

СП КП 2025-2026 н.р. // варіанти завдань для КІ-307

(мова не може містити жодних інших синтаксичних конструкцій, окрім тих, що вказані в таблиці 1 та таблиці 2)

Таблиця 1

Обов'язкові синтаксичні елементи мови
(також вказано номер варіанту для виконання лабораторних робіт)

ПІБ студента	Номер варіанту (для лабораторних робіт)	Тіло програми	Оператори вводу-виводу	Оператор присвоєння	Оператори	Регістр ключових слів	Формат ідентифікаторів (заданий регулярним виразом)	Операції			Оголошені типи даних	Коментар
								арифметичні	порівняння	логічні		
Альошин Олександр	1	program <name>; data...; begin - end	input output	<rvalue> :> <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) >: goto (як у мові C) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	_[A-Z][0-9][A-Z]	+, -, *, /, %	eg; ne; lt; gt	!!; && or	int та масив таких елементів	??...
Бигар Андрій	2	PROGRAM <name>; DATA...; BEGIN - END	SCAN PRINT	<rvalue> :> <lvalue> (присвоєння зліва направо)	IF [-ELSE] (як у мові Rust) FOR-DOWNT0-DO (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	_[A-Z][A-Z][0-9]	ADD; SUB; MUL; DIV; MOD	EQ; NE; > <	NOT; AND;	LONG INT та масив таких елементів	?? ... ??
Боднар Мар'ян	3	Program <name>; Data...; Begin - End	Get Put	<rvalue> :> <lvalue> (присвоєння зліва направо)	If [-Else] (як у мові Rust) For-To-Do (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	_[A-Z][0-9][0-9]	+, -, Mul; Div; Mod	Eg; Ne; >> <<	!; And;	Int16 та масив таких елементів	!! ...!!
Васьків Дмитро	4	program <name>; begin data...; end	read write	<rvalue> :> <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) while (як у мові Rust) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	[A-Z][A-Z][0-9]	add; sub; *; /; %	==; !=; lt; gt	!; &; or	int32 та масив таких елементів	/*...
Вінчур Максим	5	PROGRAM <name>; BEGIN DATA...; END	INPUT OUTPUT	<rvalue> :> <lvalue> (присвоєння зліва направо)	IF [-ELSE] (як у мові Rust) REPEAT-UNTIL (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	[A-Z][A-Z][0-9][0-9][0-9]	+, -, *; DIV; MOD	=; <> LT; GT	!!; && OR	INT 2 та масив таких елементів	*# ... *#

Гога Анастасія	6	Program <name>; Start Data ...; Finish	Scan Print	<rvalue> > <lvalue> (присвоєння зліва направо)	If [-Else] (як у мові Rust) Goto (як у мові C) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	_[a-z][a-z][a-z][0-9]	++; --; **; Div; Mod	Eg; Ne; Le; Ge	Not; And;	Int_4 та масив таких елементів	%* ... **%
Даниляк Мирослав	7	program <name>; data ...;start - finish	get put	<rvalue> > <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) for-downto-do (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	_[a-z][a-z][0-9]	++; --; *; /; %	eg; ne; >; <=	!; and;	int16_t та масив таких елементів	??...
Довгунь Павло	8	PROGRAM <name>; DATA ...;START - FINISH	READ WRITE	<rvalue> > <lvalue> (присвоєння зліва направо)	IF [-ELSE] (як у мові Rust) FOR-TO-DO (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	_[a-z][0-9][0-9]	ADD; SUB; MUL; DIV; MOD	=; <; LE; GE	!; AND;	INT32_t та масив таких елементів	%% ... %%%
Досужий Петро	9	Program <name>; Data ...;Start - Finish	Input Output	<rvalue> > <lvalue> (присвоєння зліва направо)	If [-Else] (як у мові Rust) While (як у мові Rust) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	[a-z][a-z][a-z][a-z][0-9]	+; -; Mul; Div; Mod	==; !=; Le; Ge	!!; And;	Integer та масив таких елементів	\$\$...
Дубовіч Андрій-Богдан	10	program <name>; start data ...; finish	scan print	<rvalue> > <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) repeat-until (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	[a-z][a-z][a-z][0-9]	add; sub; *; /; %	=; <; >; <	not; and;	integer16 та масив таких елементів	/ * ... */
Клуб Віктор	11	# PROGRAM <name>; START DATA ...; STOP	GET PUT	<rvalue> > <lvalue> (присвоєння зліва направо)	IF [-ELSE] (як у мові Rust) GOTO (як у мові C) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	[a-z][a-z][0-9]	+; -; *; DIV; MOD	=; <; >; <=	!; AND;	INTEGER32 та масив таких елементів	!! ...
Клуб Олег	12	#Program <name>; Data...; Start – Stop	Read Write	<rvalue> > <lvalue> (присвоєння зліва направо)	If [-Else] (як у мові Rust) For-Downto-Do (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	_[0-9][A-Z][A-Z][0-9]	++; --; **; Div; Mod	==; !=; >; <	!; And; Or	Integer_2 та масив таких елементів	// ...
Микитич Михайло	13	#program <name>; start data...; stop	input output	<rvalue> > <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) for-to-do (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	_[0-9][A-Z][0-9]	+; -; *; /; %	==; !=; >; <=	!!; &&; or	integer_4 та масив таких елементів	// ... //
Михальченко Вероніка	14	# PROGRAM <name>; DATA...; START - STOP	SCAN PRINT	<rvalue> > <lvalue> (присвоєння зліва направо)	IF [-ELSE] (як у мові Rust) WHILE (як у мові Rust) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	_[A-Z][0-9][0-9][0-9]	ADD; SUB; MUL; DIV; MOD	==; !=; >; <	NOT; AND;	INTEGER16_t та масив таких елементів	## ... ##
Мітяєв Олександр	15	#Program <name>; Start Variable...; End	Get Put	<rvalue> > <lvalue> (присвоєння зліва направо)	If [-Else] (як у мові Rust) Repeat-Until (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	[A-Z][0-9][A-Z][0-9]	+; -; Mul; Div; Mod	=; <; >; <	!; And;	Integer32_t та масив таких елементів	\\ ...

Николишин Остап	16	startprogram startblock data...; endblock	read write	<rvalue> > <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) goto (як у мові C) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	[A-Z][A-Z][0-9][0-9]	add; sub; *; /; %	==; <; >; <<	!; and; or	longint та масив таких елементів	@ @ ...
Падура Максим	17	STARTPROGRAM STARTBLOCK DATA...; ENDBLOCK	INPUT OUTPUT	<rvalue> > <lvalue> (присвоєння зліва направо)	IF [-ELSE] (як у мові Rust) FOR-DOWNT0-DO (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	_[a-z][0-9][a-z][0-9]	++; --; *; DIV; MOD	=, !=, <, >	!!; &; OR	INT16 та масив таких елементів	## ...
Підлужний Петро	18	Startprogram Startblock Data...; Endblock	Scan Print	<rvalue> > <lvalue> (присвоєння зліва направо)	If [-Else] (як у мові Rust) For-To-Do (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	_[a-z][a-z][0-9][0-9]	++; --; **; Div; Mod	Eq; Neq; Ls; Gr	Not; And;	Int32 та масив таких елементів	## ... ##
Пікалюк Сергій	19	startprogram data...; startblock - endblock	get put	<rvalue> > <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) while (як у мові Rust) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	_[a-z][0-9][0-9]	++; --; *; /; %	eq; noteq; less; gr	!; and;	int_2 та масив таких елементів	!! ...!!
Пільганський Владислав	20	STARTPROGRAM DATA...; STARTBLOCK - ENDBLOCK	READ WRITE	<rvalue> > <lvalue> (присвоєння зліва направо)	IF [-ELSE] (як у мові Rust) REPEAT-UNTIL (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	[a-z][a-z][a-z][0-9][0-9]	ADD; SUB; MUL; DIV; MOD	EQ; NE; LT; GT	NOT; &; OR	INT_4 та масив таких елементів	/*...
Романів Назарій	21	Mainprogram Var...; Start - End	Input Output	<rvalue> > <lvalue> (присвоєння зліва направо)	If [-Else] (як у мові Rust) Goto (як у мові C) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	[a-z][a-z][0-9][0-9]	++; --; Mul; Div; Mod	Eg; Ne; >; <	!!; And;	Int16_t та масив таких елементів	### ... ###
Савчак Марко	22	mainprogram var...; start - end	scan print	<rvalue> > <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) for-downto-do (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	[a-z][0-9][0-9]	add; sub; *; /; %	eg; ne; >; <	not; and;	int32_t та масив таких елементів	##* ... *##
Салаш Юрій	23	PROGRAM <name>; DATA...; BEGIN - END	SCAN PRINT	<rvalue> > <lvalue> (присвоєння зліва направо)	IF [-ELSE] (як у мові Rust) FOR-DOWNT0-DO (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	_[a-z][A-Z][A-Z][0-9]	++; SUB; MUL; DIV; %	EQ; NE; >; <	NOT; AND;	LONG INT та масив таких елементів	?? ... ??
Сенюк Дмитро	24	Program <name>; Data...; Begin - End	Get Put	<rvalue> > <lvalue> (присвоєння зліва направо)	If [-Else] (як у мові Rust) For-To-Do (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	_[a-z][A-Z][0-9]	++; --; *; Div; Mod	Eg; Ne; >; <	!; And;	Int16 та масив таких елементів	!! ...!!
Стафесв Андрій	25	program <name>; start var...; end	read write	<rvalue> > <lvalue> (присвоєння зліва направо)	if [-else] (як у мові Rust) while (як у мові Rust) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	[A-Z][A-Z][0-9]	add; --; mul; /; %	==; !=; lt; gt	not; and; or	int32 та масив таких елементів	/*...

Стефанишин Любомир	26	PROGRAM <name>; BEGIN DATA...; END	INPUT OUTPUT (присвоєння зліва направо)	<rvalue> > <lvalue>	IF [-ELSE] (як у мові Rust) REPEAT-UNTIL (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	[a-z][A-Z][A-Z][0-9]	++; SUB; MUL; DIV; %	=; <; LT; GT	!!; &&; OR	INT_2 та масив таких елементів	## ... ##
Чапля Роман	27	Program <name>; Start Var ...; End	Scan Print (присвоєння зліва направо)	<rvalue> > <lvalue>	If [-Else] (як у мові Rust) Goto (як у мові C) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	_[A-Z][a-z][0-9]	++; Sub; **; Div; Mod	Eg; Ne; Le; Ge	Not; And;	Int_4 та масив таких елементів	%* ... **%
Янишин Олег	28	program <name>; var ...;start - end	get put (присвоєння зліва направо)	<rvalue> > <lvalue>	if [-else] (як у мові Rust) for-downto-do (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Low	_[A-Z][a-z][a-z][0-9]	++; --; *; /; %	eg; ne; >; <=	!; and;	int16_t та масив таких елементів	??...
	29	PROGRAM <name>; VAR ...;BEGIN - END	READ WRITE (присвоєння зліва направо)	<rvalue> > <lvalue>	IF [-ELSE] (як у мові Rust) FOR-TO-DO (як у мові Паскаль) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up	_[a-z][A-Z][a-z][0-9]	++; --; MUL; DIV; MOD	=; <; LE; GE	!; &; OR	INT32_t та масив таких елементів	%% ... %%%
	30	Program <name>; Var ...;Start - End	Input Output (присвоєння зліва направо)	<rvalue> > <lvalue>	If [-Else] (як у мові Rust) While (як у мові Rust) оператор індексації: [] (як у мові C) порожній оператор: ; (як у мові C) початковий символ блоку: { (як у мові C) кінцевий символ блоку: } (як у мові C)	Up-Low перший символ Up	[a-z][a-z][0-9]	Add; -; Mul; Div; Mod	==; !=; Le; Ge	!!; And;	Integer та масив таких елементів	\$\$...

Таблиця 2
Додаткові синтаксичні елементи мови (можна не реалізовувати)

Тіло підпрограми	Оператори повернення з підпрограми	Погодження про виклики підпрограм	Оператор виклику підпрограми (повертає значення)
subprogram <name>; arguments...; var...; begin - end	return <expression>	cdecl (C calling convention)	run (<name>, this, ...)
* реєстр ключових слів має відповідати варіанту вказаному в таблиці 1			