

## Тема №1 2.

### РОБОТА З ФАЙЛАМИ

Файл – це тип даних, який використовується для довготривалого збереження інформації і має доступ до “вторинної” пам’яті, тобто магнітного диску. Кожен файл має ім’я, яке зареєстроване у відповідному каталозі на диску. Файл можна вважати структурованим типом даних, який складається з компонентів одного і того ж типу, однакової довжини і структури.

Існує три види файлів і, відповідно, три способи оголошення файлу:

```
F1=file of <тип даних>;  
F2=file;  
F3=text;
```

Де  $F1$  – типізований,  $F2$  – нетипізований,  $F3$  – текстовий файли. При оголошенні типізованого файлу вказується тип даних компонентів, який може бути як простим, так і структурованим.

Для отримання доступу до оголошеного файлу перше, що потрібно зробити, це пов’язати (асоціювати) файлову змінну з реальним файлом на диску. Для цього використовується стандартна процедура *Assign*. Наприклад:

```
Assign(f3, 'd:\prg\examp.txt');
```

Після асоціювання файлу його потрібно відкрити, до того ж відкрити його можна для зчитування, запису та зчитування і запису одночасно.

Процедура *Reset(f1)*, застосована до текстового файлу, відкриває його для зчитування, до типізованого або нетипізованого – для зчитування та запису.

Процедура *Rewrite(f1)*, застосована до текстового файлу, відкриває його для запису, до типізованого або нетипізованого – для зчитування та запису.

Різниця між цими двома процедурами в застосуванні до типізованих та нетипізованих файлів полягає в тому, що *Reset* видасть помилку під час звернення до неіснуючого файлу, а *Rewrite* завжди буде знищувати вміст файлу і замінювати його на новий.

Процедура *Append(f)* використовується для того, щоб відкрити текстовий файл для дозапису. Тобто старі дані не затираються, а нові дописуються в кінці файлу.

Для зчитування та запису даних в файлі використовують стандартні процедури *Read*, *Write* – для текстових та типізованих файлів, *BlockRead*, *BlockWrite* – для нетипізованих файлів. Першим параметром в цих процедурах вказується файлова змінна. Процедури *ReadLn* та *WriteLn* застосовуються лише до текстових файлів.

Функція *IOResult* повертає код помилки, що виникла під час звернення до файлу або при виконанні останньої операції вводу, виводу. Якщо помилок не було, функція повертає нуль. Для того, щоб контролювати помилки вводу/виводу за допомогою *IOResult*, потрібно виключити автоматичний контроль таких помилок за допомогою директиви компілятора *{SI-}* (додаток 5).

Часто при роботі з файлами застосовують функцію *EOF(f)*, яка повертає значення *TRUE* при досягненні покажчиком (поточним положенням в файлі) кінця файлу. У всіх інших випадках – *FALSE*.

Після роботи з файлом його необхідно закрити, що здійснює процедура *Close(f)*.

## **Завдання 1**

Скласти блок-схему алгоритму та написати програму, яка записує у файл символи псевдографіки (символи рамок та символи заповнення).

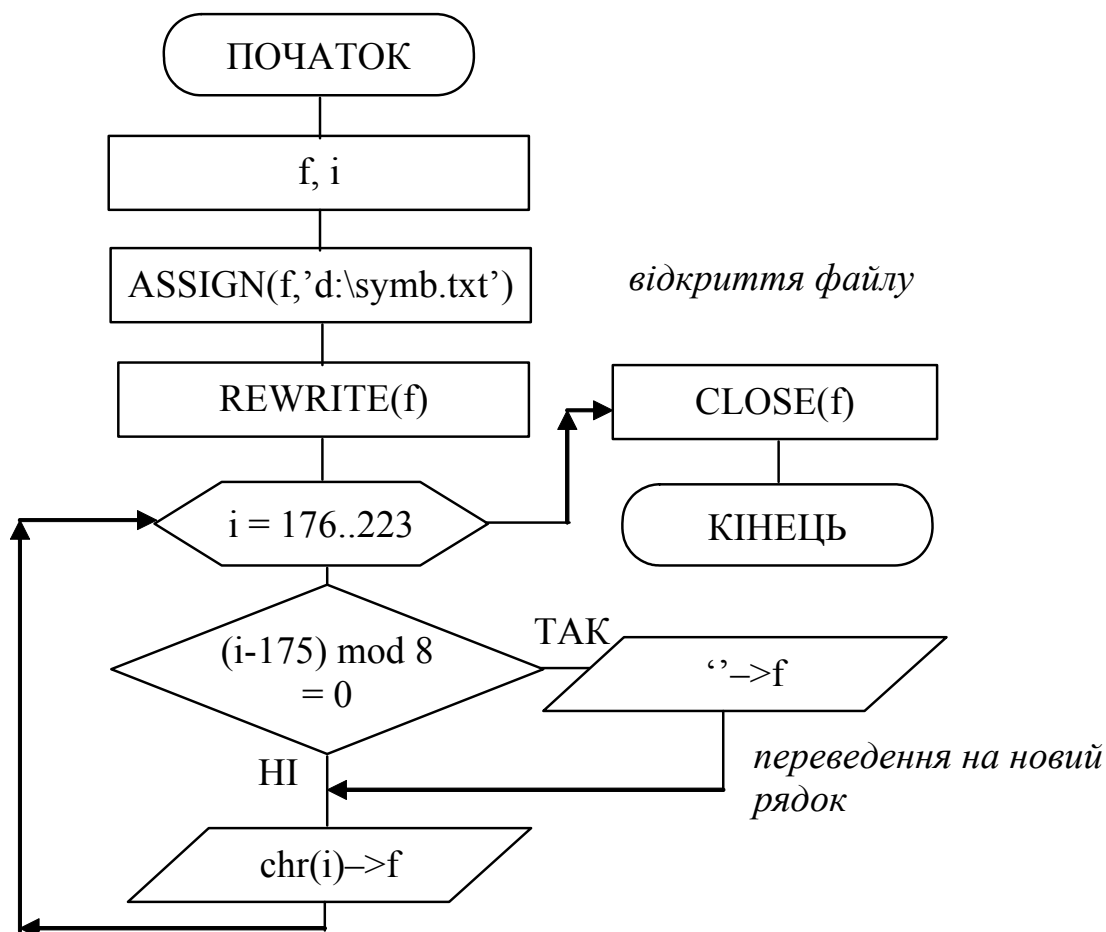
**Методичні рекомендації**

1. Відповідно до поставленого завдання проаналізувати задачу.

Кожен символ псевдографіки має своє кодове значення (число), до того ж всі символи рамок та заповнення мають коди, які відрізняються на одиницю (додаток 8). Для їх виводу доцільно використати цикл *FOR*, оскільки відомо від якого та по який код символу потрібно виводити. А сам процес виведення символів направити у текстовий файл.

2. Скласти блок-схему задачі.

Блок-схема буде мати такий вигляд:



3. Скласти програму з наведеної блок-схеми.  
Код програми буде мати такий вигляд:

```

{*****}
* SymbFile - програма запису символів                *
*                псевдографіки у текстовий файл      *
* Copyright (c) Шищук В.В. гр.ІС-04-1, 09/12/04    *
*****}
PROGRAM SymbFile;
VAR
    f: text;
    i: integer;
BEGIN
    writeln('ЗАПИС СИМВОЛІВ ПСЕВДОГРАФІКИ У ФАЙЛ');
    assign(f, 'd:\symb.txt');
    rewrite(f);
    writeln(f, 'Символи рамок та заповнення:');
    for i:=176 to 223
        begin
            if ((i-175) mod 8 = 0) then writeln(f, '');
            write(f, chr(i):5);
        end;
    close(f);
END.

```

4. Зберегти написану програму в персональну папку.

---

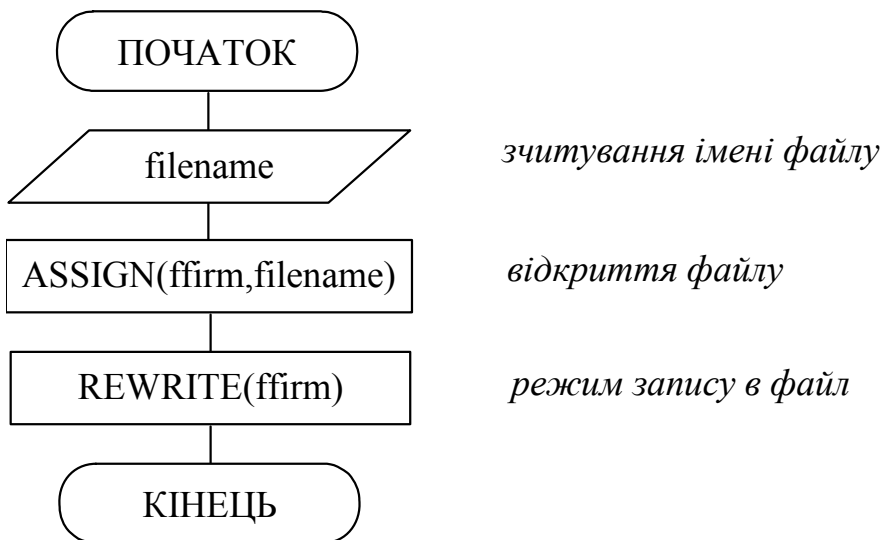
## **Завдання 2**

Скласти дві програми для задачі попередньої теми №11 (масив записів про фірми), одна з яких виконувала б зчитування з клавіатури даних та їх запис до файлу, інша – зчитувала б з цього файлу дані та проводила їх обробку (перелік відмінників).

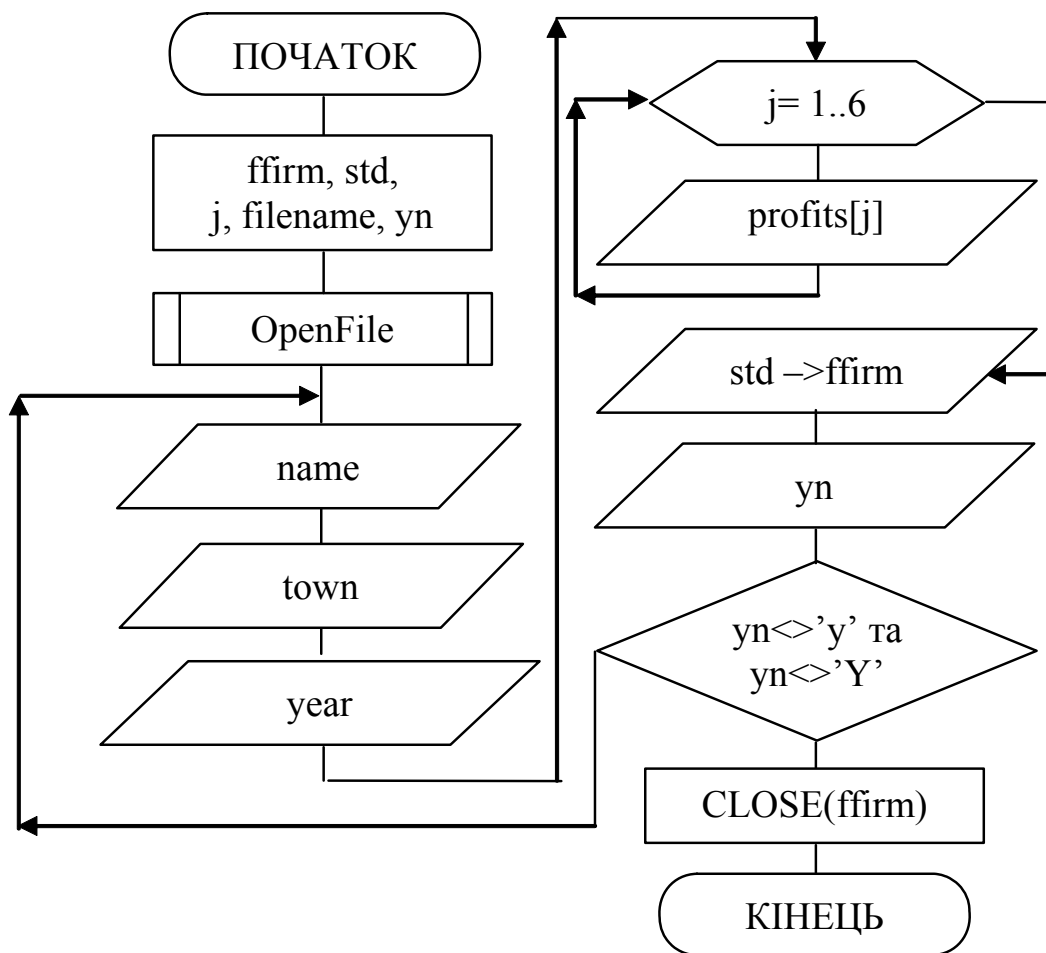
### **Методичні рекомендації**

1. Проаналізувати технічну постановку задачі.  
Оскільки дані, якими оперує програма, мають тип запис, потрібно оголосити файл типу запис та змінну того ж типу для збереження поточних даних (буферна змінна).
2. Скласти блок-схему програми запису даних у файл.  
Для відкриття файлу для запису скористаємося окремою процедурою, блок-схема якої буде мати такий вигляд:

процедура FileOpen відкриття файлу для запису



Блок-схему програми запису даних у файл з використанням даної процедури можна записати у такому вигляді:



### 3. Записати код програми запису у файл за складеною блок-схемою.

```

{*****}
* WriteFile - програма запису введених даних      *
*           у типізований файл                  *
* Copyright (c) Шишук В.В. гр.ІС-04-1, 09/12/04  *
*****}
PROGRAM WriteFile;
TYPE
  TdataFirm = RECORD
    name      : string[50];
    town      : string[30];
    year      : integer;
    profits   : array[1..6] of real;
  END;
VAR
  ffirm : file of TdataFirm;
  std   : TdataFirm;
  filename: string[12];
  j: integer;
  yn:char;
BEGIN

PROCEDURE OpenFile;
BEGIN
  write('Введіть назву файлу(та повний шлях): ');
  readln(filename);
  assign(ffirm,filename); /* підключення файлу */
  rewrite(ffirm);        /* відкриття для запису */
END;

BEGIN
  OpenFile;
  with std do
    repeat
      write('Вкажіть назву фірми: ');
      readln(name);
      write('Вкажіть місто розташування фірми: ');
      readln(name);
      write('Вкажіть рік заснування фірми: ');
      readln(year);
      for j:=1 to 6 do

```

```

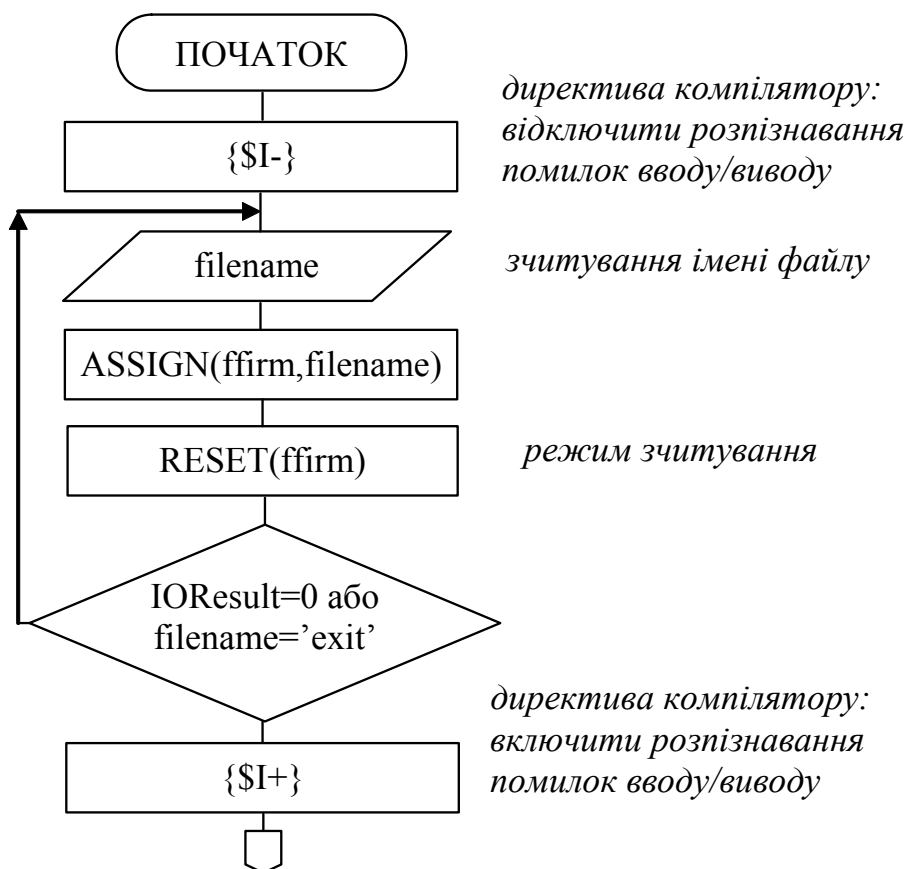
begin
    write(j, '-ий місяць: ');
    readln(profits[j]);
end;
write(ffirm, std);    /* запис до файлу */
write('Будемо записувати ще одного(y/n)? ');
readln(yn);
until (yn<>'y') and (yn<>'Y');
close(ffirm);        /* закриття файлу*/
END.

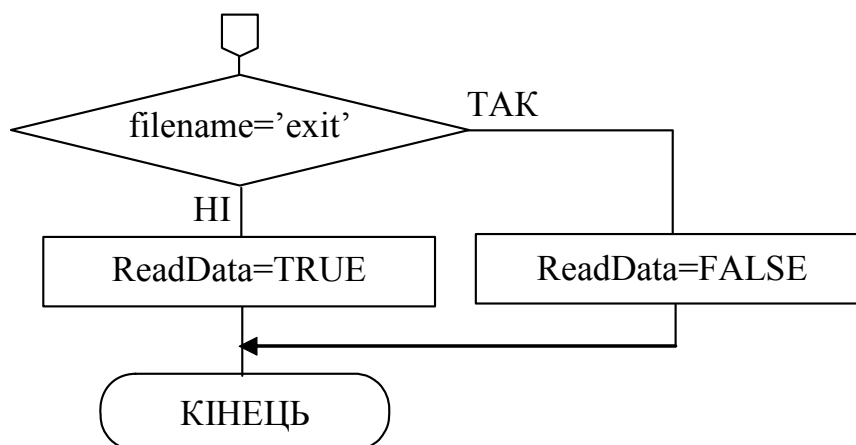
```

4. Відкомпілювати програму та перевірити її роботу.
5. Зберегти програму у персональну папку.
6. Скласти блок-схему для зчитування даних з файлу.

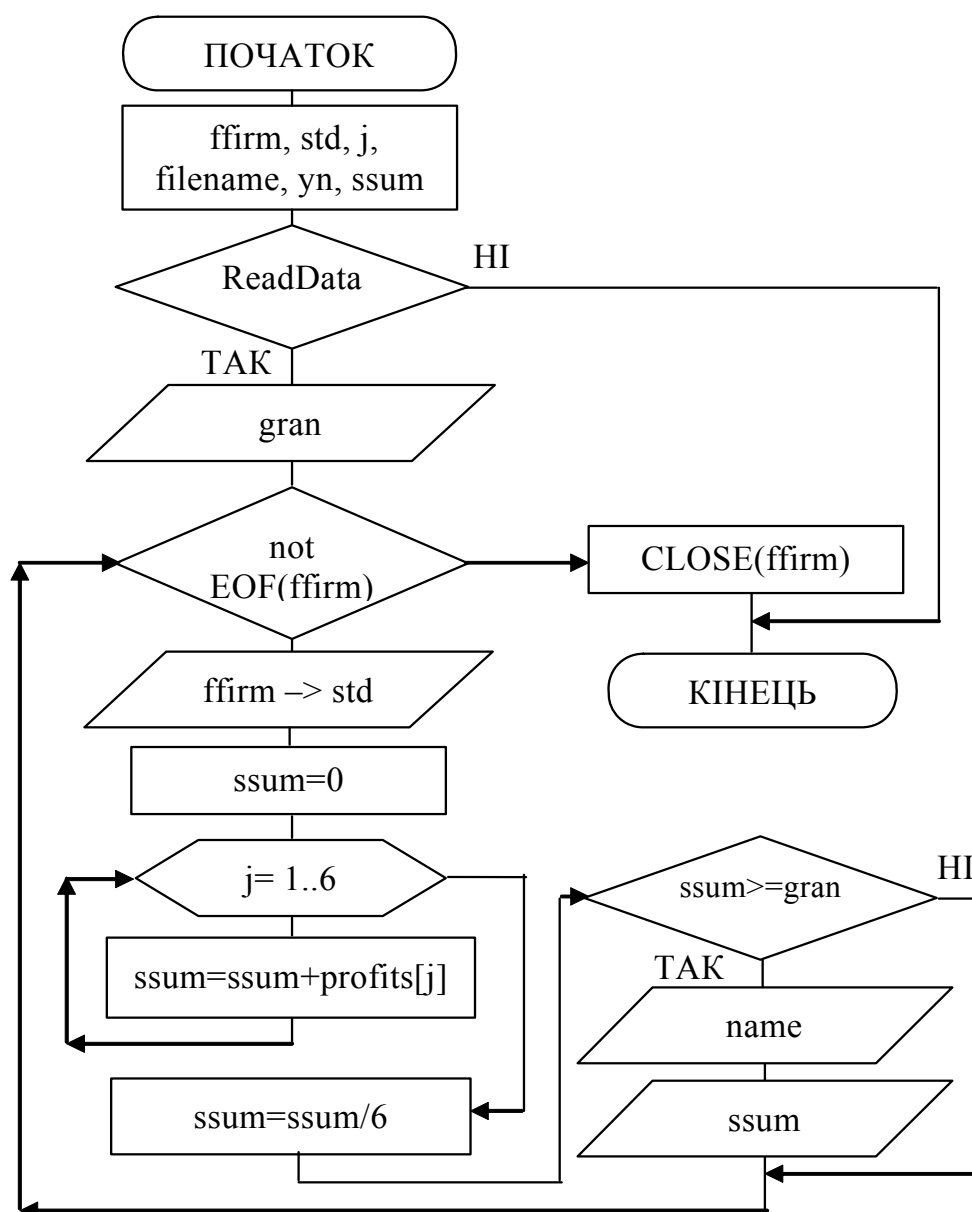
Відкриття файлу для зчитування даних запишемо в окремій функції, в якій передбачимо перевірку коректності вводу імені файлу. Блок-схема такої функції буде мати такий вигляд:

*функція ReadData відкриття файлу для зчитування*





Блок-схема програми зчитування даних з використанням описаної функції буде мати вигляд:





## 7. Записати код програми запису у файл за складеною блок-схемою.

```
{*****}
* ReadFile - програма зчитування даних з файлу      *
*           та знаходження студентів-відмінників  *
* Copyright (c) Шищук В.В. гр.ІС-04-1, 04/12/04    *
{*****}
PROGRAM ReadFile;
TYPE
  TdataFirm = RECORD
    name      : string[50];
    town      : string[30];
    year      : integer;
    profits   : array[1..6] of real;
  END;
VAR
  fstud: file of TdataFirm;
  std   : TdataFirm;
  filename: string[12];
  j: integer;
  yn:char;
  ssum, gran: real;

FUNCTION ReadData:Boolean;
BEGIN
  {$I-}          /* директива компілятора */
  repeat
    write('Введіть ім'я файлу або "exit" для виходу:');
    readln(filename);
    assign(ffirm,filename);
    reset(ffirm);
    until (IOResult=0) or (filename='exit');
  {$I+}          /* директива компілятора */
  if filename = 'exit' then ReadData:=false
  else ReadData:=true; /*повернення значення*/
END;

BEGIN
  if ReadData then /* чи вказаний файл */
  begin
    write('Введіть граничну суму: ');
    readln(gran);
```

```
while NOT EOF(ffirm) do
  begin
    read(ffirm, std);
    with std do
      begin
        ssum := 0;
        for j:=1 to 6 do
          ssum := ssum+profits[j];
        ssum:=ssum/6;
        if ssum >= gran then
          writeln(name:25, ssum:10:4);
        end;
      end;
    close(ffirm);
    write('Натисніть Enter для виходу:');
    readln
  end
END.
```

8. Відкомпілювати програму та перевірити її роботу.
9. Зберегти програму у персональну папку.
10. Скласти звіт про виконану роботу (дві програми).

### **Індивідуальні завдання:**

Скласти дві програми: для обробки типізованого файлу та для обробки текстового файлу. Передбачити перевірку на коректність роботи з файлом. За основу завдання обробки типізованого файлу взяти завдання з попередньої теми №11.

**рівень** 

#### *а) Програма обробки типізованого файлу:*

Скласти дві програми, одна з яких виконує зчитування даних з клавіатури та їх запис до файлу, інша – зчитує з цього файлу дані та проводить необхідну обробку (див. індивідуальні завдання теми №11).

*б) Програма обробки текстового файлу:*

**Варіант 1.**

Визначити кількість слів та букв в заданому текстовому файлі.

**Варіант 2.**

Визначити кількість голосних та приголосних букв в заданому текстовому файлі.

**Варіант 3.**

Визначити скільки разів входить зазначене слово в заданий текстовий файл.

**Варіант 4.**

Визначити найдовше та найкоротше слово в заданому текстовому файлі.

**Варіант 5.**

Визначити найдовше речення в заданому текстовому файлі та вказати його порядковий номер.

**Варіант 6.**

Визначити кількість слів в заданому текстовому файлі, які містять м'який знак.

**Варіант 7.**

Вилучити з заданого текстового файлу всі слова довжиною в три символи.

**Варіант 8.**

Визначити кількість розділових знаків в заданому текстовому файлі.

**Варіант 9.**

Підрахувати кількість цифр, які зустрічаються в заданому текстовому файлі.

**Варіант 10.**

Визначити кількість слів, які містять букви W,V,X в заданому текстовому файлі.

**Варіант 11.**

Визначити найкоротше речення в заданому текстовому файлі та вказати його порядковий номер.

**Варіант 12.**

Визначити середню кількість букв в рядках заданого текстового файлу.

**Варіант 13.**

Підрахувати кількість прогалін (пробілів) в заданому текстовому файлі. Визначити скільки подвійних, потрійних і т.д. прогалін.

**Варіант 14.**

Вилучити з заданого текстового файлу всі розділові знаки.

**Варіант 15.**

Вміст заданого текстового файлу без розділових знаків переписати в інший текстовий файл.

рівень **Б**

*а) Програма обробки типізованого файлу*

Скласти одну програму, яка б виконувала всі можливі дії над записами: перегляд існуючих даних, зміна даних по ключу, додавання нових записів, вилучення існуючих записів та обробка даних у відповідності з варіантом. Вибір дії представити у вигляді номера операції, для чого створити відповідний опис (див. індивідуальні завдання теми №11).

*б) Програма обробки текстового файлу:*

**Варіант 1.**

Створити текстовий файл на базі двох заданих текстових файлів, вміст якого є чергування слів в заданих файлах.

### **Варіант 2.**

Створити текстовий файл, який містить слова, що зустрічаються як в одному, так і в другому із заданих файлів.

### **Варіант 3.**

Користуючись заданим текстовим файлом, в якому в три стовпчики записані дані: назва товару (одне слово), кількість товару (шт.), гуртова та роздрібна ціна (в гривнях). За умови, що всі товари, записані в файлі були продані, визначити виручку підприємця.

### **Варіант 4.**

Вивести всі букви, які містяться в заданому текстовому файлі з зазначенням кількості повторень в тексті.

### **Варіант 5.**

Підрахувати кількість процедур та функцій користувача, які використовуються в програмі, код якої розміщений в заданому текстовому файлі.

### **Варіант 6.**

Знайти цикли та їх кількість, що використовуються в коді програми на мові Pascal, яка зберігається в заданому текстовому файлі.

### **Варіант 7.**

Визначити кількість входжень цілих чисел з діапазону, який зазначається, в заданому текстовому файлі.

### **Варіант 8.**

Підрахувати середнє значення всіх чисел, які зустрічаються в заданому текстовому файлі, вказати найбільше та найменше з них.

### **Варіант 9.**

Визначити, з якими файлами працює Pascal-програма, код якої записаний в заданому текстовому файлі.

### **Варіант 10.**

Підрахувати загальну суму в гривнях та доларах, виходячи з заданого текстового файлу. Всі суми повинні записуватись цифрами, після яких слідує “грн.” або перед якими стоїть знак “\$”.

### **Варіант 11.**

В одному заданому файлі міститься відомість нарахувань (прізвище-сума). В другому файлі міститься відомість утримань (прізвище-сума). Послідовність прізвищ в обох файлах не співпадає. Створити зведену відомість (прізвище-сума нараховано-сума утримано).

### **Варіант 12.**

В заданому файлі міститься надходження товару (назва товару – кількість), його ціна знаходиться в іншому файлі-довіднику (назва товару – ціна). Назва товару в першому файлі може повторюватися за різною кількістю. Дописати в перший файл в кожен рядок загальну суму кожного приходу.

### **Варіант 13.**

Поєднати два заданих текстових файли, по черзі вибрати з них речення, починаючи з кінця.

### **Варіант 14.**

Надрукувати всі слова-паліндроми (в прямому та зворотному напрямку читаються однаково), що містяться в заданому текстовому файлі.

### **Варіант 15.**

В заданому текстовому файлі міститься перелік літератури (бібліографічний опис). Визначити середню кількість сторінок всіх книжок та рік видання, в який було надруковано найбільше книжок.

# рівень **B**

## *а) Програма обробки типізованого файлу*

Скласти одну програму, яка б виконувала всі можливі дії з записами: перегляд існуючих даних, зміна даних по ключу, додавання нових записів, вилучення існуючих записів та обробка даних у відповідності з варіантом. Вибір дії представити у вигляді номеру операції, для чого створити відповідний опис (див. індивідуальні завдання теми №11).

## *б) Програма обробки текстового файлу:*

### **Варіант 1.**

Перевірити заданий файл, що містить програму на мові Pascal, на коректність використання складового оператора (begin..end).

### **Варіант 2.**

Перерахувати в алфавітному порядку всі слова, які містяться в заданому текстовому файлі.

### **Варіант 3.**

В двох заданих текстових файлах записані тексти процедур на Pascal-і (по одній). В третьому файлі знаходиться код програми. Вставити в потрібне місце коду програми тексти процедур.

### **Варіант 4.**

Підрахувати кількість глобальних змінних та загальну кількість локальних змінних, оголошених в заданому текстовому файлі, який містить код програми на Pascal.

### **Варіант 5.**

Визначити чи коректно проставлені символи “крапка з комою” в кодї Pascal-програми, яка розміщена в заданому текстовому файлі.

**Варіант 6.**

Визначити за кодом Pascal-програми, який записаний в заданому текстовому файлі, кількість умовних операторів і чи є серед них вкладені умовні оператори.

**Варіант 7.**

Визначити за кодом Pascal-програми, який записаний в заданому текстовому файлі, кількість викликів функції користувача, яка описана в цьому ж файлі.

**Варіант 8.**

По заданому текстовому файлу, в якому розміщений код програми на Pascal, створити стільки нових файлів, скільки циклів записано в заданому коді, до кожного з цих файлів записати відповідні цикли.

**Варіант 9.**

В заданому текстовому файлі знайти слово, яке містить максимальну кількість заданих букв (кількість заздалегідь невідома).

**Варіант 10.**

Згідно заданому текстовому файлу, в якому розміщений код програми на Pascal, створити два нових файли, в один записати всі оператори вводу інформації, які зустрічаються в заданому коді, а в другий – всі оператори виводу інформації.

**Варіант 11.**

Визначити в коді Pascal-програми, що записаний в заданому текстовому файлі, які стандартні підпрограми обробки символьних рядків використовуються в програмі і скільки разів.

**Варіант 12.**

Визначити яка кількість слів в заданому текстовому файлі утворюється з заданого слова шляхом перестановки його букв.



### **Варіант 13.**

Визначити помилки, які зустрічаються в заданому текстовому файлі. Помилками вважати повторення слова більше одного разу в реченні. Кінцем речення вважати крапку, знак оклику чи знак запитання. Створити інший файл, в який записати виправлений текст.

### **Варіант 14.**

По заданому текстовому файлу, в якому розміщений код програми на Pascal, створити стільки нових файлів, скільки підпрограм записано в заданому коді, в кожен з цих файлів записати відповідні підпрограми, а з заданого файлу вилучити тексти всіх підпрограм.

### **Варіант 15.**

Вибрати з заданого текстового файлу лише такі слова, в яких є символи, що зустрічаються в файлі один, два та три рази.

### **Контрольні запитання:**

- 1) Що таке файл, файлові типи та файлові змінні?
- 2) Наведіть приклади оголошення файлів різних типів.
- 3) Опишіть процедуру відкриття та організації доступу до файлу.
- 4) Назвіть всі відомі Вам процедури та функції роботи з файлами.
- 5) Як організовується перевірка коректності звернення до файлу?
- 6) Для чого використовують директиви препроцесора при роботі з файлами?
- 7) Якими засобами можна переглянути вміст файлу, котрий був створений засобами мови Pascal?
- 8) Що таке файл довільного доступу?
- 9) Як виконати дозапис даних в існуючий файл?
- 10) Що таке покажчик файлу?