних функцій для обробки комплексних чисел (заголовний файл complex.h)

Функція	Призначення	Тип	
		аргументів	результату
real(z)	$Re [z]$	complex	double
imag(z)	$Im [z]$	complex	double
abs(z)	$ z $	complex	double
arg(z)	$Arg [z]$	complex	double
norm(z)	$ z ^2$	complex	double
conj(z)	комплексно-зв'язане число	complex	complex
sin(z)	$\sin z = \frac{e^{iz} - e^{-iz}}{2i}$	complex	complex
cos(z)	$\cos z = \frac{e^{iz} + e^{-iz}}{2}$	complex	complex
tan(z)		complex	complex
asin(z)	$\arcsin z = -i \cdot \ln \left(iz + \sqrt{1 - z^2} \right)$	complex	complex
acos(z)	$\arccos z = -i \cdot \ln \left(z + i \sqrt{1 - z^2} \right)$	complex	complex
atan(z)	$\arctg z = -0.5i \cdot \ln \left(\frac{1 + iz}{1 - iz} \right)$	complex	complex
sinh(z)	$sh z = \frac{e^z - e^{-z}}{2}$	complex	complex
cosh(z)	$ch z = \frac{e^z + e^{-z}}{2}$	complex	complex
tanh(z)		complex	complex
log(z)	$\ln z = \ln z + i Arg [z]$	complex	complex
log10(z)	$lg z = \ln z / \ln 10$	complex	complex
exp(z)	$e^z = e^{Re [z]} (\cos(Im [z]) + i \sin(Im [z]))$	complex	complex
sqrt(z)	$\sqrt{z} = \sqrt{ z } \cdot \left(\cos \frac{Arg [z]}{2} + i \sin \frac{Arg [z]}{2} \right)$	complex	complex
pow(y, x)	$x^y = e^{y \ln x}$	complex, complex	complex
pow(y, x)		complex, double	complex
pow(x, y)		double, double	complex
polar(m, a)	завдання комплексного числа у показовій формі m^{ia}	double, double	complex



СПИСОК ЛІТЕРАТУРИ

1. Франка П. С++. Учебный курс. – Санкт-Петербург, Москва, Харьков, Минск: «Питер», 1999. – 521 с.
2. Шилдт Г. С++. – 3-е изд., перераб. и доп. – Дюссельдорф, Киев, Москва, Санкт-Петербург: «ВНУ – Санкт-Петербург», 1998. – 683 с.
3. Пол А. Объектно-ориентированное программирование на С++. – 2-е изд. – Москва, Санкт-Петербург, 1999. – 461 с.
4. С/С++. Программирование на языке высокого уровня / Т.А. Павловская. – СПб.: «Питер», 2001. – 464 с.: ил.
5. Страуструп Б. Язык программирования С++ - 3-е изд. – СПб.:»Невский диалект», 1999. – 991с.



Методичні вказівки до лабораторних
робіт по дисципліні «Введення в інформатику»
(для студентів спеціальності 7.0922.03, «Електромеханічні
системи автоматизації та електропривод», частина 1)

Розробили: © Ізмайлов Геннадій Григорович
Прокопенко Олена Василівна
Масло Світлана Володимирівна