Тема №9 .

МЕТОДИ СОРТУВАННЯ ТА ПОШУКУ. ВИКОРИСТАННЯ СИМВОЛІВ ПСЕВДОГРАФІКИ

При роботі з масивами чи іншими наборами однотипних даних часто виникає потреба сортування елементів, тобто розміщення їх за зростанням значень. У програмуванні застосовують основні методи сортування.

Сортування методом бульбашок

Полягає в перестановці сусідніх елементів. Масив переглядається від першого елемента до останнього, якщо два сусідніх елементи розташовані за зростанням, вони залишаються без зміни, якщо ж ні – переставляються місцями. В результаті одного проходження максимальний елемент буде розташований на останньому місці. Наступне проходження буде виконуватися знову від першого, але до передостаннього елемента і т.д. З кожним проходженням кількість елементів, які порівнюють, буде скорочуватись на один. В результаті отримаємо відсортовані елементи.

Сортування методом вибору

Масив переглядається перший раз – знаходиться мінімальний його елемент. Знайдений мінімальний елемент міняється місцями з першим елементом. Другий раз масив переглядається починаючи з другого елемента. Знову знаходиться мінімальний елемент і міняється місцями з другим елементом і т.д. Процес продовжується до тих пір, поки не поміняються місцями (або залишаться) передостанній елемент з останнім.

Сортування методом вставок

Масив переглядається, починаючи з першого елемента. При знаходженні наступного елемента, який менший за будь-який попередній, він ставиться перед ним. Тобто від-

бувається вставка елемента в потрібну позицію серед елементів, відсортованих на попередньому етапі.

Сортування методом Шелла

Цей метод найбільш ефективний. Спочатку порівнюються віддалені елементи, а потім ближчі. На першому етапі встановлюють інтервал, який рівний N/2, де N – розмірність масиву. Отже, порівнюється перший елемент і елемент в позиції 1+N/2, потім 2-й і 2+N/2 і т.ін. При необхідності елементи, що порівнюються, міняються місцями. На другому проходженні встановлюється інтервал N/4 і знову проводять порівняння відповідних елементів. І так доти, поки не будуть порівнюватися сусідні елементи (як в методі бульбашок), після чого масив буде відсортованим.

Під час роботи з масивами досить часто виникає потреба пошуку необхідного значення. При реалізації цього можливі лише два варіанти алгоритмів, залежно від того, відсортовані дані чи ні в шуканому наборі.

Метод послідовного пошуку

Використовується для пошуку даних в невпорядкованому масиві. Починаючи з першого елемента, порівнюється шукане значення з кожним елементом по черзі.

Пошук припиняється коли необхідні дані знайдено або досягнуто кінець масиву.

Метод бінарного (двійкового) пошуку

Використовується у випадках, коли елементи масиву відсортовані. Для цього застосовується метод половинного ділення. Спочатку порівнюють шукане значення з середнім елементом масиву. Якщо він більший за шуканий, другу частину масиву відкидають, якщо він менший за шуканий, відкидають першу половину масиву. Частину, яка залишилася, ділять пополам і знову порівнюють шукане значення з середнім елементом.

Пошук припиняється у випадку, коли при порівнянні на будь-якому етапі знаходиться шукане значення або поки не залишиться наступного елемента.

Завдання

Скласти блок-схему алгоритму та написати програму сортування методом бульбашок елементів одновимірного масиву цілих чисел розмірністю в 10 елементів. Заповнення масиву здійснити випадковим чином (від –50 до 50). На кожному етапі перестановки елементів виводити масив на екран у вигляді таблиці.

Методичні рекомендації

1. Згідно індивідуального завдання скласти загальний план сортування елементів масиву за заданим методом.

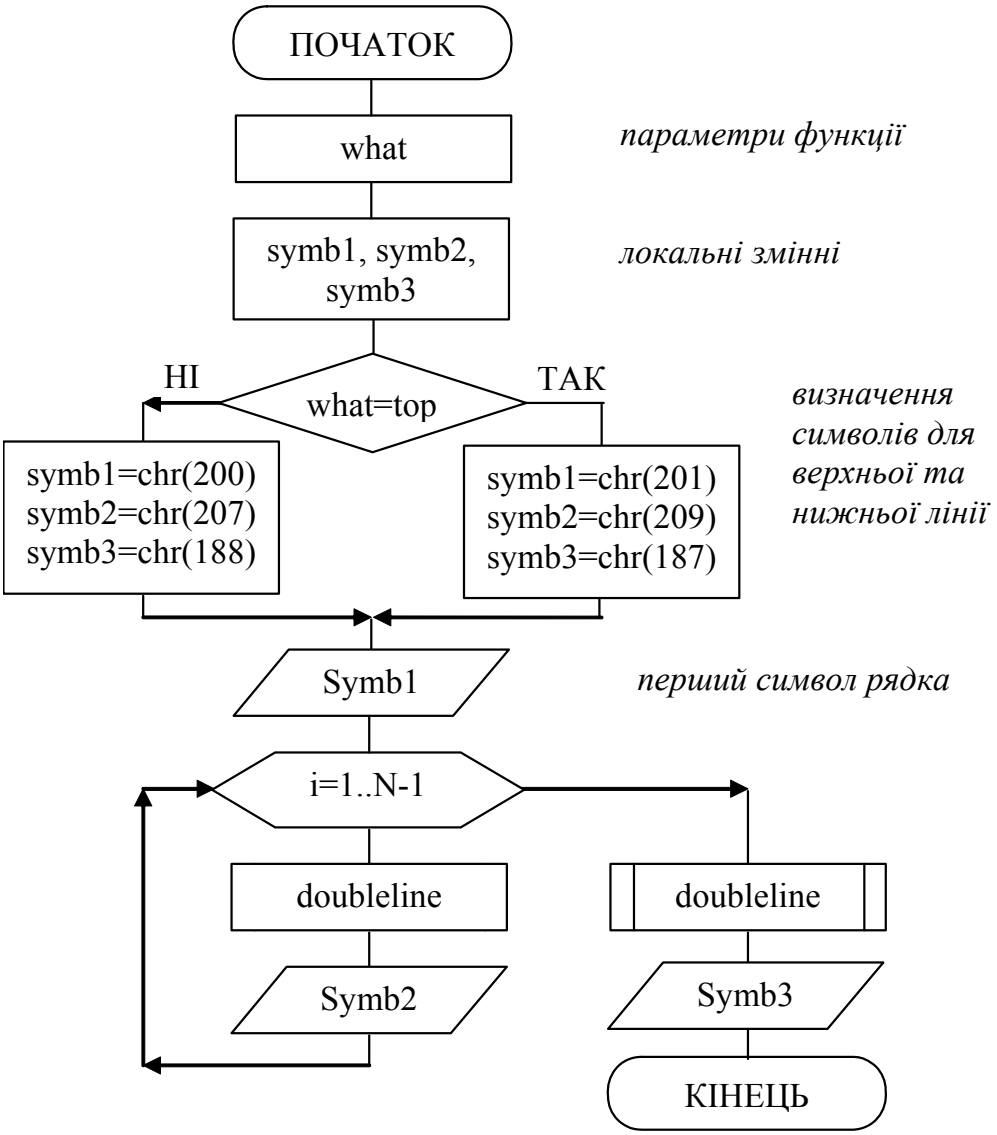
   Спочатку необхідно заповнити масив випадковими значеннями, потім методом бульбашок виконати поетапну перестановку елементів.

   Для виводу масиву на екран таблиці створимо окремі процедури. Наприклад, відобразимо масив на кожному етапі сортування у вигляді рядка такої таблиці:

5	-10	4	0	…	-21	1
-10	4	0	5	…	1	42
…	…	…	…	…	…	…
-36	-21	0	4	…	18	42
-36	-21	0	4	…	37	42

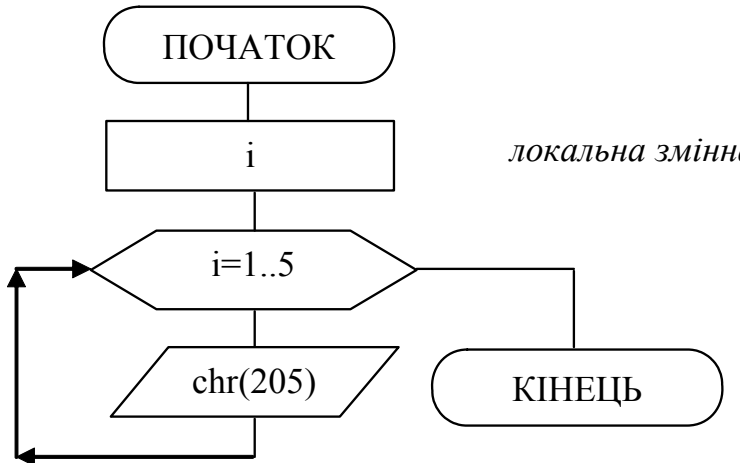
   Для побудови такої таблиці потрібно вивести подвійні лінії зверху та знизу таблиці, для чого створимо процедуру з одним параметром, який буде визначатиме, які символи псевдографіки використовувати на перетині ліній таблиці та в її кутках. На кожен елемент таблиці відведемо по 4 символи, після яких добавимо пробіл. Блок-схема такої процедури буде мати такий вигляд:

процедура outline(what) – відображення верхньої та нижньої лінії таблиці

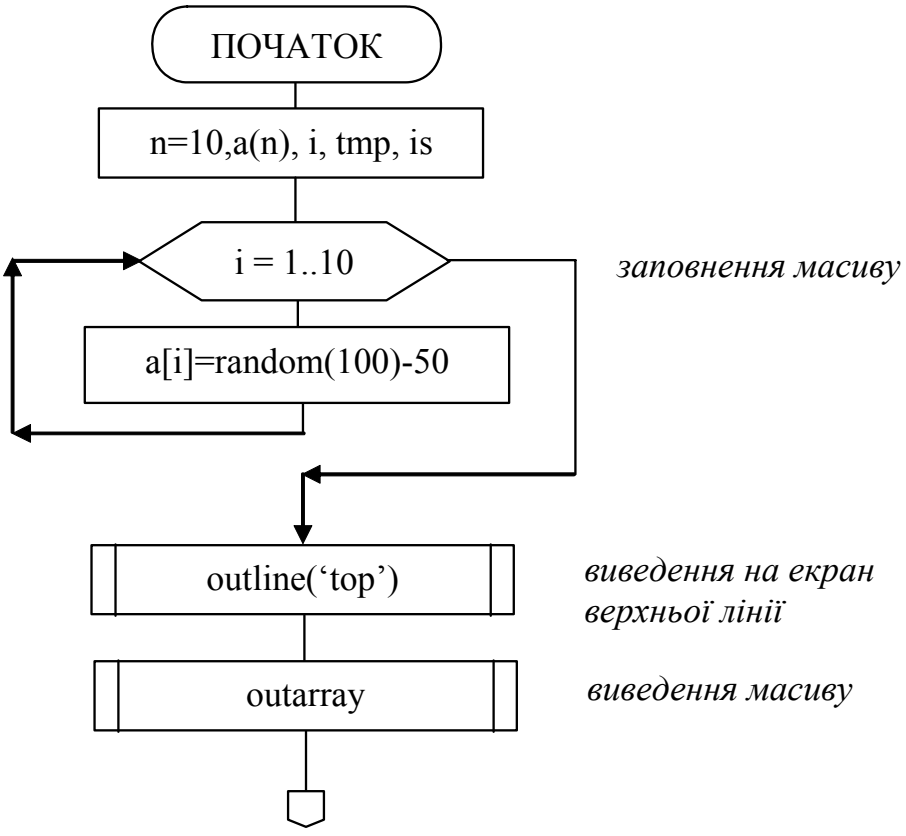


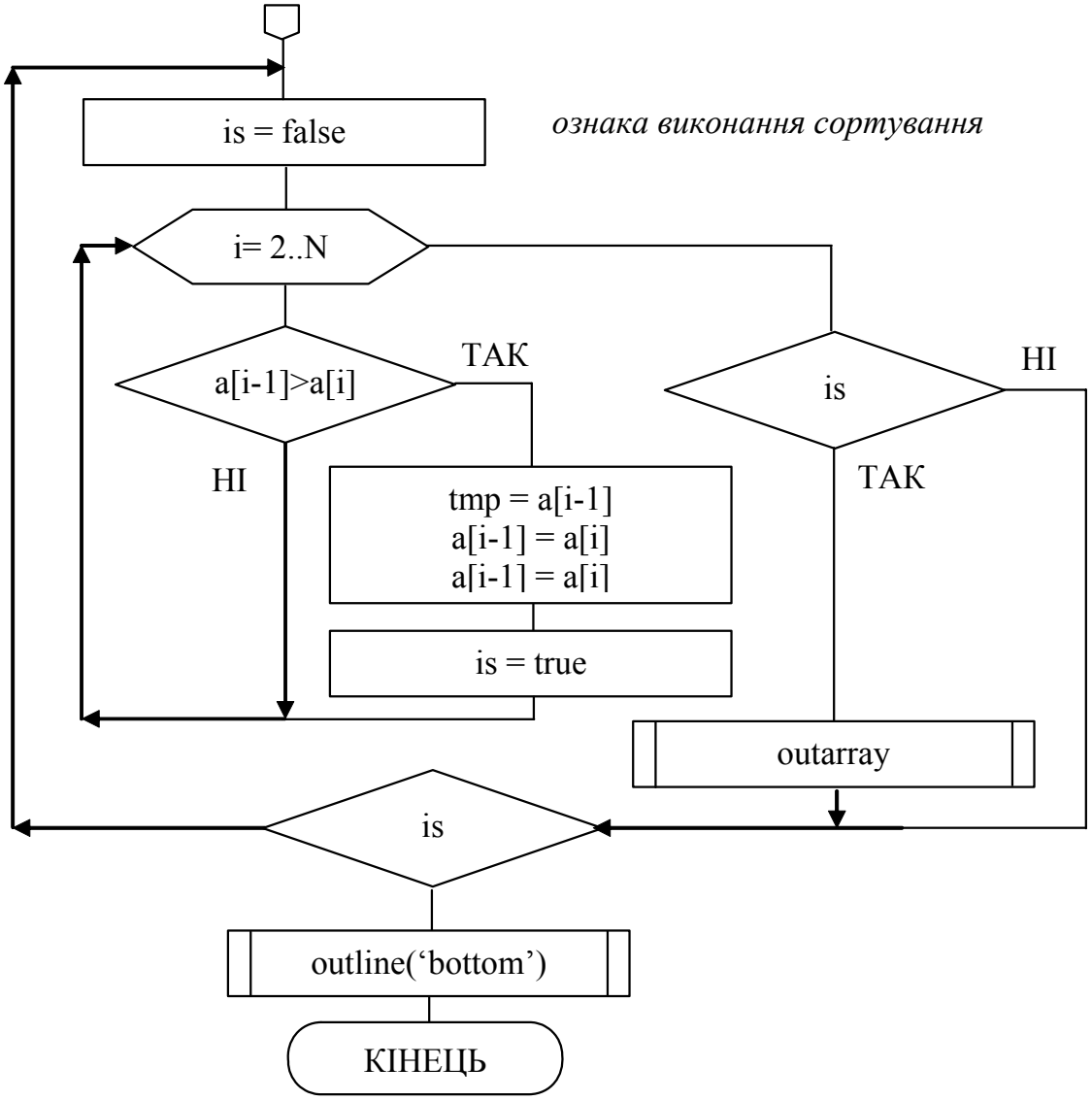
В цій процедурі використовується ще одна процедура `doubleline` (без параметрів), яка виводить на екран подвійну лінію над чи під таблицею (верхня чи нижня лінія). Ця процедура буде проводити подвійну лінію в кожній клітинці першого рядка (над таблицею) та в кожній клітинці останнього рядка (під таблицею). Довжина такої лінії дорівнює величині клітинки таблиці – 5 символів. Отже, блок-схема процедури `doubleline`:

процедура doubleline – відображення 5 символів подвійної лінії



Для виведення самих елементів масиву в рядок та одинарної лінії між ними (елемент таблиці), використаємо ще одну процедуру – outarray, блок-схема якої наведена нижче:





2. Написати програму реалізації задачі.

   Програма реалізації сортування елементів масиву за складеною блок-схемою буде мати вигляд:

```
{*************************************************
 * sort - програма сортування масиву методом     *
 *        бульбашок                              *
 * Copyright (c) Шищук В.В. гр.ІС-04-1, 28/11/04 *
 *************************************************}
PROGRAM sort;
CONST N=10;
VAR
   a: array[1..N] of integer;
   i,tmp: integer;
   is : boolean;
PROCEDURE doubleline;
```

```
VAR
   i:integer;
BEGIN
   for i:=1 to 5 do
       write(chr(205));
END;

PROCEDURE outline(what:string);
VAR
   symb1,symb2,symb3: char;
BEGIN
   if what='top' then
      begin
         symb1:=chr(201);
         symb2:=chr(209);
         symb3:=chr(187);
      end
   else
      begin
         symb1:=chr(200);
         symb2:=chr(207);
         symb3:=chr(188);
      end;
   write(symb1);
   for i:=1 to N-1 do
      begin
         doubleline;
         write(symb2);
      end;
   doubleline;
   writeln(symb3);
END;

PROCEDURE outarray;
VAR
   i: integer;
BEGIN
   write(chr(186),a[1]:4,' ');
   for i:=2 to N do
       write(chr(179),a[i]:4,' ');
   writeln(chr(186));
END;

BEGIN
```

```
   writeln('СОРТУВАННЯ БУЛЬБАШКАМИ');
   randomize;
   for i:=1 to N do
      a[i]:=random(100)-50;
   outline('top');
   outarray;
   repeat
      is:=false;
      for i:=2 to N do
         if a[i-1]>a[i] then
            begin
               tmp:=a[i-1];
               a[i-1]:=a[i];
               a[i]:=tmp;
               is:=true;
            end;
      if is then outarray;
   until not is;
   outline('bottom');
   write('Для завершення натисніть <Enter>:');
   readln
END.
```

3. Виконати компіляцію та покрокове виконання програми (процедури виведення на екран).
4. Зберегти написану програму в персональну папку.
5. Оформити звіт про виконану роботу.

Індивідуальні завдання:

рівень А

Заповнити одновимірний масив випадковими цілими числами. Вивести масив на екран. Здійснити необхідну обробку масиву, результат вивести на екран.

Варіант 1.

В заданому одновимірному масиві $m(m_1, m_2 ... m_{20})$ відсортувати елементи за зростанням методом Шелла.

Варіант 2.

В заданому одновимірному масиві $m(m_1, m_2 ... m_{20})$ відсортувати елементи за зростанням методом вибору.

Варіант 3.

В заданому одновимірному масиві $m(m_1, m_2 ... m_{20})$ відсортувати елементи за зростанням методом вставок.

Варіант 4.

Використовуючи метод послідовного пошуку знайти найменший та найбільший елементи одновимірного масиву.

Варіант 5.

В заданому одновимірному масиві $m(m_1, m_2 ... m_{20})$ відсортувати елементи за спаданням методом бульбашок.

Варіант 6.

Використовуючи метод бінарного пошуку знайти найменший та найбільший елементи одновимірного масиву.

Варіант 7.

В заданому одновимірному масиві $m(m_1, m_2 ... m_{20})$ відсортувати елементи за спаданням методом вставок.

Варіант 8.

Відсортувати першу половину масиву методом вибору, а другу – методом вставок.

Варіант 9.

Відсортувати першу половину масиву методом Шелла, а другу – методом вставок.

Варіант 10.

Відсортувати першу половину масиву методом вставок, а другу – методом бульбашок.

Варіант 11.

Відсортувати елементи масиву методом бульбашок до попередньо знайденого максимального елемента.

Варіант 12.

Відсортувати елементи масиву методом вибору до попередньо знайденого мінімального елемента.

Варіант 13.

Відсортувати методом вставки елементи масиву, що розміщені між максимальним та мінімальним елементами.

Варіант 14.

Відсортувати методом вибору елементи масиву, які розміщені між максимальним та мінімальним елементами.

Варіант 15.

В заданому одновимірному масиві $m(m_1, m_2 ... m_{20})$ відсортувати елементи за спаданням методом Шелла.

рівень Б

Заповнити двовимірний масив випадковими цілими числами. Вивести масив на екран. Здійснити необхідну обробку масиву, результат вивести на екран.

Варіант 1.

На прикладі двовимірного масиву порівняти сортування методом бульбашок та методом вибору.

Варіант 2.

На прикладі двовимірного масиву порівняти сортування методом вставки та методом вибору.

Варіант 3.

Відсортувати стовпчики прямокутної матриці таким чином, щоб сума парних від'ємних елементів стовпчиків збільшувалася. Сортування здійснити методом вибору.

Варіант 4.

На прикладі двовимірного масиву порівняти сортування методом вставки, методом вибору та методом Шелла.

Варіант 5.

За допомогою бінарного пошуку знайти, чи міститься в масиві значення, зчитане з клавіатури, якщо так, то в якій позиції.

Варіант 6.

Відсортувати стовпчики прямокутної матриці таким чином, щоб сума елементів стовпчиків збільшувалася. Сортування здійснити методом вибору.

Варіант 7.

Відсортувати рядки прямокутної матриці таким чином, щоб сума модулів непарних від'ємних елементів рядка збільшувалася. Сортування здійснити методом бульбашок.

Варіант 8.

Відсортувати методом вставки рядки прямокутної матриці таким чином, щоб сума елементів рядків збільшувалася.

Варіант 9.

На прикладі двовимірного масиву порівняти сортування методом вставки та методом бульбашок.

Варіант 10.

Відсортувати рядки прямокутної матриці таким чином, щоб сума парних елементів на парних номерах стовпчиків збільшувалася. Сортування здійснити методом вибору.

Варіант 11.

Відсортувати стовпчики прямокутної матриці таким чином, щоб сума елементів стовпчиків збільшувалася. Сортування здійснити методом бульбашок.

Варіант 12.

Відсортувати стовпчики прямокутної матриці таким чином, щоб кількість непарних від'ємних елементів стовпчиків збільшувалася. Сортування здійснити методом вставок.

Варіант 13.

Відсортувати стовпчики прямокутної матриці таким чином, щоб кількість від'ємних елементів на непарних номерах рядків збільшувалася. Сортування здійснити методом Шелла.

Варіант 14.

Відсортувати стовпчики прямокутної матриці таким чином, щоб сума елементів стовпчиків збільшувалася. Сортування здійснити методом Шелла.

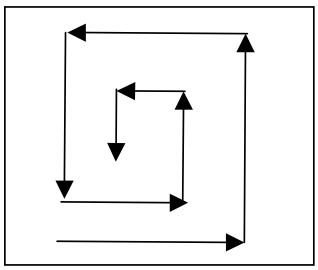
Варіант 15.

Відсортувати методом вибору стовпчики прямокутної матриці таким чином, щоб кількість від'ємних елементів стовпчиків збільшувалася.

рівень В

Заповнити двовимірний масив випадковими цілими числами. Вивести масив на екран. Здійснити необхідну обробку масиву, результат вивести на екран.

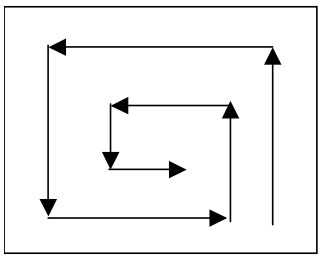
Варіант 1.



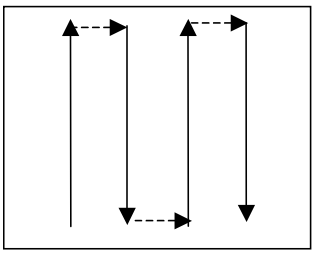
Задану прямокутну матрицю відсортувати методом бульбашок так, щоб послідовність елементів була такою, як зображено на схемі:

Варіант 2.

Задану прямокутну матрицю відсортувати методом вставки так, щоб послідовність елементів була такою, як зображено на схемі:

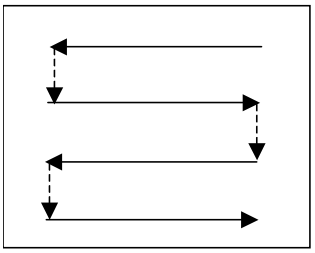


Варіант 3.



Задану прямокутну матрицю відсортувати методом вибору так, щоб послідовність елементів була такою, як зображено на схемі:

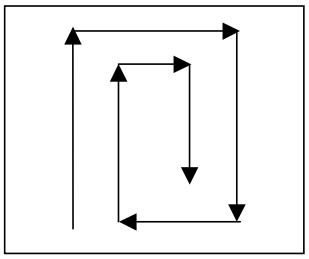
Варіант 4.



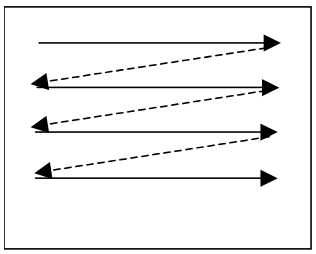
Задану прямокутну матрицю відсортувати методом вибору так, щоб послідовність елементів була такою, як зображено на схемі:

Варіант 5.

Задану прямокутну матрицю відсортувати методом бульбашок так, щоб послідовність елементів була такою, як зображено на схемі:

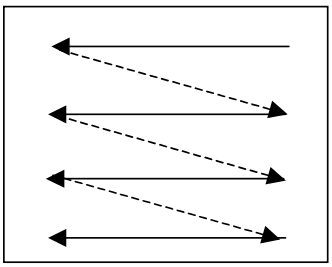


Варіант 6.



Задану прямокутну матрицю відсортувати методом бульбашок так, щоб послідовність елементів була такою, як зображено на схемі:

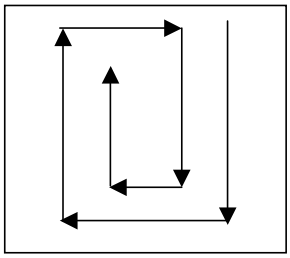
Варіант 7.



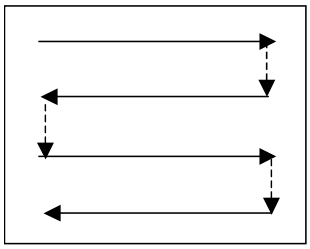
Задану прямокутну матрицю відсортувати методом вибору так, щоб послідовність елементів була такою, як зображено на схемі:

Варіант 8.

Задану прямокутну матрицю відсортувати методом вибору так, щоб послідовність елементів була такою, як зображено на схемі:



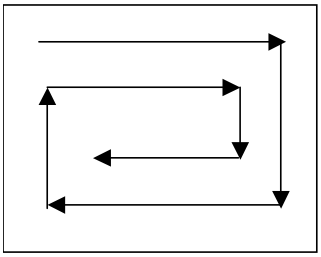
Варіант 9.



Задану прямокутну матрицю відсортувати методом вставки так, щоб послідовність елементів була такою, як зображено на схемі:

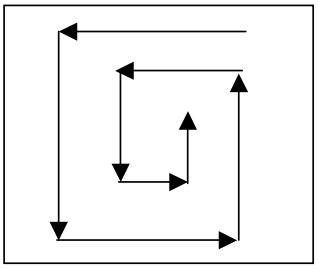
Варіант 10.

Задану прямокутну матрицю відсортувати методом вибору так, щоб послідовність елементів була такою, як зображено на схемі:

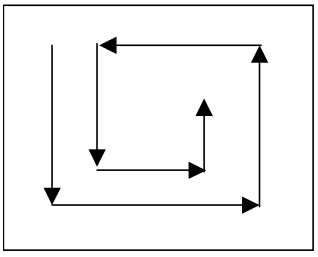


Варіант 11.

Задану прямокутну матрицю відсортувати методом бульбашок так, щоб послідовність елементів була такою, як зображено на схемі:



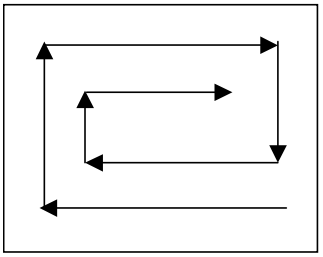
Варіант 12.



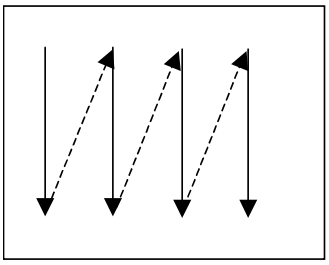
Задану прямокутну матрицю відсортувати методом вставки так, щоб послідовність елементів була такою, як зображено на схемі:

Варіант 13.

Задану прямокутну матрицю відсортувати методом бульбашок так, щоб послідовність елементів була такою, як зображено на схемі:



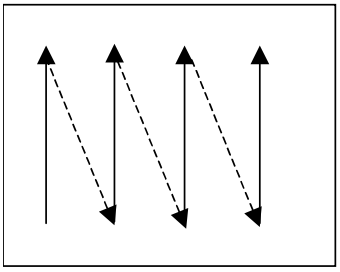
Варіант 14.



Задану прямокутну матрицю відсортувати методом бульбашок так, щоб послідовність елементів була такою, як зображено на схемі:

Варіант 15.

Задану прямокутну матрицю відсортувати методом бульбашок так, щоб послідовність елементів була такою, як зображено на схемі:



Контрольні запитання:

1) Що розуміють під пошуком та сортуванням даних?
2) Які дані можна відсортувати і яким методом?
3) Які методи сортування Вам відомі?
4) Назвіть всі відомі Вам методи пошуку?
5) Який з методів сортування та пошуку найшвидший, який найповільніший?
6) Що передбачає бінарний пошук?
7) Розкажіть принцип методу Шелла.
8) Розкажіть принцип методу вибору.
9) Коли застосовується метод послідовного пошуку, а коли бінарного?
10) Що таке символи псевдографіки і як їх використовувати?