

**УКРАЇНСЬКА ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ
ФАКУЛЬТЕТ АВТОМАТИКИ, ТЕЛЕМЕХАНІКИ ТА
ЗВ'ЯЗКУ**

Кафедра «Обчислювальна техніка та системи управління»

Завдання та методичні вказівки
до контрольної роботи
з дисциплін
«Обчислювальна техніка та програмування»,
«Інформаційні технології»
для студентів будівельного та механічного факультету всіх
форм навчання

Частина 1

Харків 2015

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри обчислювальної техніки та систем управління 26 лютого 2015 р., протокол № 6.

Рекомендуються для студентів механічного факультету для спеціальності 6.050601 «Теплоенергетика», будівельного факультету для спеціальностей 6.060101 «Будівництво», 6.070108 «Залізничні споруди та колійне господарство», 6.050503 «Машинобудування» всіх форм навчання.

Укладачі:

доценти І.В. Піскачова,
В.Г. Пчолін,
старш. викл. О.Є. Пенкіна

Рецензент

проф. С.В. Лістровий

ЗМІСТ

| | |
|---|----|
| Вступ..... | 4 |
| 1 Системи числення..... | 5 |
| Завдання 1. Переведення чисел з десяткової системи числення в систему числення з основою q | 5 |
| Завдання 2. Переведення чисел із системи числення з основою q у десяткову систему числення..... | 10 |
| 2 Алгоритмізація та програмування | 12 |
| Завдання 3. Проектування та програмування алгоритмів лінійних обчислювальних процесів..... | 17 |
| Завдання 4. Проектування та програмування алгоритмів розгалужених обчислювальних процесів..... | 19 |
| Завдання 5. Проектування та програмування алгоритмів циклічних обчислювальних процесів..... | 27 |
| Список літератури..... | 33 |

ВСТУП

Метою навчального процесу є підготовка фахівців, які мають навички постановки математичних задач, розроблення алгоритмів та програмування з використанням мов програмування високого рівня.

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

Інформатика – теоретична та прикладна (технічна, технологічна) дисципліна, що вивчає структуру і загальні властивості інформації, а також методи і (технічні) засоби її створення, перетворення, зберігання, передачі та використання в різних галузях людської діяльності. Основне теоретичне завдання інформатики полягає у визначенні загальних закономірностей, відповідно до яких створюється інформація, відбувається її перетворення, передавання та використання у різних сферах діяльності людини.

Широке впровадження засобів обчислювальної техніки (ОТ), автоматизація, комп'ютеризація у всіх сферах людської діяльності (сучасної науки та виробництва) потребує грамотної експлуатації ОТ та систем управління, що немислимо без знання апаратних засобів ОТ, принципів їх роботи, математичного й програмного забезпечення сучасних ЕОМ.

Автори методичних вказівок до контрольних робіт 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 QBasic, вибір якої зумовлений її простотою.

Методичні вказівки призначені для виконання студентами лабораторних робіт денної і заочної форм навчання «Інформаційні технології» для спеціальності 6.050601

«Теплоенергетика» та «Обчислювальна техніка та програмування» для спеціальностей 6.060101 «Будівництво», 6.070108 «Залізничні споруди та колійне господарство», 6.050503 «Машинобудування».

1 СИСТЕМИ ЧИСЛЕННЯ

ЗАВДАННЯ 1. Переведення чисел з десяткової системи числення в систему числення з основою q

Короткі теоретичні положення

Сучасні обчислювальні машини оперують із інформацією, поданою в цифровій формі. Інформація як відомості про об'єкт або явище відображається у вигляді конкретних даних, що подані у буквено-цифровій, числовій, текстовій, звуковій, графічній або іншій зафіксованій формі. Дані можуть передаватися, оброблятися, зберігатися.

Інформацію можна викласти в різноманітних формах: від природних для людини сигналів (звуків, жестів) до їх письмових позначень. Наочним прикладом перетворення форми подання інформації може бути переклад з однієї природної мови спілкування на іншу. Для запису слів використовується алфавіт – набір символів, що дає змогу кожному слову поставити у відповідність визначену послідовність символів – букв, тобто можна сказати, що кожне слово кодується.

Залежно від того, де і яким чином подається інформація, використовується відповідне кодування. Так, для запису (кодування) чисел у десятковій системі числення використовуються 10 символів. Для запису слів – букви.

Для кодування інформації в комп'ютері найзручніше (з технічних причин) використовувати мову, алфавіт якої містить усього два символи. Їх умовно позначають нулем та одиницею, а мову цю називають мовою двійкових кодів. За допомогою цих символів можна подати все розмаїття інформації. Одиницею виміру інформації є біт (0 або 1). 8 бітів складають байт. Значення

байта залежить від того, які позиції та у якій послідовності у ньому займають нулі та одиниці.

Числові дані перетворюються у двійкову систему числення, а як проміжні системи числення (СЧ) використовуються вісімкова й шістнадцяткова системи.

Система числення (numeration system) – система позначень для подання чисел.

Існують позиційні й непозиційні системи числення.

У непозиційних системах значення числового знака не залежить від його позиції в записі числа. Так, у римській системі числення в числі XXXII (тридцять два) – цифра X у будь-якій позиції дорівнює просто десяти.

ПОЗИЦІЙНА СИСТЕМА ЧИСЛЕННЯ (POSITIONAL SYSTEM) – запис числа, у якому значення числового знака залежить від його розташування в записі.

Наприклад, число 385,755 можна записати у вигляді виразу:

$$300 + 80 + 5 + 0,7 + 0,05 + 0,005 = 3 \cdot 10^2 + 8 \cdot 10^1 + 5 \cdot 10^0 + 7 \cdot 10^{-1} + 5 \cdot 10^{-2} + 5 \cdot 10^{-3}.$$

Основа позиційної системи числення – кількість різних цифр, що використовуються для зображення чисел у даній СЧ.

За основу системи можна прийняти будь-яке натуральне число — два, три, чотири та ін. Запис чисел у кожній із СЧ з основою q означає скорочений запис виразу:

$$a_{n-1} q^{n-1} + a_{n-2} q^{n-2} + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m}, \quad (1)$$

де a_i — цифри системи числення;

n, m — число цілих і дробових розрядів відповідно.

Так, десяткова система числення має тільки десять цифр – 0, 1, 2, ..., 9, за допомогою яких можливо подати будь-яке число. Десяткова система є позиційною, тобто значення кожної цифри числа визначається її місцем (позицією) у числі.

Правила запису й виконання різних операцій у всіх позиційних СЧ однакові та відрізняються одне від одного тільки основою (таблиця 1):

- у десятковій системі числення використовується десять цифр – 0 ... 9;
- у двійковій системі числення використовуються дві цифри – 0, 1;
- у вісімковій системі – вісім цифр – 0 ... 7;
- у шістнадцятковій системі числення задіяні шістнадцять символів – цифри 0 ... 9 і букви латинського алфавіту A,B,C,D,E,F для запису чисел 10, 11, 12, 13, 14, 15 відповідно.

Приклад запису числа у двійковій СЧ:

$$1101,11 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}.$$

Таблиця 1 – Натуральні числа, записані в десятковій, двійковій, вісімковій та шістнадцятковій СЧ

| 10 | 2 | 8 | 16 |
|-----------|----------|----------|-----------|
| 0 | 0000 | 0 | 0 |
| 1 | 0001 | 1 | 1 |
| 2 | 0010 | 2 | 2 |
| 3 | 0011 | 3 | 3 |
| 4 | 0100 | 4 | 4 |
| 5 | 0101 | 5 | 5 |
| 6 | 0110 | 6 | 6 |
| 7 | 0111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |
| 16 | 10000 | 18 | 10 |
| 17 | 10001 | 20 | 11 |
| ... | ... | ... | ... |

У процесі налагодження програм та в деяких інших ситуаціях у програмуванні актуальною є проблема переведення чисел з однієї позиційної СЧ в іншу.

Алгоритм перекладу цілих десяткових чисел у СЧ з основою q реалізується за допомогою виконання нижченаведених кроків:

1 Поділити вихідне десяткове число на число, що дорівнює значенню основи q . Зафіксувати остачу від ділення.

2 Послідовно повторювати крок 1 для одержуваних у процесі ділення остач до одержання нульової остачі на черговому кроці.

3 Подати остачі від ділення в СЧ з основою q .

4 Записати отримані остачі зліва направо в порядку, зворотному їх одержанню.

Приклад 1.1. Перевести десяткове число 75 у двійкову систему числення ($q = 2$).

Послідовність ділення:

$$\begin{aligned}75 : 2 &= 37 && \text{(остача 1);} \\37 : 2 &= 18 && \text{(остача 1);} \\18 : 2 &= 9 && \text{(остача 0);} \\9 : 2 &= 4 && \text{(остача 1);} \\4 : 2 &= 2 && \text{(остача 0);} \\2 : 2 &= 1 && \text{(остача 0);} \\1 : 2 &= 0 && \text{(остача 1).}\end{aligned}$$

Записуємо остачі від ділення, починаючи з останньої, одержуємо число 75 у двійковій СЧ 1001011.

Приклад 1.2. Перевести десяткове число 75 в шістнадцяткову СЧ ($q = 16$).

Послідовність ділення:

$$\begin{aligned}75 : 16 &= 4 \text{ (остача 11, або В);} \\4 : 16 &= 0 \text{ (остача 4).}\end{aligned}$$

Десяткове число 75 у шістнадцятковій СЧ має значення 4В.

Алгоритм переведення дробових десяткових чисел (менші одиниці) у СЧ з основою q полягає в такому:

1 Помножити вихідне десяткове число на основу q. Зафіксувати цілу частину отриманого добутку.

2 Послідовно повторювати пункт 1 для отриманих добутків. (Перед кожним множенням цілу частину попереднього результату треба обнулити).

3 Завершити процес послідовних множень або при одержанні нульової дробової частини в черговому добутку, або при досягненні необхідної точності (число множень визначає число знаків дробової частини числа в СЧ з основою q).

4 Праворуч від коми записати зафіксовані цілі частини в тій послідовності, у якій вони отримані.

Приклад 1.3. Перевести десяткове число 0,7 у двійкову систему числення з п'ятьма знаками після коми. Послідовні множення дають такі результати:

$$0,7 * 2 = 1,4 \text{ (ціла частина 1)};$$

$$0,4 * 2 = 0,8 \text{ (ціла частина 0)};$$

$$0,8 * 2 = 1,6 \text{ (ціла частина 1)};$$

$$0,6 * 2 = 1,2 \text{ (ціла частина 1)};$$

$$0,2 * 2 = 0,4 \text{ (ціла частина 0)};$$

...

Таким чином, результат має вигляд: 0,10110.

У тому випадку, коли вихідне число містить як дробову, так і цілу частини, треба відповідно до вищеописаних методів перевести окремо з десяткової СЧ у СЧ із основою q цілу й дробову частини вихідного числа, а потім записати їх відповідно ліворуч і праворуч від коми в результуючому числі.

Отже, результат переведення десяткового числа 75,7 у двійкову систему має вигляд: 1001011,10110...

Варіанти до завдання 1

Перевести десяткове число в СЧ з основою q.

| Варіант | Число | q |
|---------|-------|---|
| 0 | 67,43 | 2 |
| 1 | 89,12 | 2 |
| 2 | 46,32 | 2 |
| 3 | 39,55 | 2 |
| 4 | 73,12 | 2 |
| 5 | 81,54 | 2 |
| 6 | 61,33 | 2 |
| 7 | 87,32 | 2 |
| 8 | 51,32 | 2 |
| 9 | 83,52 | 2 |

ЗАВДАННЯ 2. Переведення чисел із системи числення з основою q у десяткову систему числення

Короткі теоретичні положення

Алгоритм переведення чисел зі СЧ із основою q у десяткову СЧ полягає у виконанні нижченаведених дій:

1 Записати вихідне число у вигляді полінома відповідно до формули (1).

2 Підставити замість літерних позначень значення відповідних коефіцієнтів. При записі всі коефіцієнти записувати в десятковій СЧ.

3 Зробити обчислення.

Приклад 2.1. Перевести в десяткову систему числення двійкове число 101,01.

$$\begin{aligned} 101,01_2 &= 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = \\ &= 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 + 0 \cdot 0,5 + 1 \cdot 0,25 = 5,25_{10} \end{aligned}$$

Приклад 2.2. Перевести в десяткову систему шістнадцяткове число E3C.

$$E3C_{16} = 14 \cdot 16^1 + 3 \cdot 16^0 + 12 \cdot 16^{-1} = 14 \cdot 16 + 3 + 12 \cdot 0,0625 = \\ = 224 + 3 + 0,75 = 227,75_{10} .$$

Варіанти до завдання 2

Перевести число з основою q у десяткову СЧ.

| Варіант | q | Число |
|---------|---|------------|
| 0 | 2 | 101110,101 |
| 1 | 2 | 110101,110 |
| 2 | 2 | 110011,001 |
| 3 | 2 | 111001,011 |
| 4 | 2 | 101101,111 |
| 5 | 2 | 100011,010 |
| 6 | 2 | 111010,100 |
| 7 | 2 | 101010,011 |
| 8 | 2 | 111100,101 |
| 9 | 2 | 110110,110 |

У контрольній роботі по завданнях 1 та 2 повинно бути:

- 1) мета завдання;
- 2) повний текст завдання з вихідними даними у відповідності до свого варіанта;
- 3) перелічені основні способи переведення числа з однієї СЧ в іншу;
- 4) наведено докладне розв'язання варіанта контрольної роботи із записом процесу розв'язання й проміжних результатів.

2 АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ

Короткі теоретичні положення

В основі кожної комп'ютерної програми лежить алгоритм. *Алгоритм* – упорядкований скінчений набір чітко визначених правил для розв'язування задач за кінцеву кількість кроків.



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.


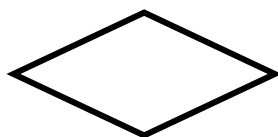



Для складення алгоритмів найчастіше використовують блок-схеми. Блок-схема – це графічна побудова, на якій за допомогою геометричних фігур показані кроки розв'язання задачі. Для складання блок-схем застосовують фігури, зображені в таблиці 2.

У процесі розроблення алгоритму програміст рисує блок-схему, позначаючи кожну дію одним з 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.

Таблиця 2 – Деякі графічні елементи блок-схеми

| Блок | Графічне позначення блоку | Дії, що виконуються |
|--------------------|---|---|
| 1 | 2 | 3 |
| Початок-закінчення |  | Початок, кінець, переривання процесу чи обробки виконання програми |
| Процес |  | Виконання операції, в результаті якої змінюється значення, форма чи подання даних |

Продовження таблиці 2

| 1 | 2 | 3 |
|-------------------------------|--|---|
| Модифікація (заголовок циклу) |  | Виконання операцій, що змінюють команди, групи команд або програму |
| 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 міжсторінковий |  | Розрив лінії потоку між сторінками |

Мова програмування QBasic

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 3.

Таблиця 3 – Деякі математичні функції мови QBasic

| Функція | Опис функції |
|----------|--|
| 1 | 2 |
| 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 більшим від нуля |
| SQR(м.в) | Квадратний корінь математичного виразу. Значення виразу повинно бути більшим або дорівнювати нулю |
| TAN(м.в) | Тангенс математичного виразу. Значення виразу розглядається як кут у радіанах |

Продовження таблиці 3

| 1 | 2 |
|----------|--|
| 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 |

Математичні вирази в будь-якій мові програмування записуються в один рядок за допомогою спеціальних команд. Спеціальні команди дають вказівки комп'ютеру при розрахунках обчислювати ту або іншу математичну функцію, наприклад, синус або квадратний корінь (таблиця 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 4 – Правила набору математичних виразів у QBasic

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

Продовження таблиці 4

| 1 | 2 |
|---------------|----------------------------------|
| $y=\sin x^2$ | $y=\text{SIN}(x^2)$ |
| $y=\ln(x)$ | $y=\text{LOG}(x)$ |
| $y=\lg(x)$ | $y=\text{LOG}(x)/\text{LOG}(10)$ |
| $y=\log_n(x)$ | $y=\text{LOG}(x)/\text{LOG}(n)$ |

Деякі оператори мови QBasic

Виведення результату обчислень на екран:

Оператор PRINT має таку структуру:

PRINT параметр,

де, наприклад,

PRINT В — оператор

PRINT — ключове слово

В — параметр.

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

Програму рекомендується 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.

Оператор END — ще один приклад ключового слова — зупиняє виконання програми. При 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 й після операторів.

Розміщення кількох операторів у рядку

Щоб скоротити кількість рядків програми, ви можете розмістити кілька коротких операторів у рядку, треба розділяти їх знаком двокрапки (:).

Уведення даних:

1) *оператор присвоєння*

У загальному вигляді його можна подати таким чином:

ім'я змінної = вираз,

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

$$D = (-A + \text{SIN}(C))$$
$$C = 2$$

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

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 курсора після натискання клавіші введення Enter.

ЗАВДАННЯ 3. Проектування та програмування алгоритмів лінійних обчислювальних процесів

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

Приклад 3.1. Обчислити значення W і R, використовуючи формулу

$$W = \operatorname{atg}^3 x^2 + \sqrt{\frac{z^2}{a^2 + x^2}},$$

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

Розв'язання

Блок-схема програми показана на рисунку 1.

Опис блок-схеми.

Блок 1 – початок розрахунків.

Блок 2 – уведення вихідних даних.

Блок 3 – розрахунок W.

Блок 4 – виведення результату на екран.

Блок 6 – завершення розрахунків.

Програма надана нижче:

```
REM Завдання 1
CLS
a=0.59: z=-4.8: x=2.1
W= a*TAN(x^2)^3+SQR(z^2/(a^2+x^2))
PRINT «W=»; W
END
```

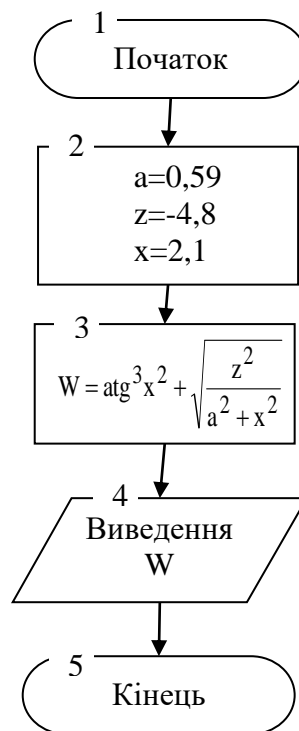


Рисунок 3.1 – Блок-схема для розв’язання прикладу 3.1

Варіанти до завдання 3

Скласти алгоритм, програму обчислення та виведення на екран значень нижченаведених виразів для довільних вихідних даних, що вводяться із клавіатури ПК у діалоговому режимі.

У контрольній роботі по завданню 3 повинно бути:

- 1) мета завдання;
- 2) повний текст завдання з вихідними даними у відповідності до свого варіанта;
- 3) алгоритм та програма для обчислення варіанта контрольної роботи.

| Варіант | Завдання |
|---------|---|
| 0 | $z = \left (x+1)^2 - y^3 \right \cdot \sin x$ |
| 1 | $z = 2 \cdot x - y \cdot \ln(x^2 + y)$ |
| 2 | $z = 3 \cdot \left \frac{x-y}{x^2+1} \right \cdot \operatorname{tg} \left(\frac{x}{y^2+2} \right)$ |
| 3 | $z = \left (3 \cdot x - y)^3 \right \cdot \operatorname{arctg} x^2$ |
| 4 | $z = x + y \cdot (7 \cdot x + 5 \cdot y) \cdot \operatorname{arctg}^3 x$ |
| 5 | $z = e^{-x} \cdot \frac{x+y}{x^2+y^2+1} + x-y $ |
| 6 | $z = \left (3 \cdot x - y)^3 \right \cdot \operatorname{tg}^2 x + 3 \sqrt{\frac{x}{y^4+1}}$ |
| 7 | $z = \left (x^3 - y)^5 \right \cdot \cos^2 x^3 + \frac{x^3}{y^2 + x + 1}$ |
| 8 | $z = x^3 - 10 \cdot \sin^5 x^3 + \frac{x+y}{y^2+10}$ |
| 9 | $z = x-y + \lg^2(x^2 + 0,5) + \sqrt{\frac{x^4+1}{y^4+1}}$ |

ЗАВДАННЯ 4. Проектування та програмування алгоритмів розгалужених обчислювальних процесів

Короткі теоретичні положення

Розгалужений алгоритм — це структура, яка розглядається як вибір дій у разі виконання або невиконання заданої умови.

Галуження бувають повними і неповними. Повне галуження — це галуження, у якому певні дії визначені й у разі виконання, і в разі невиконання умови. Неповне галуження — це

розгалуження, у якому дії визначені тільки у разі виконання (або у разі невиконання) умови.

Оператори умовного переходу – одні з найважливіших і часто використовуваних елементів у 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.

В умовах можна використовувати такі знаки порівняння:

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

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

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

Умова може задаватися досить складно і містити декілька блоків. Для уведення таких виразів використовуються спеціальні оператори – AND, OR, NOT.

Приклад 4.1. Увести число X, якщо воно менше або дорівнює 5, помножити на A та вивести результат на екран.

Розв'язання

Алгоритм у вигляді блок-схеми має вигляд, наведений на рисунку 2.

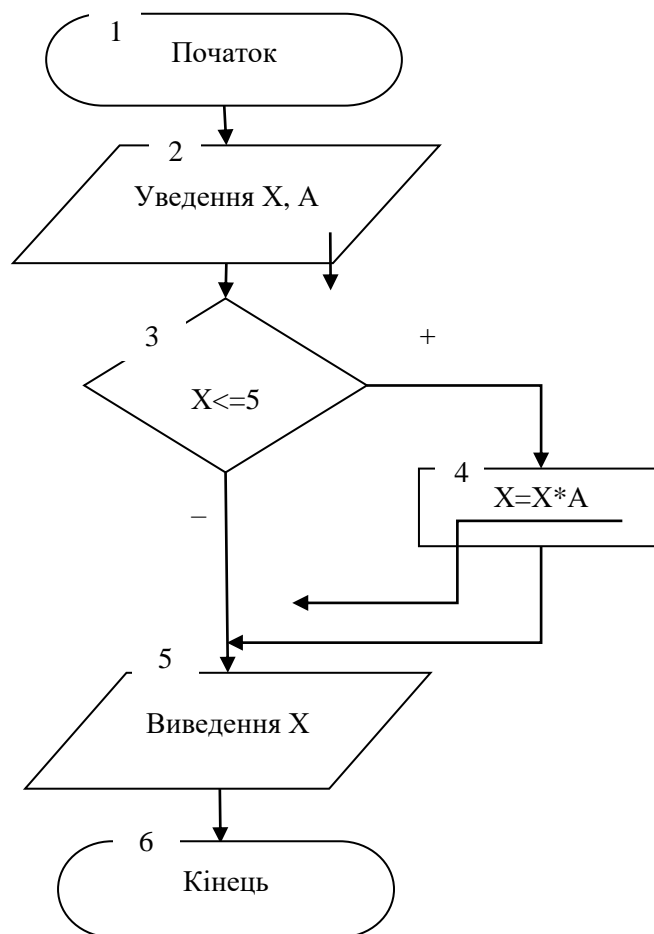


Рисунок 2 – Блок-схема до прикладу 4.1

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.

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 `IF ... THEN ... ELSE` дає змогу виконати або першу або другу дії залежно від умови.

Блок-схема формату `IF ... THEN ... ELSE ...` наведена на рисунку 3.

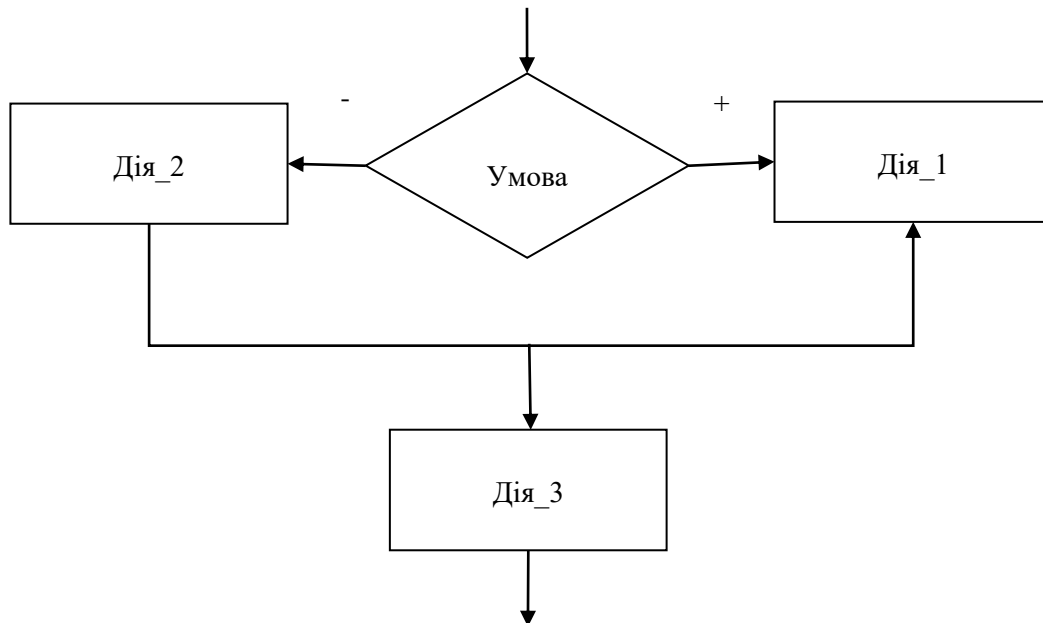


Рисунок 3 – Блок – схема конструкції `IF ... THEN ... ELSE...`

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 4.2. 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 4.

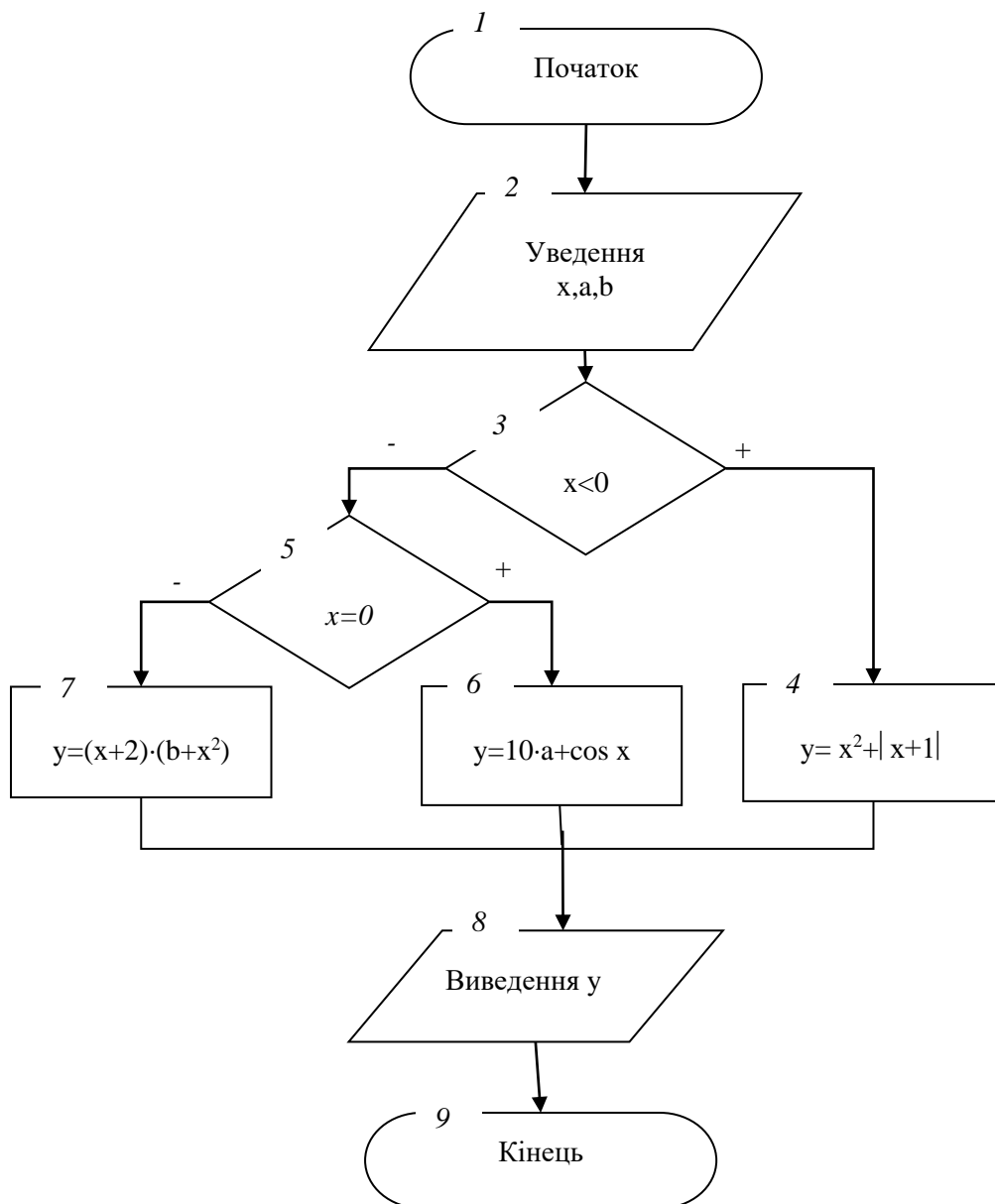


Рисунок 4 – Блок-схема до прикладу 4.2

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

Варіанти до завдання 4

Побудувати алгоритм і програму обчислення та виведення на екран значень нижченаведених виразів для вихідних даних (дані підібрати самостійно), що вводяться із клавіатури ПК у діалоговому режимі.

| Варіант | Завдання |
|---------|---|
| 0 | $y = \begin{cases} 3 \cdot x^2 + 1, & \text{якщо } x < 0, \\ y + \cos x, & \text{якщо } x = 0, \\ y - \sin x, & \text{якщо } x > 0 \end{cases}$ |
| 1 | $y = \begin{cases} 2 \cdot x + 1, & \text{якщо } x < 1, \\ 3/x + \sqrt{x^2 + 1}, & \text{якщо } x = 1, \\ (4 + x)/\sqrt{x^3 + 1}, & \text{якщо } x > 1 \end{cases}$ |
| 2 | $z = \begin{cases} x + \sin y, & \text{якщо } x - y > 0, \\ (x - 2)^2 + 6 \cdot y, & \text{якщо } x - y < 0, \\ \operatorname{tg} x + \sqrt{3 + y}, & \text{якщо } x - y = 0 \end{cases}$ |
| 3 | $z = \begin{cases} x^2 + 3 \cdot x, & \text{якщо } x \leq 0, \\ \sqrt{x^2 + 4 \cdot x}, & \text{якщо } x > 0 \end{cases}$ |
| 4 | $z = \begin{cases} x - y , & \text{якщо } x > y, \\ y + \sqrt{x^2 + 1}, & \text{якщо } x = y, \\ 2 \cdot y / \sqrt{x^2 + 1}, & \text{якщо } x < y \end{cases}$ |
| 5 | $z = \begin{cases} 15 \cdot x^y - 8 \cdot y^x, & \text{якщо } x > y, \\ 8 \cdot y + x, & \text{якщо } x = y, \\ y^2 + 8 \cdot \sqrt{2^x}, & \text{якщо } x > y \end{cases}$ |
| 6 | $z = \begin{cases} y^2 + 3 \cdot x^y, & \text{якщо } y \leq 0, \\ \sqrt{y^2 + 5 \cdot x^4}, & \text{якщо } y > 0 \end{cases}$ |
| 7 | $z = \begin{cases} 18 \cdot x^y , & \text{якщо } x < 2 \cdot y, \\ 7, & \text{якщо } x = 2 \cdot y, \\ 5 \cdot \sqrt{x^2 + 6}, & \text{якщо } x > 2 \cdot y \end{cases}$ |

| Варіант | Завдання |
|---------|---|
| 8 | $y = \begin{cases} 3 \cdot x ^3 + 5 \cdot x^2, & \text{якщо } x < 1, \\ 5, & \text{якщо } x = 1, \\ 2 / \sqrt{x^2 + 1}, & \text{якщо } x > 1 \end{cases}$ |
| 9 | $y = \begin{cases} y^3 + x^2 + 2, & \text{якщо } y < 0, \\ 3 \cdot x + \sqrt{x^2 + 1}, & \text{якщо } y = 0, \\ \frac{2 \cdot x}{\sqrt{x^2 + y}}, & \text{якщо } y > 0 \end{cases}$ |

У контрольній роботі по завданню 4 повинно бути:

- а) мета завдання;
- б) обчислення завдання, згідно з варіантом (алгоритм та програма);
- в) результати обчислень.

ЗАВДАННЯ 5. Проектування та програмування алгоритмів циклічних обчислювальних процесів

Короткі теоретичні положення

Оператори циклу застосовуються для розв'язання задач, у яких потрібно багаторазово виконувати деякі дії (оператори). В VBA використовуються три оператори циклу: For...Next, Do...Loop і While...Wend. Перший зручний у тих випадках, коли заздалегідь відомо скільки разів потрібно виконати дію. Другий і третій застосовують у тих алгоритмах, коли не відомо скільки разів повинен відробити цикл і програма виконує дії, поки не буде виконана певна умова.

Оператор For...Next (параметричний цикл).

Його формат виглядає так:

```
For лічильник = Поч_значення To Кінц_значення [Step знач_кроку]
  [Блок операторів]
  [Exit For]
Next [лічильник]
```

For...Next повторює блок операторів доти, поки значення змінної-лічильника не стане більше (а якщо початкове значення більше кінцевого, то менше) кінцевого значення циклу. Після кожної ітерації (чергового виконання блоку операторів) до значення змінної-лічильника додається значення кроку. Якщо крок дорівнює 1, то рядок Step 1 можна не писати.

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 кінцевого. Якщо початкове значення більше кінцевого, то слід застосовувати від’ємний крок.

Exit For – команда на достроковий (примусовий) вихід з циклу. У результаті її дії керування відразу передається першому оператору нижче Next, незалежно від значення лічильника.

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.

Зображення оператора For...Next у блок-схемі показане на рисунку 5.

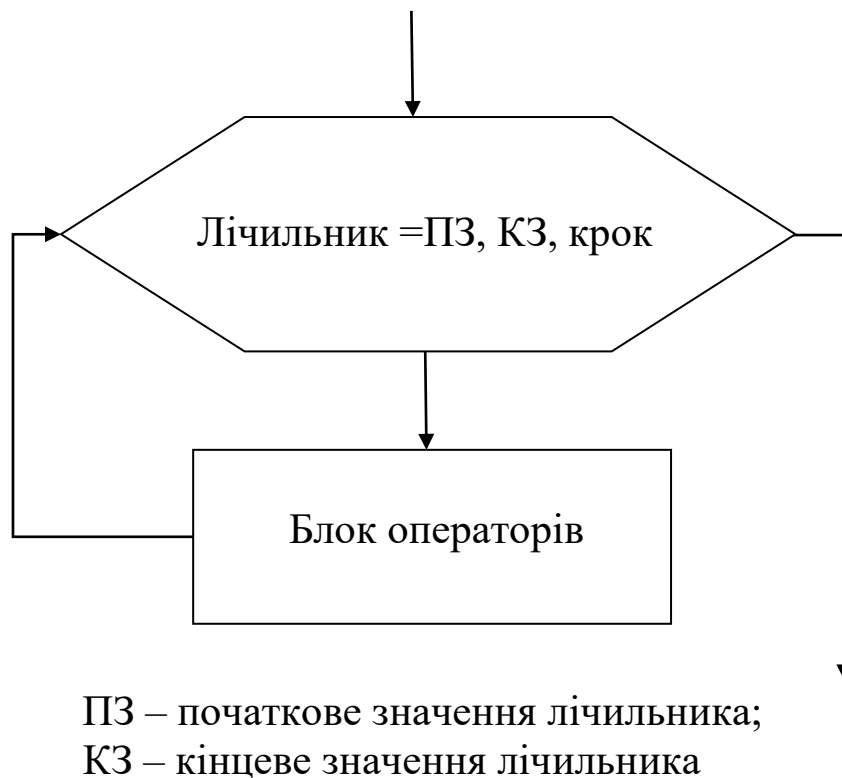


Рисунок 5 – Зображення оператора For...Next на блок-схемах

Якщо в процесі роботи циклу крок додатний, то цикл буде завершений, коли лічильник стане більше кінцевого значення. Якщо в процесі роботи крок від'ємний, то керування піде із циклу після того, як лічильник стане менше 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”.

Приклад 5.1. Розрахувати добуток кубів цілих чисел від 1 до n.
Розв’язання

```
P = 1
For x = 1 To n
  P = P * x ^ 3
Next x
Print P
```

Робота програми. Змінній присвоюється значення 1, інакше VBA приймає, що P дорівнює 0 та в кожній ітерації чергове

значення лічильника x буде множитися на 0 і загальний результат обчислень теж буде дорівнювати 0 . У кожній ітерації циклу значення лічильника збільшується на 1 , ця величина підноситься до куба і множить на поточне значення P . Потім отриманий результат записується в комірку оперативної пам'яті, яка зберігає величину P .

У математиці добуток чисел записується за допомогою знака Π . Зокрема добуток кубів цілих чисел від 1 до n запишемо так:

$$P = \prod_{1}^n x^3 .$$

Приклад 5.2. Протабулювати функцію

$$y = \begin{cases} \sin e^x, & \text{якщо } x \leq -3 \\ \cos x, & \text{якщо } -3 < x < 5, \\ 1, & \text{якщо } x \geq 5 \end{cases}$$

де a – будь-яке число; $x = -5; -4,5; -4; \dots; 9$.

Розв'язання

Позначимо початкове, кінцеве значення і крок зміни аргументу x відповідно $x_n = -5$; $x_k = 9$; $h = 0.5$

Побудуємо алгоритм для розв'язання задачі (рисунок 6).

Програмування алгоритму:

REM Табулювання функції

CLS

INPUT "Введіть а"; a

FOR x=-5 TO 9 STEP 0.5

IF X<=-3 THEN

y=SIN(EXP(x))

ELSEIF x≥5 THEN

y= 1

```

ELSE
  y= COS(x)
END IF
PRINT "x=";x, "y="; y
NEXT x
END

```

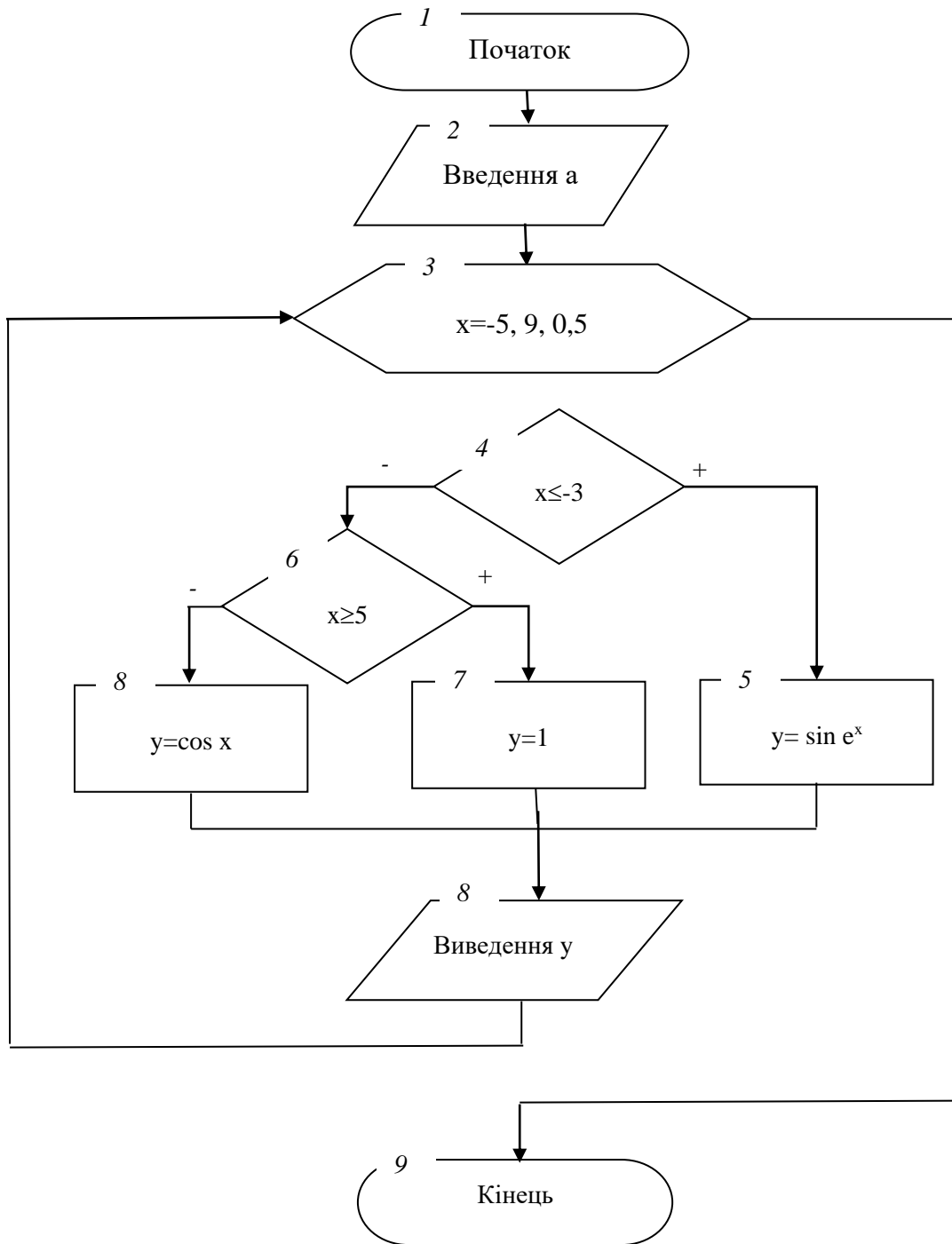


Рисунок 6

Варіанти до завдання 5

Побудувати алгоритм та програму обчислення значення функції, заданої в таблиці (протабулювати функцію). Здійснити виведення найменування змінних та обчисленого значення функції.

| Варіант | Вид функції | Умови виконання | Значення змінних | Інтервали |
|---------|--|---|--|--|
| 0 | $y = \begin{cases} a \cdot t^2 \cdot \ln t \\ 1 \\ e^{a \cdot t} \cdot \cos b \cdot t \end{cases}$ | $\begin{aligned} 1 \leq t \leq 2 \\ t < 1 \\ t > 2 \end{aligned}$ | $\begin{aligned} a = -0,5 \\ b = 2 \end{aligned}$ | $\begin{aligned} t \in [0;5] \\ h = 0,5 \end{aligned}$ |
| 1 | $y = \begin{cases} x^2 - 7 \cdot x^2 \\ a \cdot x^3 + 7 \cdot \sqrt{x} \\ \ln(x+7) \end{cases}$ | $\begin{aligned} x < 3 \\ x = 3 \\ x > 3 \end{aligned}$ | $a = 2$ | $\begin{aligned} x \in [0,5;2] \\ h = 0,1 \end{aligned}$ |
| 2 | $y = \begin{cases} a \cdot x^2 + b \cdot x + c \\ a/x + \sqrt{x^2 + 1} \\ (a + b \cdot x) \end{cases}$ | $\begin{aligned} x < 1 \\ x = 1 \\ x > 1 \end{aligned}$ | $\begin{aligned} a = 2 \\ b = -0,5 \\ c = 1 \end{aligned}$ | $\begin{aligned} x \in [0;3] \\ h = 0,1 \end{aligned}$ |
| 3 | $y = \begin{cases} x^2 - 5 \cdot a \cdot x \\ a \cdot x^3 + 7 \cdot \sqrt{x} \\ \ln(x) \end{cases}$ | $\begin{aligned} x < 4 \\ x = 4 \\ x > 4 \end{aligned}$ | $a = 1$ | $\begin{aligned} x \in [0;6] \\ h = 0,1 \end{aligned}$ |
| 4 | $y = \begin{cases} 1,8 \cdot a \cdot x \\ (x - 2)^2 + 6 \\ 3 \cdot \operatorname{tg} x \end{cases}$ | $\begin{aligned} x < 1 \\ 1 < x \leq 2 \\ x > 2 \end{aligned}$ | $a = 2,3$ | $\begin{aligned} x \in [0,2;2,8] \\ h = 0,2 \end{aligned}$ |
| 5 | $y = \begin{cases} x \cdot \sqrt[3]{x - a} \\ x \cdot \sin a \cdot x \\ e^{-a \cdot x} \cos a \cdot x \end{cases}$ | $\begin{aligned} x > a \\ x = a \\ x < a \end{aligned}$ | $a = 2$ | $\begin{aligned} x \in [0;3] \\ h = 0,1 \end{aligned}$ |

| Варіант | Вид функції | Умови виконання | Значення змінних | Інтервали |
|---------|---|---|------------------|------------------------------|
| 6 | $y = \begin{cases} b \cdot x + 2 \\ 1 \\ b \cdot x + \ln b \cdot x \end{cases}$ | $b \cdot x < 1$ $b \cdot x = 1$ $b \cdot x > 1$ | $b = 2,5$ | $x \in [0;1]$ $h = 0,1$ |
| 7 | $y = \begin{cases} \sin x \cdot \ln x \\ \cos^2 x \end{cases}$ | $x > 3,5$ $x \leq 3,5$ | – | $x \in [1;5]$ $h = 0,2$ |
| 8 | $y = \begin{cases} \ln(x+1) \\ \sin^2 \sqrt{ a \cdot x } \end{cases}$ | $x > 1$ $x \leq 1$ | $a = 2$ | $x \in [0;2]$ $h = 0,1$ |
| 9 | $z = \begin{cases} (\ln^3 x + x^2) / \sqrt{x+t} \\ \sqrt{x+t} + 1/x \\ \cos x + t \cdot \sin^2 x \end{cases}$ | $x < 0,5$ $x = 0,5$ $x > 0,5$ | $t = 2,2$ | $x \in [0,2;2]$ $h = 0,2$ |

У контрольній роботі по завданню 5 повинно бути:

- а) мета завдання;
- б) обчислення завдання згідно з варіантом (алгоритм та програма);
- в) результати обчислень.

СПИСОК ЛІТЕРАТУРИ

Основна

1 Філіппенко І.Г., Гончаров В.О., Меркулов В.С. Основи побудови ПК. Конспект лекцій з дисципліни “Обчислювальна техніка та програмування”. – Харків: УкрДАЗТ, 2005. – Ч. 1.– 38 с.

2 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.

3 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.

4 Шеховцов В.А. Операційні системи: Підручник. – К.: Видавнича група ВНУ, 2005. – 576 с.

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

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

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

8 Сафронов И.К. Бейсик в задачах и примерах. – С.Пб.: БХВ-Петербург, 2003. – 320 с.

Допоміжна

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

10 Информатика: Учеб. пособие / Под ред. И.А. Чернокустовой. – С.Пб.: Питер, 2005. – 270 с.

11 Глушаков С.В., Сурядный А.С. Персональный компьютер: Учеб. пособие. – 5-е изд., доп. и перераб. – Харьков: Фолис, 2003. – 500 с.

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

13 Информатика. Базовый курс: Учеб. пособие / Под ред. С.В. Симоновича. – 2-е изд. – С.Пб.: Питер, 2007. – 639 с.

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

15 Соломончук В. Аппаратные средства персональных компьютеров. – С.Пб.: БХВ-Петербург, 2003. – 512 с.

16 Сафронов И. Бейсик в задачах и примерах. – 2-е изд. – С.Пб.: БХВ-Петербург, 2006. – 214 с.

17 Гутман Г. Изучаем BASIC. – С.Пб.: Питер, 2003. – 320 с.

18 Борисова М.В. Основы информатики и вычислительной техники: Учеб. пособие. – Ростов н/Д: Фенікс, 2006. – 541 с.

19 Козырев А.А. Информатика: Учебник. – С.Пб.: Издат. Михайлова В.А., 2002. – 511 с.

20 Соломончук В. Аппаратные средства персональных компьютеров. – С.Пб.: БХВ-Петербург, 2003. – 512 с.

21 Гладкий А. Windows XP для всех. – С.Пб.: Питер, 2006.