

СП КП 2025-2026 н.р. // варіанти завдань для КІ-309

(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 | | |
| Брожек Ярина | 1 | program <name>; var...; begin - end | input output | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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][A-Z] | +, -, *, /, % | eg; ne; lt; gt | !!; &&; | int та масив таких елементів | ??... |
| Верещак Мар'ян | 2 | PROGRAM <name>; VAR...; BEGIN - END | SCAN PRINT | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | IF [-ELSE] (як у мові Rust) FOR-DOWNTWO-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] | ADD; SUB; MUL; DIV; MOD | EQ; NE; >; < | NOT; AND; OR | LONG INT та масив таких елементів | ?? ... ?? |
| Витрикуш Андріана | 3 | Program <name>; Var...; Begin - End | Get Put | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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] | +, -, Mul; Div; Mod | Eg; Ne; >>; << | !; And; Or | Int16 та масив таких елементів | !! ...!! |
| Ганущак Мар'яна | 4 | program <name>; begin var...; end | read write | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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][A-Z] | add; sub; *; /; % | ==; !=; lt; gt | !; &; | int32 та масив таких елементів | /*... |
| Груник Владислав | 5 | PROGRAM <name>; BEGIN VAR...; END | INPUT OUTPUT | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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] | +, -, *; DIV; MOD | =; <>; LT; GT | !!; &&; | INT_2 та масив таких елементів | *# ... *# |

| | | | | | | | | | | | | |
|-----------------------|----|--|-----------------|---|--|----------------------------|----------------------|-------------------------|-----------------|--------------|---|------------|
| Дідошак Олександр | 6 | Program <name>; Start Var ...; 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] | ++; --; **; Div; Mod | Eg; Ne; Le; Ge | Not; And; Or | Int_4 та масив таких елементів | %* ... **% |
| Думанський Дмитро | 7 | program <name>; var ...;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] | ++; --; *; /; % | eg; ne; >; <= | !; and; or | int16_t та масив таких елементів | ??... |
| Ільєв Ілля | 8 | PROGRAM <name>; VAR ...;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] | ADD; SUB; MUL; DIV; MOD | =; <; LE; GE | !; &; | INT32_t та масив таких елементів | %% ... %%% |
| Корчинський Владислав | 9 | Program <name>; Var ...;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] | +; -; Mul; Div; Mod | ==; !=; Le; Ge | !!; &&; | Integer та масив таких елементів | \$\$... |
| Кузнєцов Григорій | 10 | program <name>; start var ...; 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] | add; sub; *; /; % | =; <; >; < | not; and; or | integer16 та масив таких елементів | / * ... */ |
| Левчук Ілля | 11 | # PROGRAM <name>; START VARIABLE ...; 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] | +; -; *; DIV; MOD | =; <; >; <= | !; AND; OR | INTEGER32 та масив таких елементів | !! ... |
| Луценко Андрій | 12 | #Program <name>; Variable...; 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] | ++; --; **; Div; Mod | ==; !=; >; < | !; &; | Integer_2 та масив таких елементів | // ... |
| Мерунович Захар | 13 | #program <name>; start variable...; 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] | +; -; *; /; % | ==; !=; >; <= | !!; &&; | integer_4 та масив таких елементів | // ... // |
| Мисишин Назар | 14 | # PROGRAM <name>; VARIABLE...; 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] | ADD; SUB; MUL; DIV; MOD | ==; !=; !>; !< | NOT; AND; OR | INTEGER16_t та масив таких елементів | ## ... ## |
| Плюкар Володимир | 15 | #Program <name>; Start Variable...; Stop | 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] | +; -; Mul; Div; Mod | =; <; >; !>; !< | !; And; Or | Integer32_t та масив таких елементів | \\ ... |

| | | | | | | | | | | | | |
|--------------------|----|--|-----------------|---|--|----------------------------|----------------------|-------------------------|---------------------|--------------|--------------------------------------|-------------|
| Польчин Марія | 16 | startprogram startblock endblock variable...; | 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] | add; sub; *; /; % | ==; <; >; << | !; &; | longint та масив таких елементів | @ @ ... |
| Поп Михайло | 17 | STARTPROGRAM STARTBLOCK VARIABLE...; 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] | ++; -; *; DIV; MOD | =, !=, <<, >> | !!; &&; | INT16 та масив таких елементів | ## ... |
| Посацька Анастасія | 18 | Startprogram Startblock Endblock Variable...; | 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] | ++; --; **; Div; Mod | Eq; Neq; Ls; Gr | Not; And; Or | Int32 та масив таких елементів | ## ... ## |
| Ребенчак Святослав | 19 | startprogram variable...; 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] | ++; -; *; /; % | eq; noteq; less; gr | !; and; or | int_2 та масив таких елементів | !! ...!! |
| Рихва Віталій | 20 | STARTPROGRAM VARIABLE...; 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] | ADD; SUB; MUL; DIV; MOD | EQ; NE; LT; GT | !; &; | INT_4 та масив таких елементів | / * ... |
| Сидор Софія | 21 | Mainprogram Data...; 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] | ++; -; Mul; Div; Mod | Eg; Ne; >; < | !!; &&; | Int16_t та масив таких елементів | ### ... ### |
| Содома Вероніка | 22 | mainprogram data...; 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] | add; sub; *; /; % | eg; ne; >; << | not; and; or | int32_t та масив таких елементів | ##* ... *## |
| Спанчак Руслан | 23 | PROGRAM <name>; VAR...; 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] | ++; SUB; MUL; DIV; % | EQ; NE; >; < | NOT; AND; OR | LONG INT та масив таких елементів | ?? ... ?? |
| Стецишин Максим | 24 | Program <name>; Var...; 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] | ++; -; *; Div; Mod | Eg; Ne; >; << | !; And; Or | Int16 та масив таких елементів | !! ...!! |
| Сусяк Павло | 25 | program <name>; begin 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] | add; -; mul; /; % | ==; !=; lt; gt | !; &; | int32 та масив таких елементів | / * ... |

| | | | | | | | | | | | | |
|-----------------|----|---|-----------------|--|--|----------------------------|------------------|-----------------------|----------------|--------------|-------------------------------------|------------|
| Турак Микола | 26 | PROGRAM <name>; BEGIN VAR...; END | INPUT OUTPUT | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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] | +, SUB; MUL; DIV; % | =; <; LT; GT | !!; &&; | INT_2 та масив таких елементів | ## ... ## |
| Фещак Вікторія | 27 | Program <name>; Start Var ...; Finish | Scan Print | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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] | ++; Sub; **; Div; Mod | Eg; Ne; Le; Ge | Not; And; Or | Int_4 та масив таких елементів | %* ... **% |
| Шевченко Богдан | 28 | program <name>; var ...;start - finish | get put | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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] | ++; --; *, /; % | eg; ne; >; <= | !; and; or | int16_t та масив таких елементів | ??... |
| | 29 | PROGRAM <name>; VAR ...;START - FINISH | READ WRITE | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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] | ++; --; MUL; DIV; MOD | =; <; LE; GE | !; &; | INT32_t та масив таких елементів | %% ... %%% |
| | 30 | Program <name>; Var ...;Start - Finish | Input Output | <rvalue> > <lvalue> <i>(присвоєння зліва направо)</i> | 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] | Add; -, Mul; Div; Mod | ==; !=; Le; Ge | !!; &&; | Integer та масив таких елементів | \$\$... |

Таблиця 2
Додаткові синтаксичні елементи мови (можна не реалізовувати)

| Тіло підпрограми | Оператор повернення з підпрограми | Угода 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 | | | |