

**Методичні вказівки до практичних занять
з дисципліни
«Основи початкового курсу інформатики з елементами
програмування»**

Тема 1. Системи числення

Позиційні системи числення. Двійкова і вісімкова системи числення.

Перевід чисел у різні системи числення.

Арифметичні операції у двійковій та вісімковій системах числення

Теоретична частина

Системою числення називається сукупність правил і знаків, за допомогою яких можна відобразити (кодувати) будь-яке невід'ємне число. До систем числення висуваються певні вимоги, серед яких найбільш важливими є вимоги однозначного кодування невід'ємних чисел $0, 1, \dots$ з деякої їх скінченної множини — діапазону P за скінченне число кроків і можливості виконання щодо чисел арифметичних і логічних операцій. Розрізняють такі типи систем числення: позиційні; змішані; непозиційні.

У **позиційних системах** числення одна і та ж цифра у записі числа набуває різних значень залежно від своєї позиції. Таким чином, позиція цифри має вагу у числі. Здебільшого вага кожної позиції кратна деякому натуральному числу b , $b > 1$, яке називається основою системи числення.

Наприклад, якщо b – натуральне число, то для представлення числа x у системі числення з основою b його подають у вигляді лінійної комбінації степенів числа b :

$$x = \sum_{k=0}^n a_k b^k, \text{ де } a_k \text{ — цілі, } 0 \leq a_k < b.$$

Основа – це кількість символів, що використовуються при записуванні чисел.

Приклад: число «сто п'ять» представляється у десятковій системі числення у вигляді: $105 = 1 \cdot 10^2 + 0 \cdot 10^1 + 5 \cdot 10^0$

Двійкова система числення — це позиційна система числення з основою; використовує для запису чисел тільки два символи: 0 та 1.

Вісімкова система числення — це позиційна система числення з основою 8. Для представлення чисел в ній використовуються цифри 0 до 7.

Перевід чисел у різні системи числення

1. з десяткової у двійкову та вісімкову:

90	2							
0	45	2						
	1	22	2					
	0	11	2					
	1	5	2					
	1	2	2					
	1							1

90	8		
2	11	8	
	1	1	

$$90_{(10)} = \mathbf{1111010}_{(2)} = \mathbf{112}_{(8)}$$

2. з двійкової та вісімкової у десяткову

$$1011101_{(2)} = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 64 + 16 + 8 + 4 + 1 = 93_{(10)}$$

$$147_{(8)} = 1 \cdot 8^2 + 4 \cdot 8^1 + 7 \cdot 8^0 = 64 + 32 + 7 = 103_{(10)}$$

Таблиця додавання

Таблиця віднімання

у двійковій системі числення

$$0+0=0$$

$$0-0=1-1=0$$

$$0+1=1+0=1$$

$$1-0=1$$

$$1+1=0$$

$$10-1=1$$

Таблиця додавання у вісімковій системі числення

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Додавання та віднімання «в стовпчик»

$$\begin{array}{r} + \\ 11000 \\ 1101 \\ \hline 100101 \end{array}$$

$$\begin{array}{r} - \\ 10110 \\ 1101 \\ \hline 1001 \end{array}$$

$$\begin{array}{r} + \\ 17503 \\ 4125 \\ \hline 23630 \end{array}$$

$$\begin{array}{r} - \\ 21533 \\ 4425 \\ \hline 15106 \end{array}$$

Завдання

Перевести числа $125_{(10)}$, $199_{(10)}$, $157_{(10)}$, $112_{(10)}$, $191_{(10)}$, $182_{(10)}$ у двійкову систему числення.

Перевести числа $1567_{(10)}$, $8723_{(10)}$, $1253_{(10)}$, $8932_{(10)}$, $6791_{(10)}$ у вісімкову систему числення.

Обчислити суму та різницю чисел:

$$7353_{(8)} \text{ і } 5645_{(8)}, 6477_{(8)} \text{ і } 4656_{(8)}, 4561_{(8)} \text{ і } 2765_{(8)}, 2346_{(8)} \text{ і } 1707_{(8)},$$

$$7625_{(8)} \text{ і } 4276_{(8)}, 2721_{(8)} \text{ і } 1773_{(8)}, 6721_{(8)} \text{ і } 2753_{(8)}.$$

$$110101101_{(2)} \text{ і } 111110_{(2)}, 101010101_{(2)} \text{ і } 100110_{(2)}, 101100111_{(2)} \text{ і } 100111_{(2)},$$

$$110101001_{(2)} \text{ і } 110110_{(2)}, 110111100_{(2)} \text{ і } 101011_{(2)}.$$

Тема 2. Алгоритми.

Властивості алгоритмів. Способи опису алгоритмів.

Блок-схеми. Лінійні алгоритми.

Теоретична частина.

Алгоритм — це точні розпорядження (вказівки, команди) виконавцеві відносно здійснення послідовності дій, які спрямовані на розв'язання певної задачі.

Властивості алгоритму — це вимоги, які повинен задовольняти алгоритм:

Дискретність.

Формальність

Скінченність.

Однозначність.

Масовість

Результативність.

Способи подання алгоритмів:





Словесний — подання алгоритмів, які призначені на виконання людиною (наприклад, кулінарних рецептів, правил переходу вулиці тощо), природною мовою в усній або письмовій формі.

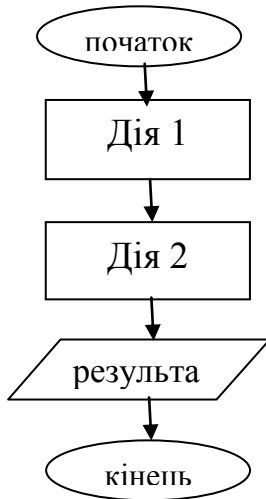
Формульно-словесний — подання алгоритмів у навчальній та науковій діяльності за допомогою мови математичних формул, хімічних процесів тощо зі словесними поясненнями природною мовою.

Графічний — подання алгоритмів у вигляді графічних схем (блок-схем або структурних схем) для спрощення розробки та аналізу алгоритмів, полегшення переходу від запису алгоритмів до написання програм.

Програмний — подання алгоритмів мовою програмування для їх подальшого опрацювання на комп'ютері.

Основні елементи схем алгоритму

Позначення	Функція
	Початок і кінець програми
	Виконання однієї або кількох послідовних операцій (зміна значення даних, форми подання, розташування)
	Обчислення (перевірка) умов
	Для обробки (введення) або відображення результатів



Лінійний алгоритм

Лінійний алгоритм — це алгоритм, який забезпечує отримання результату шляхом одноразового виконання послідовності дій незалежно від вхідних даних і проміжних результатів. Дії в таких алгоритмах виконуються послідовно одна за одною, тобто лінійно

Завдання

1. Дано два ненульових числа. Знайти їх суму, різницю, добуток і частку.
2. Дано два числа. Знайти середнє арифметичне їх квадратів і середнє арифметичне їх модулів.
3. Знайти периметр і площу прямокутного трикутника, якщо дано довжини його катетів a і b .
4. Дано довжину ребра куба. Знайти площу грані, площа повної поверхні і об'єм цього куба.
5. Знайти довжину кола і площу круга заданого радіуса R .
6. Знайти площу кільця, внутрішній радіус якого дорівнює R_1 , а зовнішній радіус дорівнює R_2 ($R_1 < R_2$).
7. Дано координати трьох вершин трикутника (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Знайти його периметр і площу.
8. Дано ціле чотиризначне число. Використовуючи операції div та mod , знайти суму його цифр.
9. Обчислити $z = \text{sign}(x) + \text{sign}(y) \text{sig}(x - y)$.

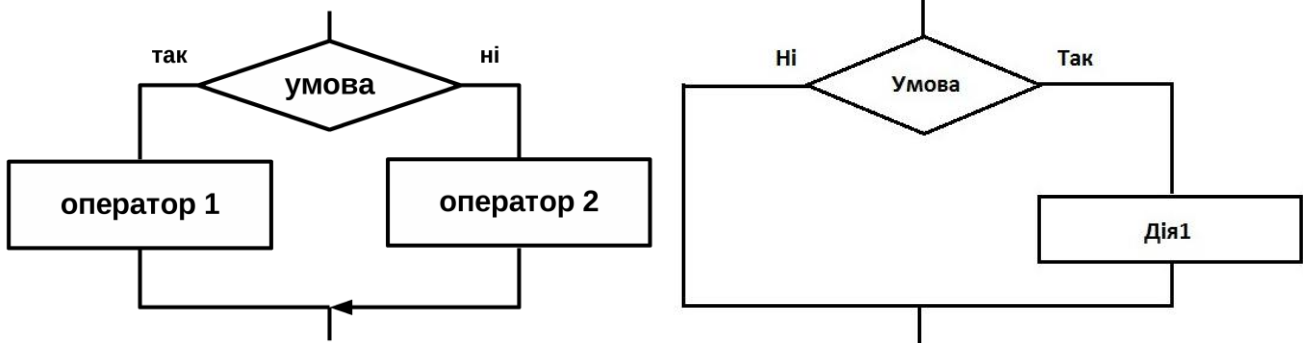
Тема 3. Алгоритми з галуженнями

Теоретична частина

Розгалужений алгоритм (галуження) — у класичному варіанті цей алгоритм розглядається як вибір однієї з двох альтернативних дій залежно від виконання заданої умови.

Повне галуження — це галуження, у якому визначені різні дії в разі виконання та невиконання заданої умови.

Неповне галуження — це галуження, у якому дії визначені тільки у разі виконання (або невиконання) заданої умови.



Завдання

1. Дано три цілих числа. Обчислити квадрати від'ємних чисел і куби додатних (число 0 не враховувати).
2. З двох (трьох) заданих чисел вибрати найменше (найбільше).
3. На числовій осі розташовані три точки: А, В, С. Визначити, яка з двох останніх точок (В або С) розташована ближче до А, і вивести цю точку і її відстань від точки А.
4. Для даного x обчислити значення наступної функції f , що приймає речові значення:
$$f(x) = \begin{cases} 3x-1, & x \leq 0, \\ x^2, & x > 0 \end{cases}$$
5. Дано ціле число, яке лежить в діапазоні від -999 до 999. Вивести рядок - словесний опис даного числа виду "від'ємне двозначне число", "нульове число", "додатне однозначне число".

Тема 4. Циклічні алгоритми

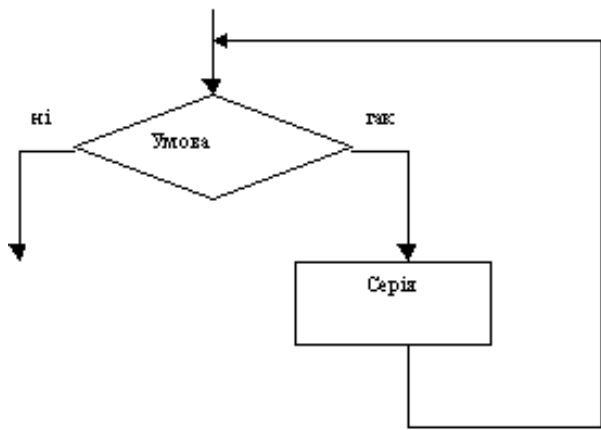
Теоретична частина

Структура повторення (циклу) — це структура, яка передбачає повторення деякої серії команд. Ця серія команд називається *тілом циклу*,

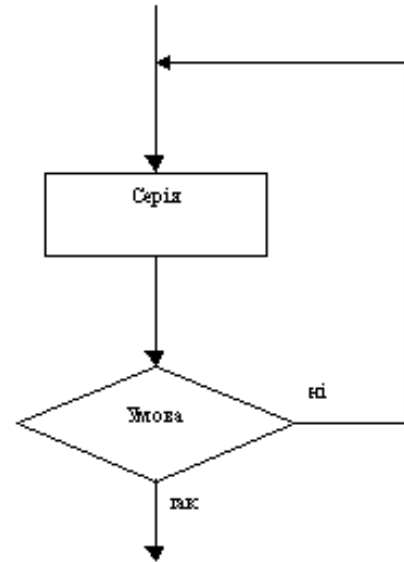
Структура циклу реалізується в трьох основних варіантах: цикл з передумовою, цикл з післяумовою, цикл з параметром, який можна розглядати і як окремий випадок циклу з передумовою.

Цикл з передумовою (цикл-«доки») — це цикл, у якому тіло циклу виконується тільки у разі виконання умови, заданої перед тілом циклу.

Цикл з передумовою реалізується таким чином. Перевіряється умова. Якщо вона виконується, то виконується тіло циклу. Знову перевіряється умова і т. д. Тіло циклу виконується, доки виконується умова. Цикл завершується, коли умова перестає виконуватися.



Цикл з передумовою



Цикл с післяумовою

Завдання

1. Дано два цілих числа A та B ($A < B$). Вивести всі цілі числа, розташовані між даними числами (включаючи самі ці числа), в порядку їх зростання, а також кількість N цих чисел.
2. Дано дійсне число A і ціле число N (> 0). Обчислити A^N .
3. Обчислити $N!$
4. Дано ціле число N (> 0). Якщо N - непарне, то обчислити добуток $1 \cdot 3 \cdot \dots \cdot N$; якщо N - парне, то обчислити добуток $2 \cdot 4 \cdot \dots \cdot N$.
5. Дано ціле число N і дві точки на числовій осі: A, B ($A < B$). Відрізок $[A, B]$ розбитий на рівні відрізки довжини H з кінцями в N точках виду $A, A + H, A + 2H, A + 3H, \dots, B$. Вивести значення H і набір з N точок, який утворює розбиття відрізка $[A, B]$.
6. Дано ціле число N (> 2) і дві точки на числовій осі: A, B ($A < B$). Функція $F(x)$ задана формулою $F(x) = 1 - \sin(x)$. Вивести значення функції F в N рівновіддалених точках, що утворюють розбиття відрізка $[A, B]$: $F(A), F(A + H), F(A + 2H), \dots, F(B)$.
7. Дано число D (> 0). Послідовність чисел A_N визначається наступним чином: $A_1 = 2, A_N = 2 + 1/A_{N-1}, N = 2, 3, \dots$. Знайти перший номер K , для яких виконується умова $|A_K - A_{K-1}| < D$, і вивести цей номер, а також числа A_{K-1} і A_K .
8. Дано N чисел. Вивести їх середнє арифметичне.
9. Дано ціле число N і набір з N дійсних чисел. Вивести суму і добуток чисел з даного набору.
10. Дано ціле число N і набір з N цілих чисел. Вивести в тому ж порядку всі парні (непарні) числа з даного набору і кількість K таких чисел.

11. Дано цілі числа K , N і набір з N цілих чисел. Якщо в наборі присутнє число, менше K , то вивести True; в іншому випадку вивести False.
12. Дано набір ненульових цілих чисел, який закінчується числом 0. Вивести кількість елементів у наборі.
13. Дано набір ненульових цілих чисел, який закінчується числом 0. Вивести суму всіх позитивних парних чисел з даного набору. Якщо необхідні числа в наборі відсутні, то вивести 0.
14. Дано ціле число N і набір з N цілих чисел. Вивести номери тих чисел в наборі, які менші свого лівого сусіда, і кількість K таких чисел.
15. Дано ціле число N і набір з N цілих чисел. Вивести номери тих чисел в наборі, які більші свого правого сусіда, і кількість K таких чисел.
16. Знайти мінімальний і максимальний з даних N елементів.
17. Знайти кількість мінімальних (максимальних) з даних N цілочисельних елементів.
18. Знайти номери мінімальних (максимальних) з даних N елементів.
19. Знайти мінімальний (максимальний) парний (непарний) з даних N ненульових цілочисельних елементів.
20. Дано числа a , b ($0 < a < b$) і набір з N елементів. Знайти мінімальний (максимальний) з елементів, що містяться в інтервалі (a, b) . Якщо необхідні елементи відсутні, то вивести -1.
21. Дано набір з N цілочисельних елементів. Знайти кількість елементів, розташованих після (перед) першого (останнього) мінімального (максимального).
22. Дано набір з N цілочисельних елементів. Знайти максимальну кількість мінімальних (максимальних) елементів, що розташовані поруч.
23. Вивести рядок довжини N (N - парне), який складається з символів S_1 і S_2 , що чергуються.
24. Дано послідовність символів, яка закінчується крапкою. Порахувати кількість малих і великих літер латинського алфавіту, які містить дана послідовність.
25. Дано послідовність символів, яка закінчується крапкою. Перевірити чи можна із цих символів скласти слово «залік».
26. Дано текст, який закінчується крапкою. Перевірити чи даний текст можна вважати записом числа у вісімковій системі числення.

Тема 5. Середовище Pascal ABC.

Оператор присвоювання, процедури read та write.

Прості лінійні програми

Теоретична частина

Основні поняття мови Паскаль

Поняття	Опис	Приклад
Алфавіт мови	Літери латинського алфавіту, цифри, спеціальні символи, зарезервовані слова.	A...Z, a...z, 0...9, + - * / = < > [] { } \$ ^ & #
Ідентифікатори	Імена об'єктів (констант, типів даних, змінних, функцій, програм). <i>Правила запису ідентифікаторів:</i> всі ідентифікатори складаються з літер латинського алфавіту, цифр, починаються з літери або знака, «_»; великі і малі літери не розрізняються.	A, B begin Write Test My_First_Program
Елементи даних	<i>Константа</i> - комірка пам'яті, значення якої протягом виконання програми залишається постійним.	45, Pi
	<i>Змінна</i> - комірка пам'яті, значення якої протягом виконання програми може змінюватися.	a, b, c
Коментар	Призначений для внесення до тексту програми пояснень. Обмежується символами { } або (**)	{Це коментар} (*Це коментар*)

Службові слова мови Pascal

and — і	end — кінець	nil — нуль	repeat —
array — масив	file — файл	not — ні	повторювати
begin — початок	for — для	of — з	set — множина
case — вибір	function — функція	or — або	then — то
const — сталі	goto — перейти до	packed - стиснутий	to — до
div — ділення без	if — якщо	procedure – роцедура	type — тип
остачі	in — в	program - програма	until — доки
do — виконати	label — позначка	record — запис	var — змінні
downto — униз до	mod — остача		while — доки та
else — інакше			інші.

Стандартні типи даних

Тип даних	Позначення	Опис	Обсяг пам'яті	Приклади
Ціло-чисельний тип	<i>Byte</i>	Цілі числа в інтервалі [0; 255]	1 байт	0, 5, 245
	<i>Shortint</i>	-128..127	1 байт	-5, 26, 96
	<i>Word</i>	0..65535	2 байта	0, 36, 6545
	<i>Integer</i>	Цілі числа в інтервалі [-32767; 32767]	2 байта	457, -568, -7, 0
	<i>longint</i>	-2147483648..2147483647	4 байта	-2356, 988456
Дійсний тип	<i>Single</i>	$1.5 \cdot 10^{-45}$ – $3.4 \cdot 10^{38}$	4 байта	236,36; 6954,369
	<i>Real</i>	Значення з плаваючою крапкою, що містить мантису (до 11 значущих цифр) і експоненту - степінь числа 10 (від 10^{-39} до 10^{38})	6 байтів	0.4,-1.8, 0.172E+3, 37E-4
	<i>Extended</i>	$3.4 \cdot 10^{-4951}$ – $1.1 \cdot 10^{4932}$	10 байт	0.4,-1.8, 0.172E+3,
Символьний тип даних	<i>Char</i>	Символьні константи містять один символ, який обмежено апострофами	1 байт	A', 'e', Y, '2'
Рядковий тип даних	<i>String</i>	Рядок символів довжиною не більше 255	Залежить від довжини	
Логічний тип даних	<i>Boolean</i>	Константи логічного типу мають два можливі значення: <i>True</i> (істинне) і <i>False</i> (хибне)	1 байт	5<3 - True 5+8>14- <i>False</i>

Операції над стандартними типами даних

Операції над цілими числами:

Операція	Позначення	Приклад
Додавання	+	X + Y
Віднімання	-	X-Y
Множення	*	X*Y
Цілочисельне ділення з визначенням цілої частини	div	X div Y

Цілочисельне ділення з визначенням дробової частини	mod	$X \bmod Y$
Рівність	=	$X=Y$
Не рівність	$\langle \rangle$	$X \langle \rangle Y$
Менше (порівняння)	<	$X < Y$
Менше або рівне (порівняння)	< =	$X \leq Y$
Більше (порівняння)	>	$X > Y$
Більше або рівне (порівняння)	> =	$X \geq Y$

Операції над дійсними числами:

Операція	Позначення	Приклад
Додавання	+	$X + Y$
Віднімання	-	$X - Y$
Множення	*	$X * Y$
Рівність	=	$X = Y$
Не рівність	$\langle \rangle$	$X \langle \rangle Y$
Менше (порівняння)	<	$X < Y$
Менше або рівне (порівняння)	< =	$X \leq Y$
Більше (порівняння)	>	$X > Y$
Більше або рівне (порівняння)	> =	$X \geq Y$

Функція	Тип аргументу	Тип результату	Математичний запис, коментар
abs(x)	integer, real	integer, real	$ x $
arctan(x)	integer, real	real	arctg x
cos(x)	integer, real	real	cosx
sin(x)	integer, real	real	sinx
exp(x)	integer, real	real	e^x
ln(x)	integer, real	real	lnx
sqrt(x)	integer, real	real	\sqrt{x}
sqr(x)	integer, real	integer, real	x^2
ord(x)	упорядкований	integer	ASCII-код символу
succ(x)	упорядкований	упорядкований	повертає наступне значення x
pred(x)	упорядкований	упорядкований	повертає попереднє значення x
round(x)	real	integer	заокруглює число x до цілого
trunc(x)	real	integer	відкидає дробову частину числа x
int(x)	real	real	відкидає цілу частину числа x
frac(x)	real	real	дробова частина числа x
odd(x)	integer	boolean	true (x – непарне), false (x – парне)

random(x)	integer	integer	Генерує випадкове число з діапазону від 0 до x
upcase(x)	char	char	замінює малу літеру латинської абетки на велику
Процедури:			
inc(x,y)	integer	integer	збільшує x на y
inc(x)	integer, char	integer, char	збільшує x на 1
dec(x,y)	integer	integer	зменшує x на y
dec(x)	integer, char	integer, char	зменшує x на 1

Деякі оператори

Оператор	Загальний вигляд	Опис	Приклади
:= Присвоювання	Ідентифікатор:= вираз	Присвоює змінній конкретне значення, заповнюючи комірку пам'яті, відведену для змінної, новим значенням, відночас знищуючи старе	A:=2 X:= Y + 2 - Z Name:= 'Іра'
Read Введення даних з клавіатури	Read(a ₁ , a ₂ ,..., a _n), де a ₁ , a ₂ , ...,a _n – змінні зазначених типів, яким присвоюються значення, що вводяться	Програма зупиняється і чекає введення необхідної кількості даних. Числа при введенні поділяються пробілами або натиском клавіші <Enter>. Введення закінчується натиском клавіші <Enter>	Read (Name) Read (A, B, Y, Z)
Readln		Після введення значень a ₁ , a ₂ , ...,a _n курсор переводиться на наступний рядок	
Write Виведення (запис) даних на монітор	Write(a ₁ , a ₂ ,..., a _n), де a ₁ , a ₂ , ...,a _n – константи або змінні зазначених типів	Виведення зазначених даних (чисел, змінних, тексту, який обмежено апострофами) виконується з позиції курсору	Write ('Відповідь',2) Відповідь 2 Write('Привіт,' , Name,!') Привіт, Федоре!
Writeln		Після виведення значень a ₁ , a ₂ , ...,a _n курсор переводиться на наступний рядок	

Завдання з теми 2.

Тема 6. Логічні вирази. Умовний оператор. Оператор вибору

Теоретична частина

У загальному випадку повна форма конструкції умовного оператора має вигляд:

якщо <логічний вираз> тоді	if <логічний вираз> then
<оператор 1>	<оператор 1>
інакше	else
<оператор 2>	<оператор 2>

Неповна форма умовного оператора має наступний вигляд.

якщо <логічний вираз> тоді	if <логічний вираз> then
<оператор>	<оператор 1>

Іноді може знадобитися виконання послідовності операторів. У цьому випадку ця послідовність вкладається в блок **begin end**;

Перед службовим словом else роздільник (;) не ставиться.

Логічні операції

not	заперечення
and	і
or	або

Завдання 1. Вивести на екран більше з двох даних чисел.

```
var x, y: integer;
begin
  WriteLn ('Введіть 2 числа'); {вводимо два цілих числа через пробіл}
  ReadLn (x, y);
  If x > y then
    WriteLn (x) {якщо x більше y, то виводимо x}
  else
    WriteLn (y); {інакше виводимо y}
  ReadLn;
end.
```

Завдання 2. Скласти програму, яка, якщо введене число негативне, міняє його на протилежне.

```
var x: real;
begin
  WriteLn ('Введіть число'); {вводимо ціле число}
  ReadLn (x);
  if x < 0 then
    x := - x;
  WriteLn (x);
  ReadLn;
end.
```

Завдання з теми 3.

**Цикли з передумовою та післяумовою.
Вкладені цикли. Опрацювання виключень
Теоретична частина**

Конструкція циклу *for*:

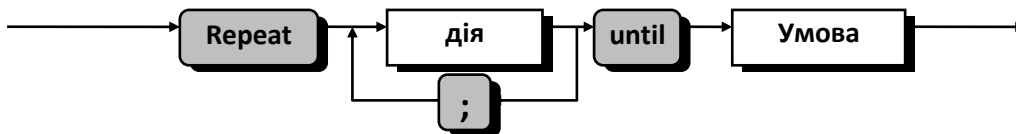
for змінна:=початок **to** кінець **do** дія (якщо початок < кінець)

for змінна:=початок **downto** кінець **do** дія (якщо початок > кінець)

Конструкція циклу *передумовою*:



Конструкція циклу *післяумовою*:



Завдання 1. Дано n цілих чисел. Обчислити їх суму.

```
var x, n, i, S: integer;
begin
  writeln ('Введіть n');
  readln (n);
  for i:=1 to n do
  begin
    readln(x);
    S:=S+x;
  end;
  writeln (s) {виводимо суму}
  readLn;
end.
```

```
var x, n, i, S: integer;
begin
  writeln ('Введіть n');
  readln (n);
  i:=1;
  while (i<=n) do
  begin
    readln(x);
    S:=S+x;
    i:=i+1;
  end;
  writeln (s)
  readln;
end.
```

```
var x, n, i, S: integer;
begin
  writeln ('Введіть n');
  readln (n);
  i:=1;
  repeat
    readln(x);
    S:=S+x;
    i:=i+1;
  until (i>n);
  writeln (s)
  readln;
end.
```

Завдання з теми 4.

Тема 10. Одновимірні масиви**Теоретична частина**

Одновимірні масиви описуються наступним чином:

```
var    mas = array [1..10] of integer;
```

Введення і виведення масивів

```
writeln('Введіть елементи масиву');
for i:=1 to n do
  readln(mas[i]);
```

```
writeln('Масив mas:');
for i:=1 to n do
  writeln (mas[i]);
```

Завдання. До кожного елемента масиву додати число 7.

```
const k=7; n=10;
var    mas = array [1..n] of integer;
      i:integer;
begin
  writeln('Введіть елементи масиву');
  for i:=1 to n do
    readln(mas[i]);

  for i:=1 to n do
    mas[i] := mas[i]+k;

  writeln('Масив mas:');
  for i:=1 to n do
    writeln (mas[i]);

end.
```

Завдання

1. Дано масив розміру N. Вивести його елементи в зворотному порядку.
2. Дано масив розміру N. Вивести спочатку його елементи з парними (непарними) номерами, а потім - з непарними (парними).
3. Дано цілочисельний масив A розміру N. Вивести номери тих його елементів A [i], які задовольняють подвійну нерівність: $A[1] < A[i] < A[10]$.
4. Дано цілочисельний масив розміру N. Перетворити його, додавши до парних (непарних) елементів перший (останній).
5. Поміняти місцями мінімальний і максимальний елементи масиву.
6. Дано масив розміру N. Здійснити циклічний зсув елементів масиву вліво (вправо) на одну позицію.

7. Дано масив розміру N і число k ($0 < k < 5$, $k < N$). Здійснити циклічний зсув елементів масиву вліво (вправо) на k позицій.
8. Перевірити, чи утворюють елементи цілочисельного масиву розміру N арифметичну (геометричну) прогресію.
9. Дано масив розміру N . Визначити кількість ділянок, на яких його елементи монотонно зростають (спадають).
10. Упорядкувати масив розміру N за зростанням (спаданням).

Тема 11. Двовимірні масиви

Теоретична частина

Одновимірні масиви описуються наступним чином:

```
var mas = array [1..10, 1..5] of integer;
```

Введення і виведення двовимірних масивів

```
writeln('Введіть елементи масиву');
for i:=1 to n do
  for j:=1 to m do
    readln(mas[i,j]);
```

```
writeln('Масив mas:');
for i:=1 to n do
  begin
    for j:=1 to m do
      write(mas[i,j], ' ');
    writeln;
```

Завдання. Для кожного рядка двовимірної масиву обчислити суму його елементів.

```
const n=10; m=5
var mas = array [1..n, 1..m] of integer;
    i, j, s: integer;
begin
  writeln('Введіть елементи масиву');
  for i:=1 to n do
    for j:=1 to m do
      readln(mas[i,j]);
  for i:=1 to n do
    begin
      s:=0;
      for j:=1 to m do
        s:=s+mas[i,j];
      writeln ('Сума елементів ', i, '-го рядка = ', s);
    end
end.
```


Завдання

1. Дано число k ($0 < k < N$) і матриця розміру $N \times M$. Знайти суму і добуток елементів k -го рядка даної матриці.
2. Дано матрицю розміру $N \times M$. Знайти суми елементів всіх її парних (непарних) рядків (стовпців).
3. Дано матрицю розміру $N \times M$. Знайти мінімальне (максимальне) значення в кожному рядку (стовпці) .
4. Дано матрицю розміру $N \times M$. Перетворити матрицю, помінявши місцями мінімальний і максимальний елемент в кожній стрічці (стовпці).
5. Дано матрицю розміру $N \times M$. Знайти мінімальний (максимальний) серед максимальних (мінімальних) елементів кожної стрічки (стовпця).
6. Дано квадратну матрицю порядку N . Знайти суму елементів її головної (побічної) діагоналі.
7. Дано квадратну матрицю порядку M . Знайти суми елементів її діагоналей, паралельних головній (побічній).
8. Дано квадратну матрицю порядку M . Дзеркально відобразити її елементи відносно горизонтальної (вертикальної) осі симетрії матриці.
9. Дано квадратну матрицю порядку M . Дзеркально відобразити її елементи відносно побічної діагоналі матриці.
10. Дано два числа k_1 і k_2 і матриця розміру $N \times M$. Поміняти місцями стрічки (стовпці) матриці з номерами k_1 і k_2 .

Тема 12. Процедури та функції**Теоретична частина****Оголошення функції:**

Function <ім'я> [(формальні параметри)]:<тип результату>;

[<розділи описів>;]

begin

<розділ операторів>

end;

Формат виклику:

Y:=<ідентифікатор функції > [(фактичні параметри)];

Оголошення процедури:

Procedure <ім'я> [(формальні параметри)];

[<розділи описів>;]

Begin

<розділи операторів>

end;

Формат виклику:

<ідентифікатор> [(фактичні параметри)];

Завдання

1. Описати функцію $\text{Min2}(A, B)$ ($\text{Max2}(A, B)$) дійсного типу, яка знаходить мінімальне (максимальне) з двох дійсних чисел A і B . За допомогою цієї функції знайти мінімальне (максимальне) з пар чисел A і B , A і C , A і D , якщо дано числа A, B, C, D .
2. Описати процедуру $\text{Minmax}(A, B)$, що записує в змінну A мінімальне зі значень A і B , а в змінну B - максимальне з цих значень (A і B - дійсні параметри, що є одночасно вхідними та вихідними). Знайти мінімальне і максимальне з чисел A, B, C, D .
3. Описати функцію $\text{Fact}(N)$ цілого типу, яка обчислює значення факторіала $N! = 1 \cdot 2 \cdot \dots \cdot N$ ($N > 0$ - параметр цілого типу). Обчислити факторіали 10 заданих чисел.
4. Описати нерекурсивну функцію $\text{Fib}(N)$ цілого типу, яка обчислює N -е число Фібоначчі $F(N)$ за формулою: $F(1) = F(2) = 1$, $F(k) = F(k-2) + F(k-1)$, $k = 3, 4, \dots$. Обчислити 10 чисел Фібоначчі із зазначеними номерами.
5. Описати функцію $\text{Exp1}(x, \text{eps})$ дійсного типу (параметри x, eps - дійсні, $\text{eps} > 0$), знаходить наближене значення функції $\exp(x)$: $\exp(x) = 1 + x + x^2 / 2! + x^3 / 3! + \dots + x^n / n! + \dots$. У сумі враховувати всі доданки, більші за eps . Знайти наближене значення експоненти для даного x і eps .
6. Описати функцію $\text{Ln1}(x, \text{eps})$ дійсного типу (параметри x, eps - дійсні, $x < 1$, $\text{eps} > 0$), яка знаходить наближене значення функції $\ln(1+x)$:

$$\ln(1+x) = x - x^2 / 2 + x^3 / 3 - \dots + (-1)^n x^n / n + \dots$$
У сумі враховувати всі доданки, більші за модулем за eps . Знайти наближене значення $\ln(1+x)$ для даного x та eps .
7. Описати процедуру $\text{Invert}(A, N)$, яка міняє порядок проходження елементів масиву A з N дійсних чисел на протилежний ("інвертування масиву"). Масив A - вхідний і вихідний параметр, N - вхідний параметр. Інвертувати масиви A, B, C розміру N_A, N_B, N_C відповідно.
8. Описати процедуру $\text{Smooth}(A, N)$, яка замінює кожен елемент масиву A розміру N на його середнє арифметичне зі своїми сусідами ("згладжування масиву"). Масив A - вхідний і вихідний параметр, N - вхідний параметр. Виконати п'ятикратне згладжування даного масиву A розміру N , виводячи на екран результати кожного згладжування.
9. Описати рекурсивні функції $\text{Fact}(N)$ дійсного типу, що обчислюють значення факторіала $N!$ ($N > 0$ - параметр цілого типу). Обчислити факторіали п'яти заданих чисел.
10. Описати рекурсивну функцію $\text{PowerN}(x, n)$ дійсного типу, яка знаходить значення n -го степеня числа x за формулою: $x^0 = 1$, $x^n = x \cdot x^{n-1}$ при $n > 0$, $x^n = 1 / x^{-n}$ при $n < 0$ ($x \neq 0$ - дійсне число, n - ціле). Обчислити значення заданого полінома.

Тема 13. Робота зі символами та стрічками. Робота з файлами

Теоретична частина

Функції і процедури для роботи зі стрічками

1. **Функція Concat.** Функція Concat здійснює об'єднання рядків R1; R2, R3 в один рядок в такому порядку, в якому вони записані.

Формат *Concat(R1,R2,R3)*; або *R1+R2+R3 (string)*

Const R1='Мова '; R2='програмування '; R3='Turbo Pascal';

Var R:string[35];

begin

R:=concat(R1,R2,R3);

writeln(R);

end.

2. **Функція Length.** Функція Length видає фактичну довжину рядка, який міститься в даній змінній. При підрахуванні довжини рядка враховуються всі символи, в тому числі і проміжки!!!

Формат: *Length(R)*; (*integer*)

Program Flength;

Const R='Turbo Pascal';

Var N: integer;

begin

N:=length(R);

writeln('n=',N);

end.

3. **Функція Copy.** Функція Copy копіює фрагмент довжиною N рядка R, починаючи з позиції Poz.

Формат *Copy(R, Poz, N)*; (*string*)

Program Fcopy;

const R='Turbo Pascal';

var R1: string[6];

begin

R1:=Copy(R, 7, 6);

writeln(R1);

end.

4. **Функція Pos.** Функція Pos знаходить номер позиції P, з якої починається перше входження слова C в рядку R. Якщо слово C в рядку R не знайдено, то буде надруковано число 0.

Формат *Pos(C,R)*; (*integer*)

```

Program Fpos;
Const R= 'інформатика'; C= 'форма';
Var P: integer;
begin
P:=Pos(C, R);
writeln('P=',P);
end.

```

5. Процедура Insert. Процедура Insert вставляє слово R1 в рядок R, починаючи з позиції Poz.

Формат *Insert(R1, R, Poz); (string)*

```

Program PInsert;
Var Poz: integer;
R, R1: string[35];
begin
R:='Мова Pascal';
R1:='Turbo ';
insert(R1, R, 6);
writeln(R);
end.

```

6. Процедура Delete. Процедура Delete знищує слово, яке починається з вказаної позиції Poz і має задану довжину N в рядку R.

Формат *Delete(R, Poz, N); (string)*

```

Program PDelete;
Var R:string[35];
begin
R:='Мова Turbo Pascal';
delete(R, 6, 6);
writeln(R);
end.

```

Практична частина

1. Дано стрічку. Вивести перші k символів стрічки (k менше, ніж довжина стрічки).
2. Дано стрічку. Вивести коди її першого і останнього символів.
3. Дано стрічку. Вивести стрічку, що містить ті ж символи, але розташовані у зворотному порядку.
4. Дано стрічку, що зображає ціле число. Вивести суму цифр цього числа.

5. Дано стрічку. Порахувати кількість входжень слова 'літо' в дану стрічку і замінити його на слово 'зима'.
6. Дано стрічку. Надрукувати її, видаливши із неї зайві пропуски.
7. Дано два числа: N1 і N2, і дві стрічки: S1 і S2. Отримати з цих стрічок нову, об'єднавши N1 перших символів стрічки S1 і N2 останніх символів стрічки S2.
8. Дано стрічки: S1 і S2. Перевірити, чи міститься стрічка S2 в стрічці S1. Якщо так, то вивести номер позиції, починаючи з якої S2 міститься в S1.
9. Дано дві стрічки: S1 і S2. Визначити кількість входжень стрічки S2 в стрічку S1.
10. Дано стрічку S і символ C. Подвоїти кожне входження символу C в стрічку S.
11. Дано речення англійською мовою і число k ($0 < k < 10$). Зашифрувати рядок, виконавши циклічну заміну кожної букви на букву того ж регістра, розташовану в алфавіті на k-й позиції після літери, що шифрується. Знаки пунктуації та пробіли не змінювати.

Тема 14. Робота з файлами. Модулі та їх застосування

Теоретична частина

Оголошення файлових змінних

```
var f1, f2: text;
```

Процедури та функції для роботи з файлами:

assign(f, 'ім'я_файлу') – встановлює зв'язок між файловою змінною і файлом;

reset(f) – відкрити файл для читання;

rewrite(f) – створити або перезаписати файл;

append(f) – створити або відкрити і дописати в кінець файлу;

close(f) – зберегти і закрити файл

readln(f, a, b); — читати з файлу f дві змінні a і b;

writeln(f, a, b); — записати у файл f змінні a і b, а потім записати туди ж символ "кінець рядка".

Завдання

1. Дано два файли цілих чисел одного розміру з іменами Name1 і Name2. Створити текстовий файл з ім'ям NameT, що містить зображення цих чисел, розташовані у два стовпці: перший містить числа з файлу Name1, другий - з файлу Name2.
2. Дано дійсні числа A, B і ціле число N. Створити текстовий файл, що містить таблицю значень функції $f(x) = \sin(x)$ на проміжку [A, B] з кроком $(B-A) / N$. Таблиця складається з двох стовпців: з аргументами x і зі значеннями f(x).

Список літератури**Базова**

1. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль. – М.: Наука, 1988.
2. Вирт Н. Алгоритмы + структуры данных = программы. – М.: Мир, 1985.
3. Вьюнова Н.И., Галатенко В.А., Ходулев А.Б. Систематический подход к программированию. – М.: Наука, 1988.
4. Дудзяний І. М. Програмування мовою Object Pascal. – Львів: ЛНУ імені Івана Франка, 2004
5. Злобін Г.Г. Посібник користувача ПК. Навч. посібник – Львів, СПОЛОМ, 2003 – 272 с.
6. Злобін Г.Г., Рикалюк Р.Є. Архітектура і апаратне забезпечення IBM-подібних ПЕОМ. Навч. посібник - К.: Каравела, 2005 – 304 с.
7. Ковалюк Т.В. Основи програмування. – К.: Видавнича група ВНУ, 2005. – 384с.

Допоміжна

1. Довгаль С.И., Литвинов Б.Ю., Сбитнев А.И. Персональные ЭВМ: Турбо Паскаль V6.0. Объектное программирование. Локальные сети. – К.: Информсистема сервис, 1993.
2. Зуев Е.А. Язык программирования Turbo Pascal 6.0. – М.: Унитех, 1992.
3. Костів О. В., Ярошко С. А. Методи розробки алгоритмів: Тексти лекцій. – Львів: ЛНУ імені Івана Франка, 2002.
4. Корнієнко М. М. Інформатика. Основи алгоритмізації і програмування: Теоретичні основи, приклади та завдання, практичні роботи / М. М. Корнієнко, І. Д. Іванова—, 2009.— 48 с.

Задачники

1. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. – М.: Наука, 1988.
2. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989.