

ФАКУЛЬТЕТ АВТОМАТИКИ, ТЕЛЕМЕХАНІКИ ТА ЗВ'ЯЗКУ

Кафедра обчислювальної техніки і систем управління

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ
МОВОЮ QBASIC**

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт

з дисциплін

«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»,

**«ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА
ПРОГРАМУВАННЯ»**

Частина 1

Харків – 2014

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри обчислювальної техніки та

систем управління 28 січня 2014 р., протокол № 6.

Методичні вказівки до виконання лабораторних робіт призначені для студентів за напрямками підготовки бакалаврів: 6.050601 "Теплоенергетика", 6.050503 "Машинобудування", 6.070108 "Залізничні споруди та колійне господарство", 6.060101 "Будівництво" усіх форм навчання

Укладачі:

доценти І.В. Піскачова,
О.В. Головка

Рецензент

проф. С.В. Лістровий

АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ МОВОЮ QBASIC

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт
з дисциплін

*«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»,
«ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА
ПРОГРАМУВАННЯ»*

Частина 1

Відповідальний за випуск Головка О.В.

Редактор Еткало О.О.

Підписано до друку 22.04.14 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 1,00. Тираж 50. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,
61050, Харків-50, майдан Фейербаха, 7.

Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
УКРАЇНСЬКА ДЕРЖАВНА АКАДЕМІЯ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

Алгоритмізація та програмування мовою QBASIC
Методичні вказівки до лабораторних робіт з дисциплін
«Обчислювальна техніка та програмування», «Основи інформа-
ційних технологій та програмування»
Частина 1

Харків, 2014

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри обчислювальної техніки та систем управління 28 січня 2014 р., протокол № 6.

Методичні вказівки до виконання лабораторних робіт призначені для студентів за напрямками підготовки бакалаврів: 6.050601 "Теплоенергетика", 6.050503 "Машинобудування", 6.070108 "Залізничні споруди та колійне господарство", 6.060101 "Будівництво" усіх форм навчання

Укладачі:

доценти І.В. Піскачова,
О.В. Головка

Рецензент

проф. С.В. Лістровий

ЗМІСТ

ВСТУП	4
Лабораторна робота 1. Проектування алгоритмів лінійних обчислювальних процесів. Програмне середовище QBASIC. Програмування мовою QBASIC алгоритмів лінійних обчислювальних процесів. Оператори введення, виведення (INPUT, PRINT)	5
Лабораторна робота 2. Проектування алгоритмів розгалужених обчислювальних процесів. Програмування мовою QBASIC алгоритмів розгалужених обчислювальних процесів з використанням оператора IF <умова> THEN <ДІЯ_1> ELSE <ДІЯ_2>	16
Лабораторна робота 3. Проектування алгоритмів та програмування мовою QBASIC алгоритмів простих арифметичних циклічних обчислювальних процесів	28
Лабораторна робота 4. Програмування ітераційних циклічних процесів з використанням операторів WHILE ... WEND, DO ... LOOP	38
Список літератури	49

ВСТУП

Метою навчального процесу є підготовка фахівців, які володіють навичками постановки математичних задач, розроблення алгоритмів та програмування з використанням мов програмування високого рівня. Вивчаються основи інформатики та інформаційно-обчислювального процесу, алгоритмізація та основи програмування на мовах Qbasic та Visual Basic.

Курс базується на знаннях, отриманих при вивченні інформатики, вищої математики, фізики у середній школі, технікумі або коледжі.

У свою чергу цей курс є базовим для вивчення у подальшому дисципліни “Математичні методи та моделі” та спеціальних дисциплін, за допомогою яких студенти навчаються методів проектування, організації та моделювання систем управління устаткуванням та інших систем і приладів, що потребують здійснення розрахунків з використанням засобів обчислювальної техніки при дослідженнях.

Формування у майбутнього фахівця алгоритмічного мислення, вміння чітко формулювати задачу, здійснювати її декомпозицію та знаходити розв’язок є абсолютно необхідним.

Автори методичних вказівок ставили за мету розвинути у студентів практичні навички в підготовці задач для їх розв’язання з використанням засобів комп’ютерної техніки. При написанні програм використовується мова програмування BASIC, вибір якої зумовлений її простотою і широким використанням.

Лабораторна робота 1

ПРОЕКТУВАННЯ АЛГОРИТМІВ ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ. ПРОГРАМНЕ СЕРЕДОВИЩЕ QBasic. ПРОГРАМУВАННЯ МОВОЮ QBasic АЛГОРИТМІВ ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ. ОПЕРАТОРИ ВВЕДЕННЯ, ВИВЕДЕННЯ (INPUT, PRINT)

Мета роботи: ознайомитися з алгоритмізацією лінійних обчислювальних процесів та програмуванням їх за допомогою мови QBasic.

Завдання до лабораторної роботи

Згідно з варіантом обраним викладачем, побудувати алгоритм та написати програму рішення завдання.

Короткі теоретичні положення

Будь-яка комп'ютерна програма розв'язує певну задачу, виконуючи конкретну послідовність дій. Отже, в основі кожної комп'ютерної програми лежить алгоритм. Алгоритм – упорядкований скінчений набір чітко визначених правил для розв'язування задач за кінцеву кількість кроків [1]. Алгоритми мають такі властивості: дискретність – розділення алгоритму на зручні, зрозумілі частини, можливість виконання яких не викликає сумніву; масовість (типовість) – можливість використання для великої множини вхідних даних та для інших задач того ж класу; результативність – отримання результату завжди; детермінованість – визначеність, однозначність результату при заданих вхідних даних та ін. Для складення алгоритмів найчастіше використовують блок-схеми. *Блок-схема* – це графічна побудова, на якій за допомогою геометричних фігур показані кроки розв'язання задачі. Для складання блок-схем застосовують фігури, зображені на рисунку 1.1. Співвідношення між висотою і шириною фігур задані А і В.

У процесі розроблення алгоритму програміст рисує блок-схему, позначаючи кожну дію одним з показаних на рисунку 1.1 блоків. У середині знака він ставить позначки, що уточнюють дану дію. Усі елементи схеми з'єднуються лініями зі стрілками,

які показують напрямок передачі даних і управління.

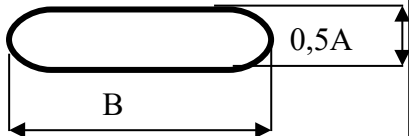
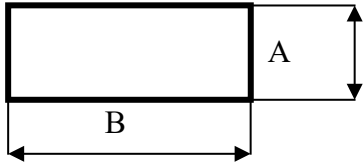
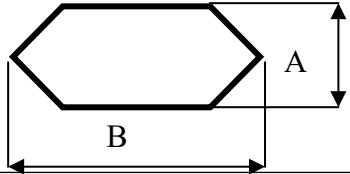
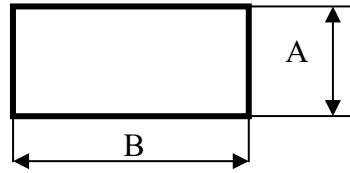
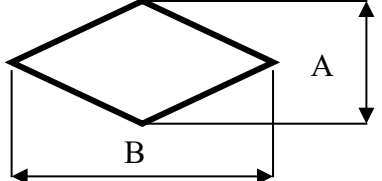
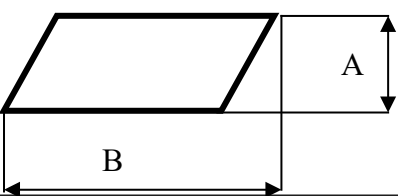
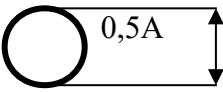
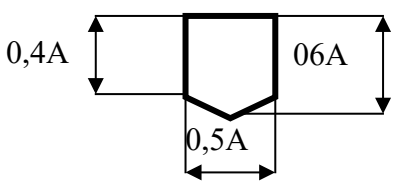
Назва блоку	Графічне позначення блоку	Дії, що виконуються
Початок-зупинка		Початок, кінець, переривання процесу чи обробки виконання програми
Процес		Виконання операції, в результаті якої змінюється значення, форма чи подання даних
Модифікація (заголовок циклу)		Виконання операцій, що змінюють команди, групи команд або програму
Визначений процес (підпрограма)		Використання раніше створених або окремо написаних алгоритмів програм
Рішення		Вибір напрямку виконання алгоритму програми в залежності від деяких змінних умов
Введення / виведення даних		Перетворення даних у форму, придатну для обробки (уведення) чи відображення результатів обробки (виведення)
З'єднувач сторінковий		Розрив лінії потоку інформації
З'єднувач між-сторінковий		Розрив лінії потоку між сторінками

Рисунок 1.1 – Деякі графічні елементи блок-схеми

Лінійний алгоритм

Алгоритм називається **лінійним**, якщо він має N кроків і кожен крок виконується послідовно один за одним від початку до

кінця.

Приклад 1.1

Увести числа А і В. Розрахувати їх середнє геометричне.
Побудувати блок-схему алгоритму рішення завдання.

Розв'язання:

Блок-схема програми показана на рисунку 1.2.

Опис блок-схеми.

- 1 – початок розрахунків;
- 2 – уведення вихідних даних;
- 3 – розрахунки середнього арифметичного;
- 4 – виведення результату на екран;
- 6 – завершення розрахунків.

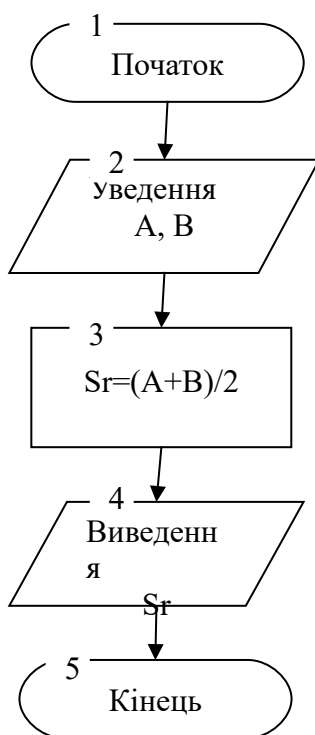


Рисунок 1.2 – Блок-схема для розв'язання прикладу 1.1

Мова програмування QBasic

Мова QBasic як і будь-яка інша мова програмування є засобом спілкування з комп'ютером. Таблиця 1.1 відображає перелік символів, що використовуються в QBasic. Поруч з буквами алфавіт вміщує різні знаки та спеціальні символи. Усі вони можуть бути знайдені на клавіатурі.

В QBasic існує два основних типи даних – числові та текс-

тові.

Такі значення, як 44499, -5.99 є прикладами числових даних, тобто даних, що складаються з цифр та десяткової крапки і, якщо число від'ємне, – знака мінус.

Тексти можуть включати літери, числа та інші знаки в будь-якій комбінації, наприклад: «Запис номер 20». Такий тип даних має текстову інформацію і повинен бути взятий у лапки.

Таблиця 1.1 – Перелік символів, що використовуються в QBasic

Сим-вол	Найменування	Символ	Найменування
A-Z	Великі літери	:	Двокрапка
a-z	Малі літери	+	Знак плюс
0-9	Цифри	-	Знак мінус
	Пропуск	*	Зірочка (знак множення)
!	Знак оклику	/	Нахилена риска
#	Дієз	\	Обернена нахилена риска
%	Знак процента	^	Знак вставки
&	Знак амперсанда	=	Знак рівності
\$	Знак долара	<	Знак менше
«	Подвійні лапки	>	Знак більше
'	Апостроф	()	Ліва і права дужки
;	Крапка з комою	?	Знак питання
.	Крапка	_	Знак підкреслення
,	Кома	~	Тильда

Правила набору математичних виразів мовою QBasic

У QBasic взагалі чотири тригонометричних функції: TAN(м.в), ATN(м.в), COS(м.в), SIN(м.в) наведені у таблиці 1.2.

Інші стандартні тригонометричні функції обчислюються на їх основі за формулами:

$$\text{SEC(м.в.)} = 1 / \text{COS(м.в.)}$$

$$\text{Cosec(м.в.)} = 1 / \text{SIN(м.в.)}$$

$$\text{COTAN(м.в.)} = 1 / \text{TAN(м.в.)}$$

$$\text{ARCSIN(м.в.)} = \text{ATN(м.в. / SQR(-м.в. * м.в. + 1))}$$

$$\text{ARCCOTAN(м.в.)} = 1.570796 - \text{ATN(м.в.)}$$

$$\text{ARCCOS(м.в.)} = \text{ATN(-м.в. / SQR(-м.в. * м.в. + 1))} + 2 * \text{ATN(1)}$$

Таблиця 1.2 – Деякі математичні функції мови QBasic

Функція	Опис функції
ABS(м.в)	Модуль математичного виразу
EXP(м.в)	Основа натурального логарифма в степені заданого математичного виразу або аргументу ($e^{м.в.}$)
LOG(м.в)	Натуральний логарифм ($\ln(м.в)$). Аргумент повинен бути більше нуля
SQR(м.в)	Квадратний корінь математичного виразу. Значення виразу повинно бути більше або дорівнювати нулю
TAN(м.в)	Тангенс математичного виразу. Значення виразу розглядається як кут у радіанах
ATN(м.в)	Арктангенс математичного виразу. Значення виразу розглядається як кут у радіанах
COS(м.в)	Косинус математичного виразу. Значення виразу розглядається як кут у радіанах
SIN(м.в)	Синус математичного виразу. Значення виразу розглядається як кут у радіанах

Особливості робіт із логарифмічними функціями

У практиці поряд з натуральними ($\ln(м.в.)$) доводиться мати справу з десятковими ($\lg(м.в.)$) та логарифмами з довільною основою ($\log_n(м.в.)$). Їх можна виразити через функцію $\text{Log}(м.в.)$ (таблиця 1.3).

Таблиця 1.3 – Використання логарифмів у мові QBasic

Вираз	У програмі QBasic
$y=\ln(x)$	$y=\text{LOG}(x)$
$y=\lg(x)$	$y=\text{LOG}(x)/\text{LOG}(10)$
$y=\log_n(x)$	$y=\text{LOG}(x)/\text{LOG}(n)$

Математичні вирази в будь-якій мові програмування записуються в один рядок за допомогою спеціальних команд. Спеці-

альні команди дають вказівки комп'ютеру при розрахунках обчислювати ту або іншу математичну функцію, наприклад синус або квадратний корінь (таблиця 1.4).

Набір математичних виразів слід виконувати з використанням нижнього регістру клавіатури (малими літерами) без введення пробілів. Це робиться для самоперевірки – після того, як буде набраний рядок з прикладом і натиснута клавіша Enter, у випадку безпомилкового набору, назви функцій автоматично запишуться із великої букви й між арифметичними знаками будуть поставлені пробіли. Якщо ж приклад набраний неправильно, то рядок не зміниться.

Таблиця 1.4 – Правила набору математичних виразів у QBasic

Математичний вираз	Запис у програмі QBasic
$y=x+a$	$y=x+a$
$y=x^n$	$y=x^{\wedge}n$
$y=\frac{x}{n}$	$y=x/n$
$y=\frac{x+2}{x-3}$	$y=(x+2)/(x-3)$
$y=\sqrt{x}$	$y=SQR(x)$
$y=\sqrt[n]{x}$	$y=x^{\wedge}(1/n)$
$y=\sqrt{x+2}\times\cos x$	$y=SQR(x+2)*COS(x)$
$y=\sin x$	$y=SIN(x)$
$y=\sin^2x$	$y=SIN(x)^2$
$y=\sin x^2$	$y=SIN(x^2)$

Деякі оператори мови QBasic

Виведення результату обчислень на екран:

Оператор PRINT має таку структуру: **PRINT** *параметр*, де, наприклад,

PRINT B — оператор

PRINT — ключове слово

B — параметр.

Замість слова **PRINT** можна використовувати знак питання. Після натискування клавіші Enter **QBasic** сам перетворює знак питання в ключове слово **PRINT**.

QBasic не очищує автоматично екран при запуску нової програми, і після виконання різних програм або багаторазового виконання тієї ж самої на екрані виведення результатів виникає безлад. Тому програму треба починати з оператора очищення екрана **CLS**, який стирає всі попередні результати виконання програми. Відзначимо, що оператор **CLS** стирає дані тільки з екрана і не діє на дані, що зберігаються в пам'яті ПК. Як правило, цей оператор розміщується на початку програми – це добра звичка і прекрасний стиль програмування.

Оператор **END** — ще один приклад ключового слова — зупиняє виконання програми. У випадку його відсутності **QBasic** вважає, що програма закінчується після останнього рядка, але **END** досить корисний при необхідності припинити виконання програми.

Оператор **REM** визначає рядок, який він починає, як рядок коментарю. Він має таку форму:

REM *текст*,

де *текст* – будь-яка послідовність символів.

Ви можете будь-яку програму починати з коментарю (заголовка), який можна розміщати також будь-де в програмі.

Знак апострофа ' також починає коментар, але його можна використовувати не тільки з початку рядка, але і після операторів.

Розміщення кількох операторів у рядку

Поки ваші програми короткі і повністю вміщаються на екрані, піклуватися про економію місця не має сенсу. Але з часом вони стануть довшими. Щоб скоротити кількість рядків програми, ви можете розмістити кілька коротких операторів в рядку, треба розділяти їх знаком двокрапки (:).

Якщо у програмі є передача управління оператору, що міститься в іншій її частині, то цьому оператору можна дати деяке додаткове ім'я — **мітку**. Її ім'я може вміщувати від 1 до 40 символів, серед яких не повинно бути пропусків. Мітка повинна закінчуватись двокрапкою.

Одним з найбільш цікавих операторів **QBasic** є оператор **присвоєння**. У загальному вигляді його можна подати таким чином:

$$\text{ім'я змінної} = \text{вираз}$$

де *вираз* може бути і конкретним значенням, і деяким виразом, результат якого буде присвоєно змінній. Приклади:

$$\begin{aligned} D &= (-A + \text{SIN}(C)) \\ C &= 2 \end{aligned}$$

обчислює змінні C і D.

Спочатку обчислюються значення виразу, що міститься справа від знака, а потім результат обчислення, числове або текстове значення, присвоюється змінній, ім'я якої вказано зліва.

Уведення вихідних даних за допомогою:

- 1) оператора присвоєння;
- 2) присвоєння змінній значення з клавіатури під час виконання програми. Найпростіша форма оператора, що дає вам можливість це зробити, така:

INPUT n,

де n – ім'я змінної, для якої потрібне введення значення.

Інструкція **INPUT** n є командою, при зустрічі з якою програма виконує такі дії:

- на екрані з'являється знак питання;
- програма призупиняє роботу і чекає, коли ви наберете будь-яке значення на клавіатурі;
- натискання клавіші **Enter** на клавіатурі повідомляє програмі, що введення закінчено;
- введене значення присвоюється змінній n;
- комп'ютер продовжує виконання програми, використовуючи введене значення. Значення, яке мала раніше ця змінна n, забувається програмою.

Загальна форма для запису оператора **INPUT**:

INPUT «текст»; перелік змінних,

де «текст» — будь-який текст, що виводиться на екран перед запитом значення, повинен бути в лапках; перелік змінних – перелік змінних, для яких знадобиться введення значень; імена змінних повинні бути розділені крапкою з комою або комою.

Знак крапка з комою (;) відразу за словом **INPUT** визначає положення курсору після натискання клавіші введення Enter.

Приклад 1.2

```
CLS
REM Додавання двох чисел
    INPUT «Введіть перше число:»; N1
    INPUT «Введіть друге число:»; N2
PRINT «Сума=»; N1+N2 ‘ Виведення результату на екран
END
```

Приклад 1.3

Обчислити значення W і R , використовуючи формули:

$$W = \operatorname{atg}^3 x^2 + \sqrt{\frac{z^2}{a^2 + x^2}}; R = \ln(a + x^2) + \sin^2 \frac{z}{a},$$

при значеннях $a=0.59$, $z=-4.8$, $x=2.1$.

Програма надана нижче:

```
REM Лабораторна робота 1
REM Обчислення значення функції
CLS
INPUT «Введіть A, Z, X»; A, Z, X
PRINT «A=»; A, «Z=»; Z, «X=»; X
    W= A*TAN(X^2)^3+SQR(Z^2/(A^2+X^2))
    R=LOG(A+X^2)+SIN(Z/A)^2
PRINT «W=»; W, «R=»; R
END
```

Варіанти самостійних завдань до лабораторної роботи 1

Обчислити вирази при довільних значеннях змінних

Вар.	Функція
1	$y = \frac{(1 - \sin^2 x)^2}{\sin^2 x^3 + 5} + 5 \sin^2 x^2$
2	$p = \frac{\sin^6 z + \cos^6 z}{\sin^4 z + \cos^2 z \cdot \sin^2 z + \cos^4 z + 0,5}$
3	$y = \frac{(1 - \cos 2 \cdot x)^2}{2 \cdot \sin^2 x + 1} + 2 \cdot \cos^2 x$
4	$y = \frac{s^2 + \sin 2 \cdot s}{\sin^2 s + 10}$
5	$y = \cos b - \frac{(1 - \cos^2 b)^2}{5} + 2 \cdot \sin b^2$
6	$z = \frac{(1 - \cos a)^2}{2 \sin^2 a + 1} + \operatorname{tga}$
7	$c = \frac{(3 + \sin^2 a)^2}{\sin^2 b^3 + 5 \cdot \sin b + 1} + 25 \cdot \sin^2 a^2$
8	$y = \frac{(\sin^2 x + \cos^2 x)^2}{\sin^2 x^2 + 1 + \sin x} + \sin^2 x$
9	$y = \frac{(10 - \sin^2 x)^2}{3 + \sin^2 x} + x^2$
10	$y = \frac{x^2}{\sqrt{\sin^2 x^3 + 5}}$
11	$y = \frac{(1 + \operatorname{tg}^2 x)^2}{\operatorname{tg}^2 x^3 + 2} + 5 \cdot \operatorname{tg}^2 x^2$
12	$p = \sin \left(3 \sqrt[3]{\frac{x + y}{z^3 + \cos x + \operatorname{tgy} + 12}} \right)$
13	$y = \frac{(1 - \sin^2 x)^2}{\sin^2 x^3 + 5} + 5 \cdot \sin^2 x^2$
14	$p = \cos \left(3 \sqrt[3]{\frac{x^2 + y^2}{\cos x + \operatorname{tg} y + 6}} \right)$
15	$q = \frac{\operatorname{tg} x^2 + \operatorname{tg}^2 x}{\operatorname{arctg}^y 2 + \operatorname{arctg} 2^y}$
16	$y = \frac{(1 - \sin^2 x)^2}{\sin^2 x^3 + 5} + 5 \cdot \sin^2 x^2$
17	$z = \frac{(10 + \sqrt{\sin^2 x})^2}{\sin^2 x^3 + 7 \cdot \cos y^2 + 2}$

Вар.	Функція
18	$d = \frac{\operatorname{tga}^2 + \operatorname{tg}^2 a^2}{\operatorname{arctg}^2 b + \operatorname{arctg}^2 b + 1}$
19	$s = \operatorname{tg} \left(\sqrt[5]{\frac{z + x + y}{\sin z^5 + \cos x + \operatorname{tgy}}} \right)$
20	$q = \frac{\sin x^2 + \cos^2 x}{\operatorname{arctg}^x 2 + \operatorname{arctg} 2^x}$

Порядок виконання лабораторної роботи

- 1 Побудувати алгоритм рішення завдання.
- 2 Запрограмувати алгоритм за допомогою мови QBasic.
- 3 Набрати текст програми в середовищі QBasic.
- 3 Вивести результати обчислень на екран.

Зміст звіту

У звіті повинно бути:

- а) мета завдання;
- б) алгоритм рішення завдання;
- в) наведено програму розрахунку завдання;
- г) висновки за лабораторною роботою.

Питання до захисту лабораторної роботи 1

- 1 Надати визначення алгоритму лінійного обчислювального процесу.
- 2 Перелічити символи для побудови блок-схеми алгоритму.
- 3 Перелічити символи та конструкції мови QBasic.
- 4 Вказати правила використання оператора PRINT.
- 5 Вказати правила використання оператора початку програми (оператор CLS).
- 6 Вказати правила використання оператора закінчення програми.
- 7 Вказати правила використання коментарів у програмі.
- 8 Вказати правила використання оператора введення даних

Лабораторна робота 2

ПРОЕКТУВАННЯ АЛГОРИТМІВ РОЗГАЛУЖЕНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ. ПРОГРАМУВАННЯ МОВОЮ QBASIC АЛГОРИТМІВ РОЗГАЛУЖЕНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ З ВИКОРИСТАННЯМ ОПЕРАТОРА IF <УМОВА> THEN <ДІЯ_1> ELSE <ДІЯ_2>

Мета роботи: ознайомитися з алгоритмізацією розгалужених обчислювальних процесів, перевіркою умов у QBasic, оператором IF <Умова> THEN <Дія_1> ELSE <Дія_2>.

Завдання до лабораторної роботи

У цій лабораторній роботі необхідно скласти програму, яка обчислює алгоритм наведених у варіантах самостійних завдань до лабораторної роботи 2.

Програма повинна передбачати введення значень, аналіз умов, розрахунок відповідного прикладу та виведення результату на екран.

Короткі теоретичні положення

Оператори умовного переходу – одні з найважливіших і часто використовуваних елементів в мовах програмування. Загальний принцип їх роботи простий: перевіряється відповідність якимсь умовам (істинність або хибність яких-небудь виразів) і, залежно від цього, виконання програми направляється по одній або іншій гілках. У QBasic передбачено два оператори умовного переходу: IF... THEN... ELSE і SELECT CASE...END SELECT.

Оператор IF... THEN... ELSE... – найбільш популярний у програмістів. Повний його синтаксис виглядає так:

```
IF <Умова> THEN
  <Дія_1>
[ELSE
  <Дія_2>]
END IF
```

Умова – вираз, що перевіряється на істинність. Якщо він істинний, то виконується Дія_1, якщо хибний – Дія_2.

Умовою у всіх форматах IF THEN є вираз зі знаком порівняння. Він може мати тільки два значення: «true» (істина) і «false» (хибність). Якщо умова виконується, то його значенням є «true» (істина), якщо не виконується – «false» (хибність).

В умовах можна використовувати такі знаки порівняння:

- > - більше;
- < - менше;
- >= - більше або дорівнює;
- <= - менше або дорівнює;
- <> - не дорівнює.

Наприклад, $x < 8$ або $d \geq 17$. Справа і зліва від знака порівняння можуть розміщатися не тільки цифри і змінні, але і цілі математичні вирази.

Зокрема $(\text{SQR}(A + 5) + A^2) \geq \text{ABS}(\text{SIN}(2 * \text{Pi}))$

Умова може задаватися досить складним чином і складатися з декількох блоків. Для уведення таких виразів використовуються спеціальні оператори - AND, OR, NOT.

Формат 1 - якщо умова виконується, то ЕОМ спочатку виконує Дію_1, потім Дію_2. Якщо умова не виконується, то машина виконує тільки Дію_2. Блок схема зображена на рисунку 2.1.

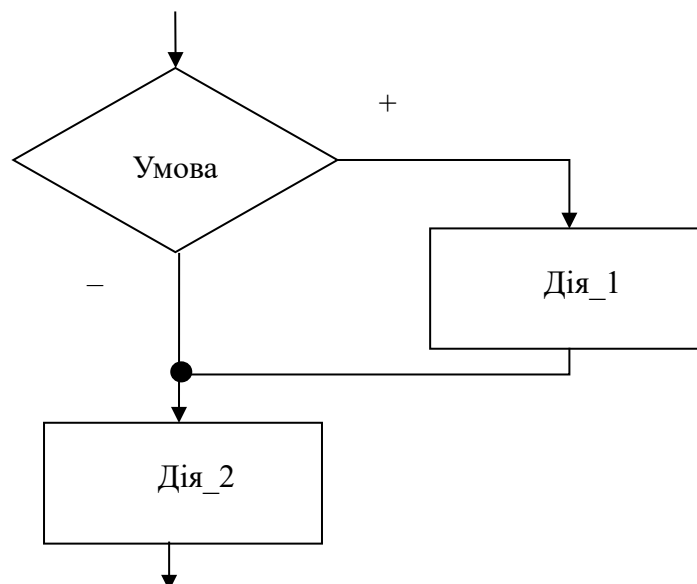


Рисунок 2.1 – Блок-схема формату 1 оператора IF ... THEN

Розглянемо коди найпростіших програм.

Приклад 2.1

Увести число X , якщо воно менше або дорівнює 5, то помножити на A та вивести результат на екран.

Розв'язання:

Алгоритм у вигляді блок-схеми має вигляд, наведений на рисунку 2.2.

Опис блок-схеми

Блок 1 - початок розрахунку.

Блок 2 - уведення X .

Блок 3 - перевірка умови.

Блок 4 - множення X на (-1) у випадку, якщо умова виконується.

Блок 5 - виведення X на екран.

Блок 6 - завершення розрахунку.

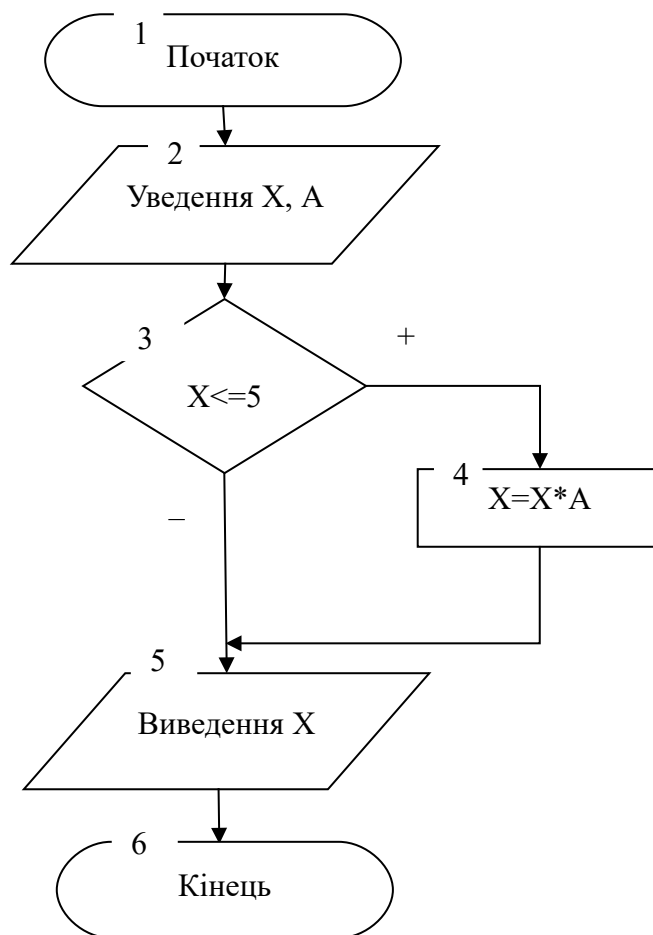


Рисунок 2.2 – Блок-схема до прикладу 2.1

Запишемо алгоритм, зображений на блок-схемі, у вигляді комп'ютерної програми.

```

CLS                'очищення екрана
REM розгалужений обчислювальний процес
INPUT "X="; X      'Уводимо X
INPUT "A="; A      'Уводимо A
IF X <= 5 THEN X = X * A 'Якщо X <= 5, то множимо його на A
PRINT "X ="; X     'Виводимо значення змінної X на екран
END
  
```

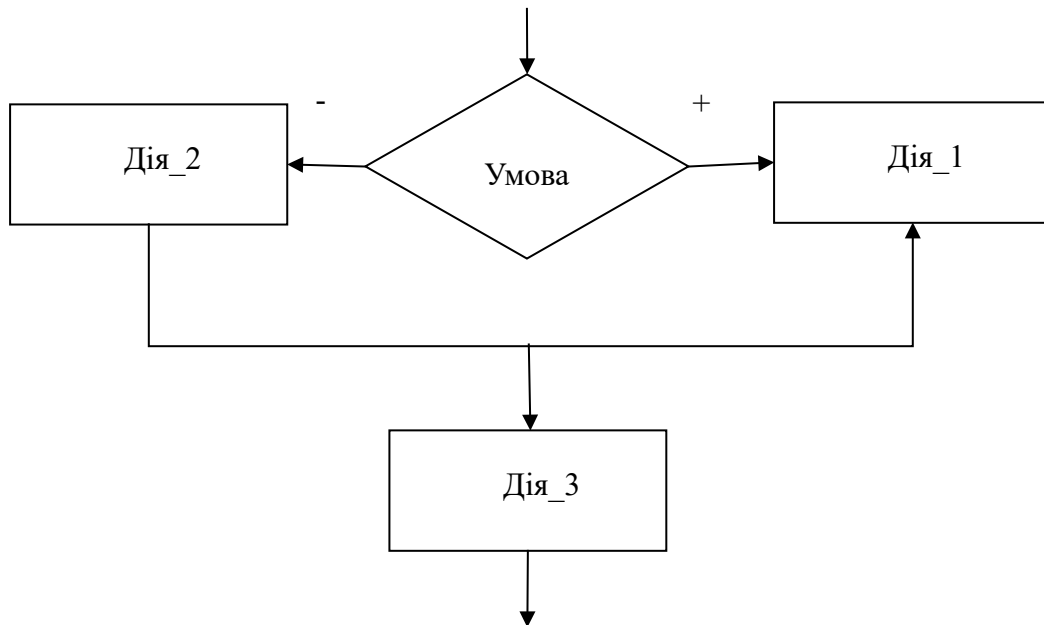
Робота програми

Після уведення X і A оператор IF ... THEN аналізує значення X . Якщо $X \leq 5$, то програма виконує дію $X = X * A$, потім виводимо оновлене значення X . Якщо $X > 5$, то буде виведено початкове значення X .

Формат 2 **IF ... THEN ... ELSE** дає змогу виконати або першу або другу дії залежно від умови.

Блок-схема формату **IF ... THEN ... ELSE ...** наведена на рисунку 2.3.

Ри



сунок 2.3 – Блок-схема конструкції **IF ... THEN ... ELSE...**

Формат **IF ... THEN ... ELSE** має вигляд:

```
IF Умова THEN Дія_1 ELSE Дія_2  
Дія_3
```

Якщо умова виконується, то машина робить Дію_1, потім Дію_3. Якщо умова не виконується, то машина робить Дію_2, а потім Дію_3.

Формат 1 і формат 2 є лінійною формою оператора **IF ... THEN ... ELSE**.

Крім лінійної форми оператор **IF ... THEN ... ELSE** має ще блокову форму. У цьому випадку формат має вигляд:

```
IF Умова_1 THEN  
    Дія_1  
ELSEIF Умова_2 THEN
```

```

    Дія_2
ELSEIF Умова_3 THEN
    Дія_3
...
ELSE
    Дія_N
END IF
    Дія_N+1

```

Якщо умова_1 виконується, то машина робить Дію_1, потім Дію_N+1. Якщо умова_1 не виконується, то машина перевіряє умову_2. Якщо ця умова істинна, то виконується Дія_2, а потім Дія_N+1. Інакше перевіряється умова_3 і так далі. Якщо всі умови хибні, то виконується Дія_N, що йде безпосередньо за ELSE. Ознакою блокової форми є відсутність операторів після службового слова **THEN**. Блокова форма припускає будь-яку кількість блоків ELSEIF. Закінчується умовний оператор блокової форми запису службовими словами **END IF**

Приклад 2.2

Скласти блок-схему й програму розв'язання завдання:

$$y = \begin{cases} x^2 + |x+1|, & \text{при } x < 0, \\ 10a + \cos x, & \text{при } x = 0, \\ (x+2) \cdot (b+x^2), & \text{при } x > 0. \end{cases}$$

Програма повинна передбачати введення трьох значень, аналіз умов, розрахунок відповідного прикладу та виведення результату на екран.

Розв'язання

Алгоритм програми записано у вигляді блок-схеми, що наведена на рисунку 2.4.

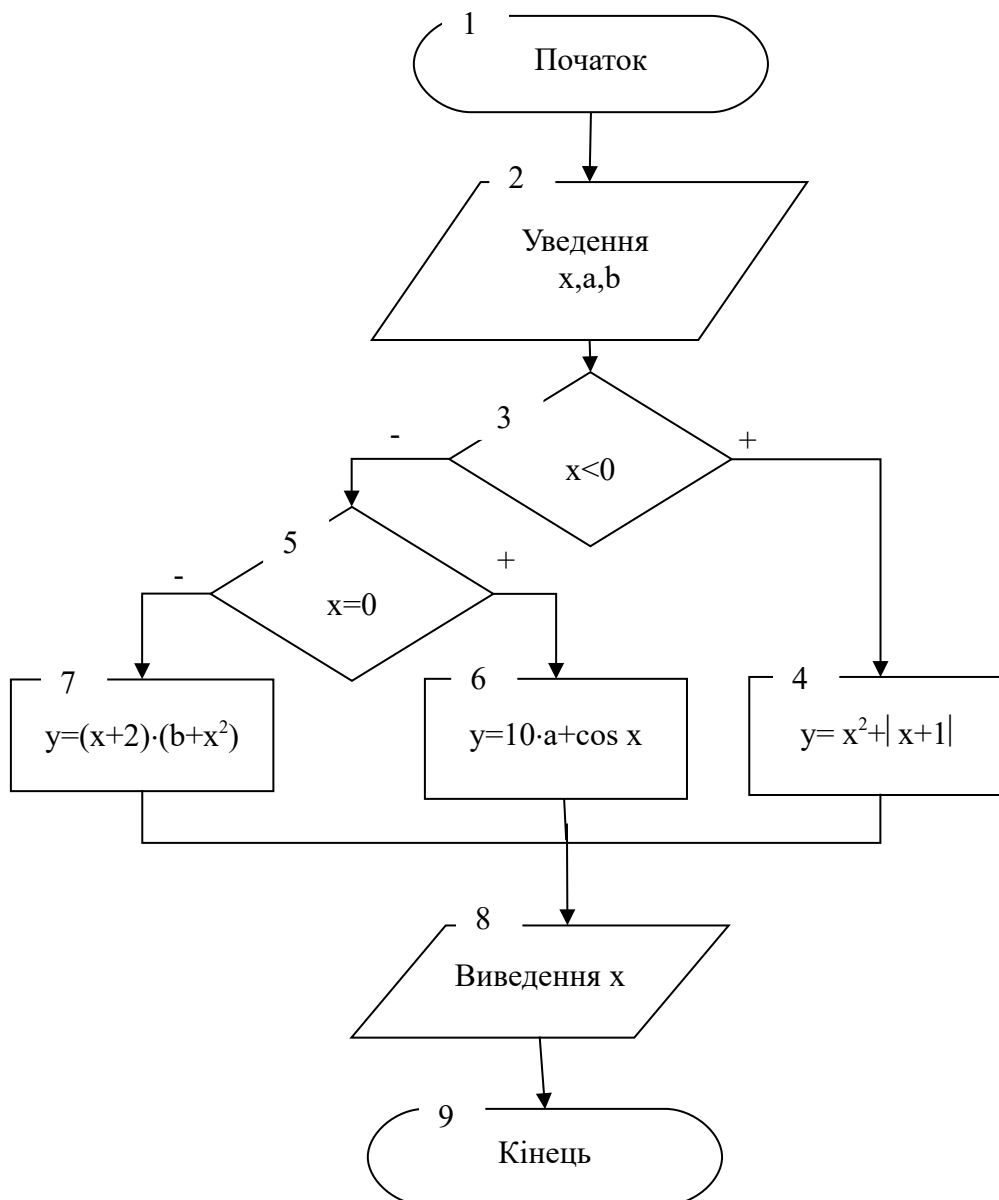


Рисунок 2.4 – Блок-схема до прикладу 2.2

Опис блок – схеми

Блок 1 – початок роботи програми.

Блок 2 – введення змінних x, a, b .

Блок 3 – перевірка умови $x < 0$.

Якщо умова виконується, то відбувається розрахунок виразу в блоці 4 з подальшим переходом у блок 8.

Якщо умова не виконується, то управління переходить до блоку 5.

У блоці 5 здійснюється перевірка умови $x = 0$. У випадку виконання умови відбувається розрахунок формули в блоці 6 та перехід до блоку 8.

Якщо умова не виконується, відбувається розрахунок формули в блоці 7 та перехід до блоку 8.

Блок 8 - виведення результату на екран.

Блок 9 – завершення роботи програми.

Запишемо алгоритм , зображений на блок – схемі , у вигляді комп'ютерної програми:

```
CLS
INPUT a,b,x
IF x < 0 THEN
    y = x^2 + ABS(X + 1)
ELSEIF x = 0 THEN
    y = 10*a + COS(x)
ELSE
    y = (x + 2)*(b + x^2)
END IF
PRINT Y
END
```

Приклад 2.3

Скласти блок-схему й програму розв'язання завдання:

$$s = 10; \quad k = t^2 - a^2; \quad v = t^3 + u;$$

$$u = \begin{cases} \ln(k^2 + 1), & \text{якщо } 15 \leq k \leq 20; \\ \frac{\sqrt{2k+7}}{\sqrt{2k+7} + 3}, & \text{якщо } 10 \leq k < 15 \text{ або } k > 25; \\ 1 - \frac{\ln(2k+7)}{|sk+3|}, & \text{у інших випадках;} \end{cases} \quad b = \begin{cases} 1\sqrt{w}, & \text{якщо } w > 10; \\ 2\lg w, & \text{якщо } 0 < w \leq 10; \\ 3 \cos(w+v), & \text{якщо } w = 0 \text{ або } v > 30; \\ 4, & \text{у інших випадках.} \end{cases}$$

Розв'язання

Алгоритм програми записано у вигляді блок-схеми, що наведена на рисунку 2.5.

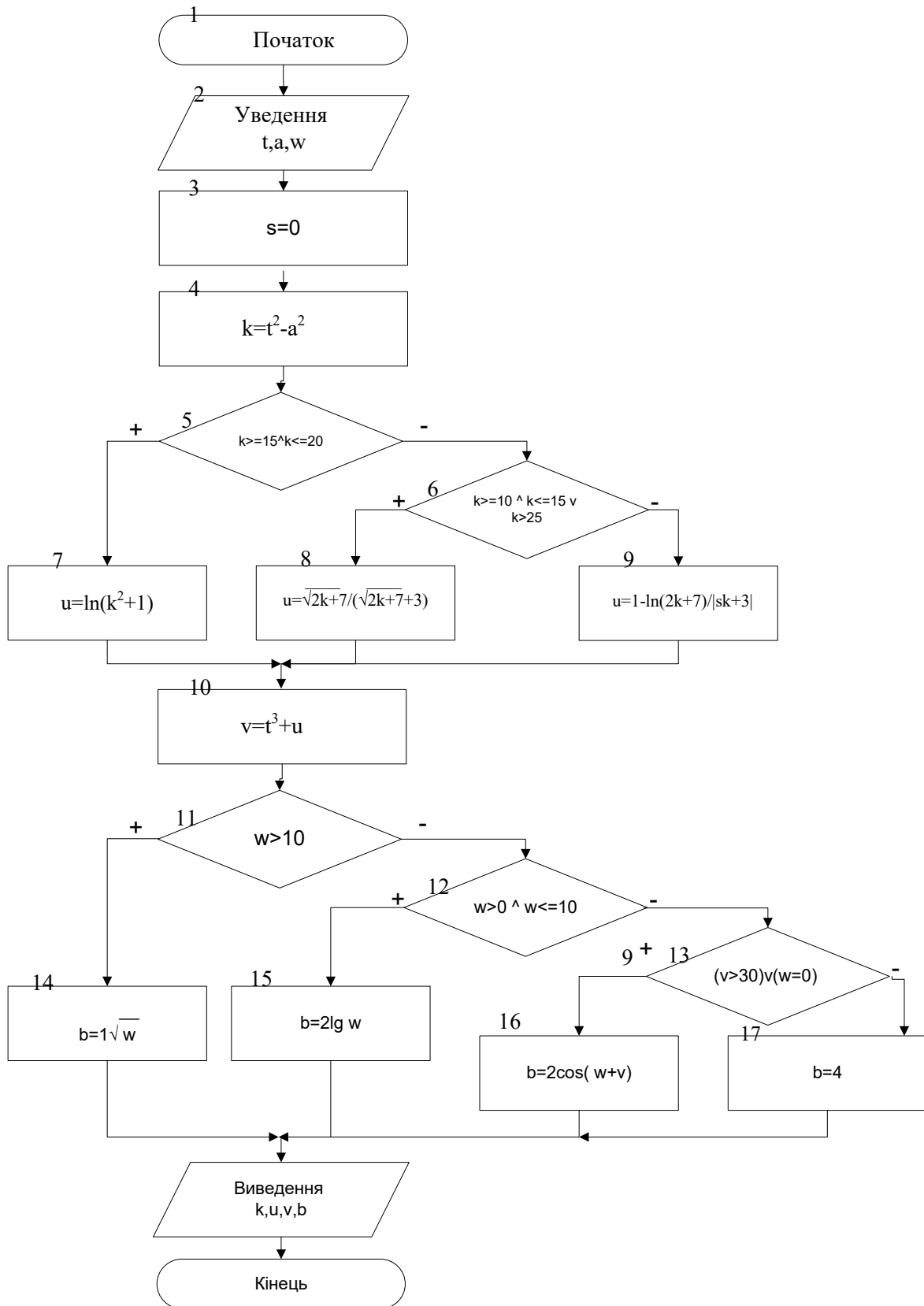


Рисунок 2.5 – Блок-схема до прикладу 2.3

Запишемо алгоритм, зображений на блок-схемі, у вигляді комп'ютерної програми:

```

CLS
INPUT t,a,w
s=10 : k=t^2-a^2
  IF 15 <=k AND k<=20 THEN
    u = LOG(k^2 +1)
  ELSEIF 10< = k AND k<15 OR k>25 THEN
    u = SQR(2*k+7)/ SQR(2*k+7) + 3
  ELSE
    u = LOG(2*k+7)/ABS(s*k+3)
  END IF
v=t^3+u
  IF 10 <w THEN
    b = 1*SQR(w)
  ELSEIF 0<w AND w<10 THEN
    b = 2*LOG(w)/ LOG(10)
  ELSEIF 0=w OR v>30 THEN
    b = 3*COS(w+v)
  ELSE
    b=4
  END IF
PRINT "b="; b, "v="; v, "u="; u, "k="; k
END

```

Варіанти самостійних завдань до лабораторної роботи 2

Варіант	Завдання
1	$y = \begin{cases} 10 \cdot x^2 + 7 \cdot x^2, & \text{якщо } x \leq 0, \\ 3, & \text{якщо } x = 0, \\ x^5 + 7\sqrt{x}, & \text{якщо } x > 0 \end{cases}$
2	$y = \begin{cases} 5 \cdot x^2 + 2 \cdot x + 1, & \text{якщо } x \leq 1, \\ 3/x + \sqrt{x^2 + 1}, & \text{якщо } x = 1, \\ (4 + 2 \cdot x) / \sqrt{x^2 + 1}, & \text{якщо } x \geq 1 \end{cases}$
3	$y = \begin{cases} 1 \cdot 5 \cdot x - x^3, & \text{якщо } x < 5, \\ (x - 2)^2 + 6 \cdot x, & \text{якщо } x = 5, \\ 3 \cdot x + \sqrt{3 + x}, & \text{якщо } x \geq 5 \end{cases}$
4	Дано три відрізки a, b, c. Перевірити, чи існує трикутник з такими сторонами

Варіант	Завдання
5	Дано дійсні числа x, y ($x \neq y$). Менше із цих двох чисел замінити половиною їх суми, а більше - їх подвоєним добутком.
6	$z = \begin{cases} x^2 + 3 \cdot x, & \text{якщо } x \leq 0, \\ \sqrt{x^2 + 4 \cdot x}, & \text{якщо } x > 0 \end{cases}$
7	$y = \begin{cases} x^3 + x^2 - 5, & \text{якщо } x \leq 1, \\ 3 + \sqrt{x^2 + 1}, & \text{якщо } x = 1, \\ 2 \cdot x / \sqrt{x^2 + 1}, & \text{якщо } x \geq 1 \end{cases}$
8	$y = \begin{cases} 15 \cdot x^4 - 8 \cdot x^2, & \text{якщо } x \leq 2, \\ 8 + x^2, & \text{якщо } x = 2, \\ x^3 + 8 \cdot \sqrt{x}, & \text{якщо } x > 2 \end{cases}$
9	Змінній D присвоїти мінімальне значення змінних цілого типу A, B, C. Вивести на екрану всі значення.
10	$z = \begin{cases} y^2 + 3, & \text{якщо } y \leq 0, \\ \sqrt{y^2 + 5}, & \text{якщо } y > 0 \end{cases}$
11	$y = \begin{cases} 18 \cdot x^2 , & \text{якщо } x \leq 2, \\ 7, & \text{якщо } x = 2, \\ 5 \cdot \sqrt{x}, & \text{якщо } x > 2 \end{cases}$
12	$y = \begin{cases} 3 \cdot x ^3 + 5 \cdot x^2, & \text{якщо } x \leq 1, \\ 5, & \text{якщо } x = 1, \\ 2 / \sqrt{x^2 + 1}, & \text{якщо } x \geq 1 \end{cases}$
13	$y = \begin{cases} 4 \cdot x^3 + 4 \cdot x^2, & \text{якщо } x \leq 0, \\ 3, & \text{якщо } x = 0, \\ \sqrt{x^2 + 1}, & \text{якщо } x \geq 0 \end{cases}$
14	$y = \begin{cases} x + x^2 + 2, & \text{якщо } x \leq 10, \\ 3 + \sqrt{x^2 + 1}, & \text{якщо } x = 10, \\ 2 \cdot x / \sqrt{x^2 + 4}, & \text{якщо } x > 10 \end{cases}$
15	<p>Скласти алгоритм та програму процесу обчислення величини q.</p> $n = m^3 + \sqrt{f^2 - 1}; \quad b = 25 - f^2;$ $m = \begin{cases} 1 + \frac{2b}{ f }, & \text{якщо } f \geq 100 \text{ або } f < -20; \\ \sqrt{b^2 + 3} - b^2, & \text{якщо } 50 \leq f < 100 \text{ або } -5 \leq f < 25; \\ \ln(b^2 + f), & \text{якщо } 25 \leq f < 50; \\ 25, & \text{у інших випадках;} \end{cases} \quad q = \begin{cases} \ln n, & \text{якщо } f > 0; \\ \sqrt[4]{e^b + 2}, & \text{якщо } f < 0; \\ 0, & \text{якщо } f = 0 \end{cases}$
16	<p>Скласти алгоритм та програму процесу обчислення величини x.</p> $x = \begin{cases} 2a, & \text{якщо } a > b \text{ та } b > 0 \text{ або } b > a; \\ \ln a + 5 - k^3, & \text{якщо } a > b \text{ та } b < -5; \\ \cos(3a) - 2, & \text{якщо } a = b \neq 0; \\ w + \sin^3 c, & \text{у інших випадках;} \end{cases} \quad k = w^2 - 2c + 24;$ $w = \begin{cases} \frac{a + a^3}{c + \ln(c^2 + 1)} + 32, & \text{якщо } c > 0; \\ b - 4e^{c+5}, & \text{якщо } c \leq 0 \end{cases}$

Порядок виконання лабораторної роботи

- 1 Побудувати алгоритм розв'язання задачі.
- 2 Запустити QBASIC.

3 Програма повинна містити введення вхідних даних, розрахунок формул та виведення результатів на екран.

Зміст звіту

У звіті повинно бути:

- а) мета завдання;
- б) розв'язання прикладів 2.1, 2.2;
- в) обчислення завдання, згідно з варіантом;
- г) результати обчислень;
- д) висновки за лабораторною роботою.

Питання до захисту лабораторної роботи 2

1 Навести формат оператора IF ... THEN і описати його роботу.

2 Навести блок-схеми роботи оператора IF ... THEN.

3 Які знаки порівняння можна використовувати при записі умов в операторі IF ... THEN.

4 Навести приклад використання оператора IF, якщо треба, щоб після THEN виконувались декілька операторів.

Лабораторна робота 3

ПРОЕКТУВАННЯ АЛГОРИТМІВ ТА ПРОГРАМУВАННЯ МОВОЮ QBASIC АЛГОРИТМІВ ПРОСТИХ АРИФМЕТИЧНИХ ЦИКЛІЧНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

Мета роботи: Ознайомитися з алгоритмізацією простих арифметичних циклічних обчислювальних процесів, програмуванням їх за допомогою оператора циклу FOR...NEXT мови QBasic.

Завдання до лабораторної роботи

У цій лабораторній роботі потрібно обчислити всі наведені приклади та скласти програму розрахунків математичних прикладів (варіанти завдань вказує викладач), що містить суму й добуток числового ряду.

Короткі теоретичні положення

Цикл – це процес виконання певного набору команд деяку скінчену кількість разів. Розрізняють цикли, де кількість повторень відома заздалегідь, і цикли, де вона заздалегідь невідома, але її можна визначити під час виконання циклу.

Розглянемо побудову циклу. Циклові передують етап підготовки циклу. Тут задають початкові значення змінних, які використовуються в циклічному обчислювальному процесі. Власне цикл складається з таких трьох етапів:

1) перевірка умови циклу. Перевіряється умова, яка забезпечує вихід з циклу після досягнення мети. Якщо умова складена неправильно, то можна ніколи не вийти з циклу. Така ситуація називається зациклюванням і її потрібно уникати;

2) виконання серії команд (основні дії);

3) зміна значення параметра. Згідно з умовою завдання змінюється значення змінної, яка є в умові. Ця змінна називається параметром циклу.

В QBasic оператори циклу застосовуються для розв'язання завдання, у яких потрібно багаторазово виконувати деякі дії (тіло циклу). У тих випадках, коли заздалегідь відомо скільки раз потрібно виконати дію, використовується оператор циклу FOR... NEXT. Його формат виглядає так:

```
FOR лічильник = Поч_значення To Кінц_значення <STEP Знач_кроку>  
  < Тіло циклу >  
NEXT <Лічильник>
```

Лічильник – числова змінна, значення якої автоматично змінюється в процесі роботи й відслідковується циклом.

Поч_значення – початкове значення лічильника циклу. Воно може бути задане числом, змінною, математичним виразом. Значення може бути цілим або дробовим, додатним або від'ємним.

Кінц_значення – кінцеве значення лічильника. Воно може бути задане числом, змінною, математичним виразом. Значення

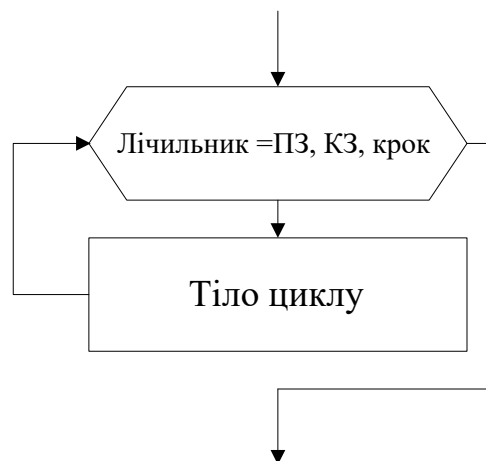
може бути цілим або дробовим, додатним або від'ємним.

Знач_кроку – значення кроку, величина, що додається до лічильника після кожної ітерації. Вона може бути задана числом, змінною або математичним виразом. Додатне значення кроку використовується тоді, коли початкове значення лічильника менше від кінцевого. Якщо початкове значення більше від кінцевого, то слід застосовувати від'ємний крок. Якщо крок дорівнює 1, то STEP 1 можна не писати.

NEXT – повертає керування в рядок FOR....

Назву лічильника за цією командою вказувати не обов'язково. Якщо програмний блок містить декілька вкладених циклів, то за NEXT можна ставити кілька лічильників через кому.

Програма визначає як тіло циклу усі команди, розташовані між FOR і NEXT. Цикл припиняє свою роботу після того, як лічильник вийде за рамки свого кінцевого значення. Якщо в процесі роботи циклу крок додатний, то цикл буде завершений, коли лічильник стане більше від кінцевого значення. Якщо в процесі роботи крок від'ємний, то керування піде із циклу після того, як лічильник стане менше від свого кінцевого значення. У тому випадку, коли в програмі не зазначена одна з команд Next або For, а інша введена, при запуску буде видане повідомлення про помилку “For without Next” або “Next without For”. Зображення оператора For...Next у блок-схемі показано на рисунку 3.1.



ПЗ – початкове значення лічильника;

КЗ – кінцеве значення лічильника

Рисунок 3.1 – Оператор FOR...NEXT на блок-схемах

Циклічні структури зручно використовувати при рішенні завдань, де необхідно обчислювати суму, добуток або кількість деякої множини довільних числових констант. Для циклічного накопичення сум, добутків і кількості при складанні відповідних алгоритмів використовуються приписи стандартного виду подані в таблиці 3.1.

Таблиця 3.1 – Циклічне накопичення сум, добутків і кількості

Обчислення	Початкове значення	Рекурентне співвідношення
Сума чисел (S)	S=0, x - доданок	S=S+x
Кількість чисел (K)	K=0	K=K+1
Добуток чисел (D)	D=1, x - множник	D=D*x

Рекурентним співвідношенням називається формула виду $a_{n+1}=F(a_n, a_{n-1}, \dots, a_{n-k+1})$, де F деяка [функція](#) від k [аргументів](#), яка дає змогу обчислювати наступні члени [послідовності](#) через значення попередніх членів. Якщо вказати перших k членів послідовності, то рекурентне співвідношення однозначно визначає послідовність a_n .

Наприклад:

Рекурентне співвідношення [арифметичної прогресії](#):

$$a_{n+1}=a_n+d.$$

Рекурентне співвідношення [геометричної прогресії](#):

$$a_{n+1}=a_n \cdot q.$$

Рекурентне співвідношення послідовності [n!](#):

$$a_{n+1}=a_n \cdot (n+1).$$

Приклад 3.1

Обчислити суму всіх цілих чисел від 1 до 25.

Розв'язання:

Побудова блок-схеми алгоритму (рисунок 3.2).

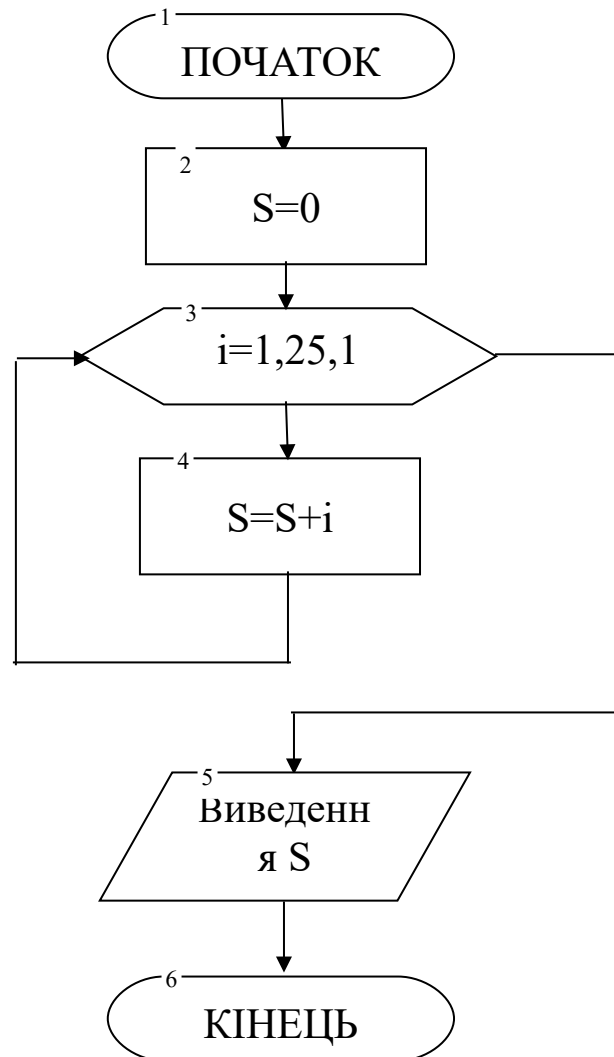


Рисунок 3.2 –Алгоритм розв'язання прикладу 3.1

Опис блок-схеми:

Блок 1 – початок обчислень.

Блок 2 – введення початкового значення змінної S.

Блок 3 – запис параметра циклу.

Блок 4 – тіло циклу (накопичення в оперативній пам'яті змінної S).

Блок 5 – виведення остаточного значення змінної S.

Запишемо алгоритм, зображений на блок-схемі, у вигляді комп'ютерної програми.

```

CLS
    S=0
FOR i = 1 TO 25
    S = S + i
NEXT i
    PRINT "S=";S
END

```

Робота програми

Програма повторює дію $S = S + i$ 25 разів, складаючи суму всіх цілих чисел від 1 до 25 ($S = 1+2+3+4+\dots+24+25$).

Лічильник циклу i після кожної ітерації збільшується на 1.

Сума накопичується в оперативній пам'яті в змінної S . При першій ітерації $S = S + i = 0 + 1 = 1$. При другій $S = S + i = 1+2 = 3$ і так далі. У ході чергової ітерації програма додає до значення, що зберігається в S поточне значення i , розраховує суму й записує її в комірку пам'яті, у якій розташована змінна S .

У математиці сума різних чисел з кроком 1 записується за допомогою знака Σ :

$$S = \sum_{i=1}^{25} i$$

Приклад 3.2

Розрахувати добуток кубів цілих чисел від 1 до 30.

Розв'язання:

1. Побудова блок-схеми алгоритму (рисунок 3.3).

Розрахувати суму та добуток ряду 1, 3, 5, 7, 9, 11.

```
CLS
S = 0: D = 1
FOR x = 1 TO 11 STEP 2
S = S + x
D = D * x
NEXT x
PRINT "S="; S, "D="; D
END
```

Робота програми

У циклі розраховуються сума й добуток чисел, а потім програма обчислює приклад і виводить результат на екран. Числа, яким стає рівний лічильник x у кожній ітерації змінюються із кроком 2, тому в першій ітерації $x=1$, у другій – $x=3$, у третій $x=5$ і т.п. до $x=11$.

Приклад 3.4

Протабулювати дві функції $y = \sin x$, $z = \cos x$ на проміжку $[0;1]$, змінюючи значення аргументу на 0,01.

Табулювання функції – це побудова таблиці значень функції для різних значень аргументу.

```
CLS
REM Приклад 3.4-Табулювання функцій
FOR X = 0 TO 1 STEP 0.01
    Y = SIN(X)
    Z = COS(X)
    PRINT "X="; X, "Y="; Y, "Z="; Z
NEXT X
END
```

Варіанти самостійних завдань до лабораторної роботи 3

Вар.	Завдання
1	Скласти алгоритм та програму табулювання функції $y=x^2+2x+5$ на відрізку $[1,2]$ з кроком 0,1

2	Скласти алгоритм та програму табулювання функції $y=3x^3-x^2$ на відрізку $[3,5]$ з кроком 0,5
3	Скласти алгоритм та програму знаходження суми чисел від 50 до 100
4	Скласти алгоритм та програму розрахунку виразу $y = \sum_{i=1}^9 i^2 + \prod_{i=1}^9 i^3$
5	Знайти суму непарних чисел від 20 до 40. Скласти алгоритм та програму.
6	Скласти алгоритм та програму розрахунку виразу $y = \sqrt{\prod_{x=1}^{15} x^2 + \sum_{i=2}^5 (i+1)}$
7	Скласти алгоритм та програму розрахунку суми ряду $S=1/2+2/3+\dots+(n-1)/n$
8	Скласти алгоритм та програму розрахунку суми ряду $S=1+1/2^2+1/3^2+\dots+1/n^2$
9	Скласти алгоритм та програму розрахунку суми ряду $S_n=1+1/2+1/4+\dots+1/2^n$
10	Скласти алгоритм та програму розрахунку суми ряду $X_n=1-1/3+1/5-\dots+(-1)^{n-1}/(2n-1)$
11	Скласти алгоритм та програму розрахунку суми ряду $S=1+1/3^2+\dots+1/(2n-1)^2$
12	Скласти алгоритм та програму розрахунку суми ряду $S=x-x^2/2+x^3/3-x^4/4+\dots+(-1)^{n-1} \cdot x^n/n$
13	Скласти алгоритм та програму розрахунку добутку ряду $Y=n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-k+1)$
14	Скласти алгоритм та програму розрахунку виразу $y = \cos\left(\sum_{x=1}^8 \frac{1}{x^2} + \prod_{x=1}^8 \frac{1}{x+1}\right)$
15	Скласти алгоритм та програму розрахунку виразу $y = \sqrt{\prod_{x=1}^{15} x^2 + \sum_{i=2}^5 \left(\frac{i+1}{i^4+2}\right)}$

Порядок виконання лабораторної роботи

1 Побудувати алгоритм розв'язання задачі.

2 Запустити QBASIC.

3 Програма повинна містити введення вхідних даних, розрахунок формул та виведення результатів на екран.

Зміст звіту

У звіті повинно бути:

- 1) мета завдання;
- 2) розв'язання завдання, згідно з варіантом (побудова алгоритму та програми). Програма повинна містити введення вхідних даних, обчислення системи рівнянь та виведення результатів на екран;
- 3) результати обчислень;
- 4) висновки за лабораторною роботою.

Питання до захисту лабораторної роботи 3

- 1 Навести формат циклу FOR...NEXT і опишіть його дію.
- 2 Призначення елементів формату циклу FOR...NEXT.
- 3 Скласти програму розрахунків суми всіх цілих чисел від 10 до 20.
- 4 Скласти програму розрахунків добутку всіх цілих чисел від 1 до 10.

Лабораторна робота 4

ПРОГРАМУВАННЯ ІТЕРАЦІЙНИХ ЦИКЛІЧНИХ ПРОЦЕСІВ З ВИКОРИСТАННЯМ ОПЕРАТОРІВ WHILE ... WEND, DO ... LOOP

Мета роботи: Оволодіння практичними навичками розроблення та програмування алгоритмів ітераційної циклічної структури.

Завдання до лабораторної роботи

У цій лабораторній роботі потрібно обчислити всі наведені приклади та скласти програму розрахунків математичних прикладів (варіанти завдань вказує викладач).

Короткі теоретичні положення

Ітераційним циклом називається цикл, число повторень

дій у якому невідоме і не може бути обчислено до виконання циклу. Розглянемо як задати цикл, що виконується за певних умов.

Для циклу з передумовою перевіряється значення логічного виразу; якщо воно має значення *<true>*, то оператори, що входять у цикл, виконуються, у противному випадку здійснюється виконання оператора, що впливає за циклом.

Цикл з післяумовою виконується хоча б один раз. Перевіряється логічний вираз; якщо його значення *<false>*, то оператори, що входять у цикл, виконуються, у противному випадку здійснюється вихід із циклу.

Оператор WHILE... WEND

Базовий формат:

```
WHILE <умова>  
<Тіло циклу>  
WEND,
```

де умова – логічний вираз; тіло циклу – послідовність операторів.

Цей цикл управляється умовами, які можуть мати значення «*true*» або «*false*». Перед тим, як будуть виконуватися оператори, що складають тіло циклу, QBASIC визначає результат умови в рядку WHILE. Якщо умова має значення «*true*», то виконуються оператори тіла циклу, після чого управління повертається в початок оператора WHILE, де знову перевіряється умова. Виконання операторів, що складають тіло циклу, буде повторюватися доти, поки умова залишається «*true*». Як тільки результат перевірки умови змінюється на «*false*», QBasic обходить цикл і передає управління рядку програми, що стоїть за WEND.

Оператори, що формують тіло циклу, повинні яким-небудь чином впливати на змінні, які входять в умову, інакше програма буде ходити по колу, процесор ніколи не зможе зупинитися. Це явище називають *зациклюванням*.

Конструкцію WHILE... WEND застосовують тоді, коли невідомо, скільки разів має повторитися цикл. Щоб умова припинення роботи циклу коли-небудь здійснилась, ви повинні самі, усередині циклу, змінювати значення змінної, що використовується в перевірці.

Приклад 4.1

Обчислити функцію

$$P(y) = (y+1!) \cdot y \cdot (y^2+2!) \cdot y^2 \cdot (y^3+3!) \cdot y^3 \times \dots,$$

де $y \geq 1$. Обчислення закінчити, коли значення функції стане більше від заданого числа E .

Розв'язання

1 Побудова блок-схеми алгоритму (рисунок 4.1).

Опис блок-схеми

Блок 1 – початок обчислень.

Блок 2 – введення значень змінних y та E .

Блок 3 – введення початкових значень змінних P та K .

Блок 4 – перевірка умови закінчення обчислювального процесу.

Блок 5 – увімкнення і збільшення значення лічильника на 1.

Блок 6 – введення початкового значення факторіала.

Блок 7 – запис параметра циклу.

Блок 8 – рекурентне співвідношення для обчислення факторіала.

Блок 9 – рекурентне співвідношення для обчислення D .

Блок 10 – виведення з оперативної пам'яті остаточного значення змінної P .

Блок 11 – закінчення обчислювального процесу.

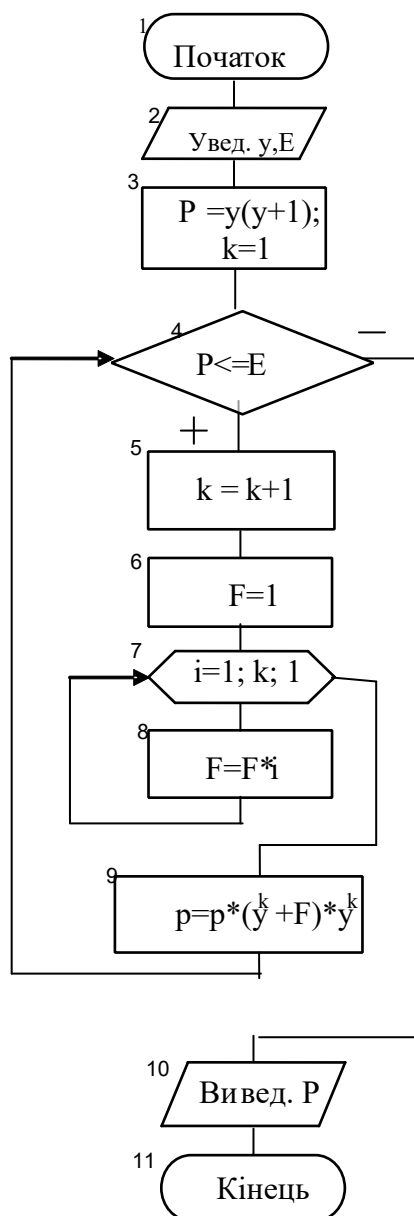


Рисунок 4.1. – Блок-схема алгоритма вирішення прикладу 4.1

2 Програма

CLS

INPUT "y=";y

INPUT "E=";E

P=y*(y+1): k=1

WHILE P<=E

k=k+1

F=1

FOR i = 1 TO k

F = F*i

NEXT i

```
    P=P*(y^k+F)*y^k
WEND
PRINT "P=";P
END
```

Оператор DO... LOOP

Оператор DO... LOOP дуже схожий на оператор WHILE... WEND, але він більш гнучкий і в цьому його перевага. Управляюча умова в цьому випадку може бути розміщена як на початку циклу (передумова), так і наприкінці (післяумова). Оператор має чотири різноманітні форми запису:

Цикл з передумовою:

```
DO WHILE <умова>
<Тіло циклу>
LOOP
```

```
DO UNTIL <умова>
<Тіло циклу>
LOOP
```

Цикл з післяумовою:

```
DO
<Тіло циклу>
LOOP WHILE <умова>
```

```
DO
<Тіло циклу>
LOOP UNTIL <умова>
```

Обидві форми запису починаються з ключового слова DO і закінчуються ключовим словом LOOP. Кожен логічний вираз починається зі слів WHILE або UNTIL. Якщо ви використовуєте слово WHILE, цикл буде виконуватися доти, поки умова буде мати значення «*true*». При записі через слово UNTIL цикл виконується тільки тоді, коли умова має значення «*false*».

Приклад 4.2

Побудувати алгоритм і програму, що виконують розрахунок та виведення на екран результатів обчислення суми ряду

$$y = \sum_{i=2}^{\infty} \frac{i-a}{(b+1)^i}$$

з точністю $E=0.0001$. Обчислення закінчити, якщо черговий елемент послідовності стане менше від заданого E .

Розв'язання:

1 Побудова блок-схеми алгоритму (рисунок 4.2).

Опис блок-схеми алгоритму:

Блок 1 – початок обчислень.

Блок 2 – введення значень змінних a, b .

Блоки 3 – введення значення точності E .

Блоки 4 – введення початкових значень змінних суми u та лічильника i .

Блок 5 – обчислення поточного складника суми z .

Блок 6 – рекурентне співвідношення для обчислення u .

Блок 7 – збільшення лічильника i .

Блок 8 – перевірка умови закінчення обчислювального процесу. Якщо умова істинна, то перехід до блоку 9, інакше перехід до блоку 5.

Блок 9 – виведення з оперативної пам'яті остаточного значення змінної u .

Блок 10 – закінчення обчислювального процесу.

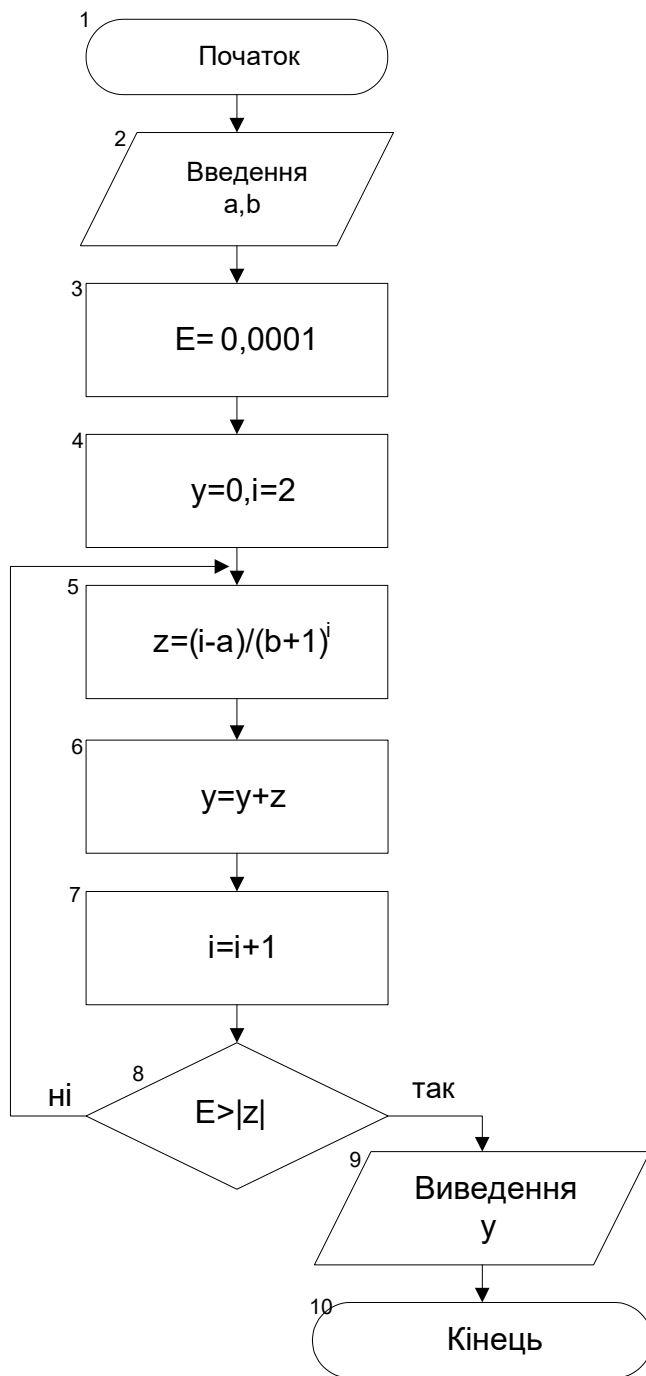


Рисунок 4.2 – Алгоритм вирішення прикладу 4.2

2 Програма

CLS

INPUT "a=";a

INPUT "b=";b

E=0.0001

y=0 : i=2

DO

```

z=(i-a)/(b+1)^i
y=y+z
i=i+1
LOOP UNTIL E>ABS(z)
PRINT "y=";y
END

```

Приклад 4.3

Побудувати алгоритм і програму, що виконують розрахунок та виведення на екран результатів обчислення суми S та кількість складників:

$$S = \sum_{(n)} \frac{2 \cdot n - 1}{(3 \cdot n)!} y^k, n=1,2,3,\dots$$

Обчислення закінчити, коли чисельник дробу $(2 \cdot n - 1)/(3 \cdot n)!$ стане в K раз менше знаменника.

Розв'язання

1 Побудова блок-схеми алгоритму (рисунок 4.3)

Опис блок-схеми алгоритму

Блок 1 – початок обчислень

Блок 2 – введення значень змінних y та K .

Блок 3 – введення початкових значень змінних S та n .

Блок 4 – збільшення значення лічильника на 1.

Блок 5 – обчислення поточного значення чисельника m складника суми.

Блок 6 – введення початкового значення факторіала.

Блок 7 – запис параметра циклу.

Блок 8 – рекурентне співвідношення для обчислення факторіала.

Блок 9 – рекурентне співвідношення для обчислення S .

Блок 10 – перевірка умови закінчення обчислювального процесу. Якщо умова істинна, то перехід до блоку 4, інакше перехід до блоку 11.

Блок 11 – виведення з оперативної пам'яті остаточного значення змінної S та n .

Блок 12 – закінчення обчислювального процесу.

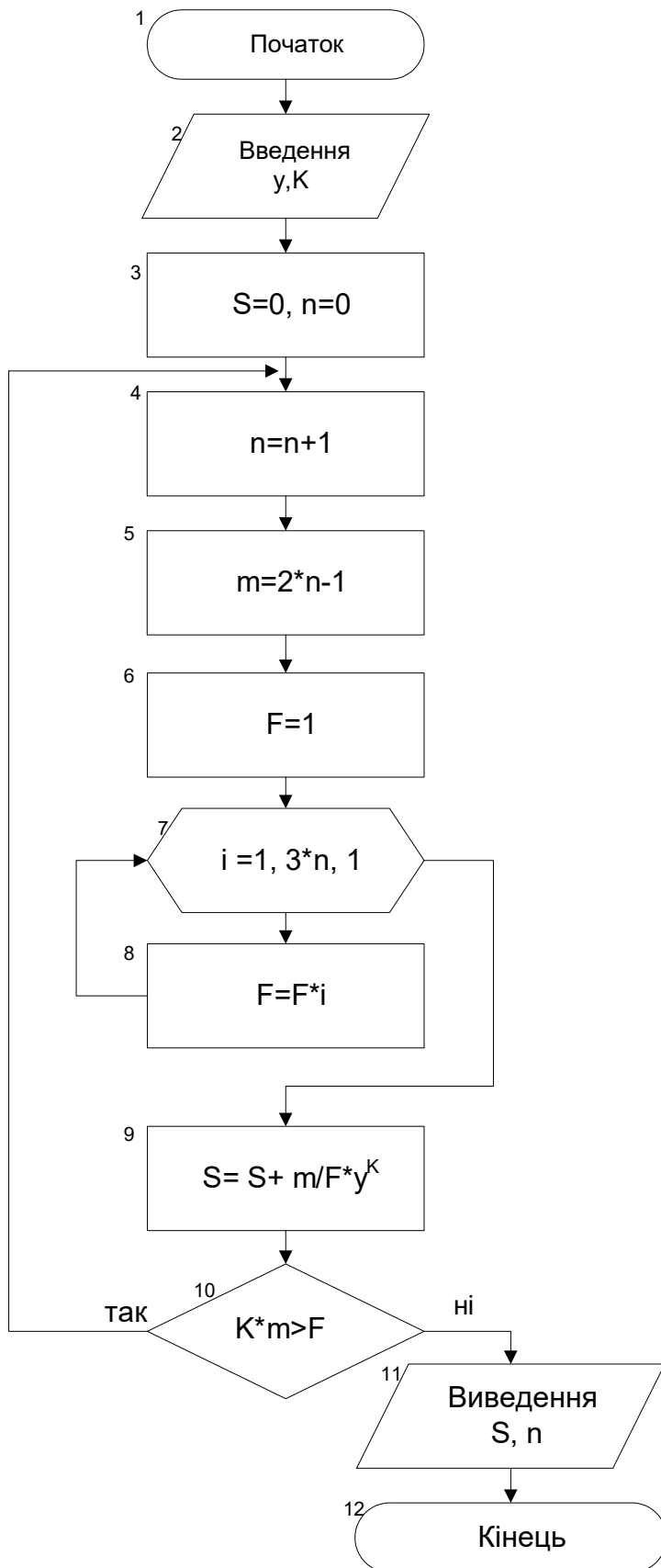


Рисунок 4.3 – Алгоритм розв’язання прикладу 4.3

2 Програма

```
CLS
  INPUT "y=";y
  INPUT "K=";K
  S=0 : n=0
DO
  n=n+1
  m=2*n-1
  F=1
FOR i=1 TO 3*n
  F = F * i
NEXT i
  S = S + m/F*y^K
LOOP WHILE K*m > F
  PRINT "S=";S, "n="; n
END
```

Варіанти самостійних завдань до лабораторної роботи 4

Вар.	Завдання
1	Обчислити та вивести результати на екран $S = \sum_{i=1}^{\infty} \frac{i!}{2 \cdot i}.$ Обчислення закінчити, коли значення S стане не менше від заданого числа E
2	Обчислити та вивести результати на екран $D = \prod_{n=1}^{\infty} \frac{1}{n!}$ Обчислення закінчити, якщо черговий елемент послідовності стане менше від заданого E
3	Обчислити та вивести результати на екран $S = \sum_{k=1}^{\infty} \frac{k}{(2 \cdot k)!}.$ Обчислення закінчити, якщо черговий елемент послідовності стане менше від заданого E
4	Обчислити та вивести результати на екран $y = \sum_{k=1}^{\infty} \frac{1}{(k+1)!}.$ Обчислення закінчити, якщо значення суми стане

	більше від заданого числа E
5	Обчислити та вивести результати на екран $D = \prod_{n=3}^{\infty} \frac{n+3}{(n+2)!}$ Обчислення закінчити, якщо черговий елемент $\frac{n+3}{(n+2)!}$ стане менше від заданого E
6	Обчислити та вивести результати на екран $D = \prod_{m=1}^{\infty} \frac{m+1}{(m+5)!}$ Обчислення закінчити, якщо черговий елемент $\frac{m+1}{(m+5)!}$ стане менше від заданого E
7	Обчислити та вивести результати на екран $y = 1 - x + \frac{x^2}{2} - \frac{x^3}{2 \cdot 3} + \frac{x^4}{2 \cdot 3 \cdot 4} - \dots + (-1)^k \cdot \frac{x^k}{k!} - \dots$ Обчислення закінчити, коли модуль останнього члена ряду стане менше від E
8	Обчислити та вивести результати на екран $D = \prod_{m=2}^{\infty} \frac{(2 \cdot m + 1)}{(m - 1)!}$ Обчислення закінчити, коли черговий елемент послідовності стане менше від заданого E
9	Обчислити та вивести результати на екран $S = \sum_{x=1}^{\infty} \frac{(2 \cdot x)!}{x!}$ Обчислення закінчити, коли в черговому елементі послідовності $(2 \cdot x)!/x!$ чисельник стане в K раз більше від знаменника
10	Обчислити та вивести результати на екран $S = \sum_{i=1}^{\infty} \frac{i}{2^i}$ Обчислення закінчити, коли чисельник дроби $i/2^i$ стане в K раз менше від знаменника
11	Обчислити та вивести результати на екран $S = \sum_{(n)} \frac{3 \cdot n - 1}{(2 \cdot n)!} y^k, n=1,2,3,\dots$ Обчислення закінчити, коли чисельник дроби $(3 \cdot n - 1)/(2 \cdot n)!$ стане в K раз менше від знаменника
12	Обчислити та вивести результати на екран

	$S = \sum_{p=1}^{\infty} \frac{1}{p^2}.$ <p>Обчислення закінчити, коли черговий доданок стане менше від заданого E</p>
13	<p>Обчислити та вивести результати на екран</p> $D = \prod_{(k)} \frac{k+1}{(k+4)!} \cdot x^k, \quad k = 4,5,6\dots$ <p>Обчислення закінчити, коли знаменник дроби (k+1)/(k+4)! стане перевищувати чисельник в N раз</p>
14	<p>Обчислити та вивести результати на екран</p> $y = \frac{1}{1!} + \frac{2}{2!} + \frac{3}{3!} + \dots + \frac{k}{k!} + \dots$ <p>Обчислення закінчити, коли модуль останнього члена ряду стане менше від E</p>
15	<p>Обчислити та вивести результати на екран</p> $y = \frac{2^x}{1!} + \frac{3^x}{2!} + \frac{4^x}{3!} + \dots + \frac{(n+1)^x}{n!} + \dots$ <p>Обчислення закінчити, коли модуль останнього члена ряду стане менше від E</p>

Порядок виконання лабораторної роботи

- 1 Побудувати алгоритм розв'язання задачі.
- 2 Запустити QBASIC.
- 3 Програма повинна містити введення вхідних даних, розрахунок формул та виведення результатів на екран.

Зміст звіту

У звіті повинно бути:

- 1) визначена мета завдання;
- 2) розв'язання завдання, згідно з варіантом (побудова алгоритму та програми). Програма повинна містити введення вхідних даних, обчислення та виведення результатів на екран;
- 3) результати обчислень;
- 4) висновки за лабораторною роботою.

Питання до захисту лабораторної роботи 4

- 1 Що таке ітераційний циклічний процес? Його відмінності від циклу з заданим числом повторень.
- 2 Які оператори організують цикл в програмі обчислення суми членів нескінченного ряду?

СПИСОК ЛІТЕРАТУРИ

ОСНОВНА

1 Філіппенко І.Г., Гончаров В.О., Меркулов В.С. Основи алгоритмізації: Конспект лекцій з дисципліни “Обчислювальна техніка та програмування”. – Харків.: УкрДАЗТ, 2005. – Ч. 2. – 54 с.

2 Філіппенко І.Г., Гончаров В.О., Меркулов В.С. Програмування інженерно-технічних задач в середовищі QBasic: Конспект лекцій з дисципліни “Обчислювальна техніка і програмування”. – Харків: УкрДАЗТ, 2007. – Ч. 3. – 132 с.

3 Новиков В.С., Пылькин А.Н. Начала программирования на языке QBasic: Учеб. пособие. – М.: Горячая линия-Телеком, 2007. – 268 с.

4 Острейковский В.А. Информатика: Учеб. пособие. – М.: Высш. шк., 2003. – 319 с.

5 Колмыкова Е.А., Кумскова И.А. Информатика: Учеб. пособие. – М.: Академия, 2005. – 416 с.

6 Острейковский В.А. Информатика: Учеб. пособие – М.: Высш. шк., 2003. – 319 с.

7 Сафронов И. К.Бейсик в задачах и примерах. – СПб.: БХВ-Петербург, 2003. – 320 с.

ДОПОМІЖНА

1 Информатика: Учеб. пособие / Под ред. А.П. Курносова. – М.: КолосС, 2005. – 270 с.

2 Информатика: Учеб. пособие / Под ред. И.А. Чернокустовой. – СПб.: Питер, 2005. – 270 с.

3 Истомин Е.П., Неклюдов С.Ю., Романченко В.И. Информатика и программирование: Учебник. – СПб.: ООО “Андреевский издат.дом”, 2006. – 248 с.

4 Информатика. Базовый курс: учеб. пособие / Под ред. С.В. Симоновича. – 2-е изд. – СПб.: Питер, 2007. – 639 с.

5 Дибкова Л.М. Информатика і комп'ютерна техніка: Навч. посібник. – 2-ге вид., перероб. – К.: Академвидав, 2007. – 415 с.

6 Сафронов И. Бейсик в задачах и примерах. – 2-е изд. –

СПб.: БХВ-Петербург, 2006. – 214 с.

7 Гутман Г. Изучаем BASIC. – СПб.: Питер, 2003. – 320 с.

8 Борисова М.В. Основы информатики и вычислительной техники: Учеб. пособие. – Ростов н/Д: “Феникс”, 2006. – 541 с.

9 Козырев А.А. Информатика: Учебник. – СПб.: Издат. Михайлова В.А., 2002. – 511 с.

10 Кетков Ю., Кетков А. Самоучитель. Практика программирования: Бейсик, Си-Паскаль. – СПб.: БХВ-Петербург, 2002. – 480 с.