 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №13**

з дисципліни «Алгоритмізація та програмування»

на тему: «Алгоритми сортування»

Варіант №2

**Виконав:**

студент гр. БС-71

Батуркіна А.М.

**Перевірив:**

ас. каф. БМК

Рисін С. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2018

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

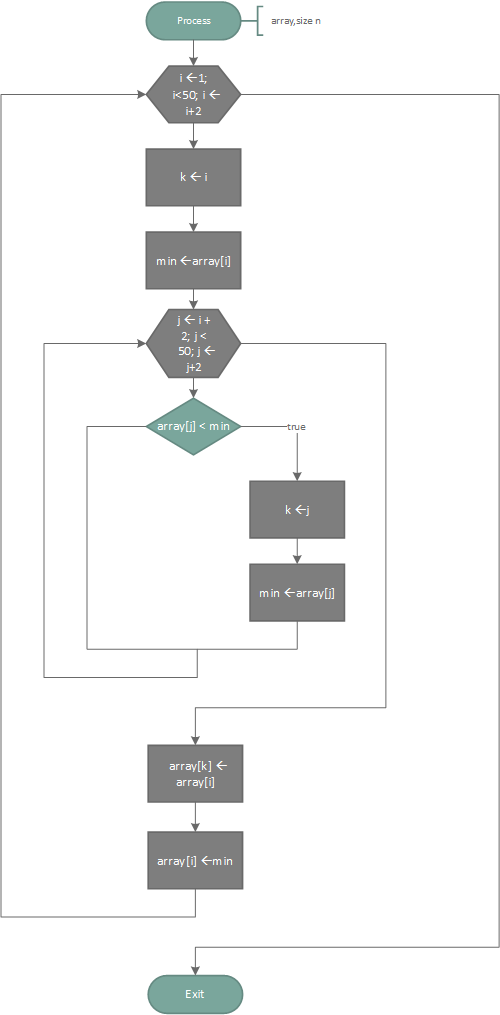
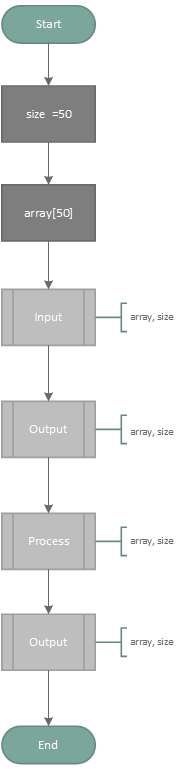
**Завдання:**

1. Ознайомитися з теоретичними відомостями сортування масивів, розглянути алгоритми бульбашкового, коктейльного сортування, сортування гребінцем, швидкого сортування, сортування простими вставками, сортування Шелла, злиттям, вибором та пірамідального сортування.
2. Побудувати блок-схему алгоритму для вирішення завдання відповідно до свого варіанту.
3. Відповідно до свого варіанту розробити програмний застосунок, що включає створення масиву цілих чисел, заповнення його елементів випадковими pгенерованими значеннями з діапазону [0, 50], сортування і форматований вивід на екран початкового і відсортованого масиву (глобальні змінні не використовувати, розміри матриці задавати з клавіатури, функція main має бути призначена тільки для виклику функцій користувача):

*Мається одновимірний масив довжиною* ***N=50****. Відсортувати алгоритмом сортування вибором елементи масиву, що знаходяться на парних позиціях, в такий чин, щоб вони розташовувались за збільшенням.*

1. Скласти та захистити звіт по роботі.

**Блок-схеми:**



**Лістинг:**

#include <iostream>

#include <conio.h>

#include <cstdlib>

#include <ctime>

using namespace std;

void Input(int array[], const int n)

{

srand(time(NULL));

cout << "RAND ARRAY is : ";

for (int i = 0; i < 50; i++)

{

array[i] = rand() % 50;

}

}

void Process(int array[], const int n)

{

for (int i = 1; i < 50; i += 2)

{

int k = i;

int min = array[i];

for (int j = i + 2; j < 50; j += 2)

if (array[j] < min)

{

k = j;

min = array[j];

}

array[k] = array[i];

array[i] = min;

}

cout << "NEW ARRAY is: ";

}

void Output(int array[], const int n)

{

cout << endl << "ARRAY ---> ";

for (int i = 0; i < 50; i++)

{

cout << array[i] << " ";

}

cout << endl;

}

int main()

{

const int size = 50;

int array[size] = {};

Input(array, size);

Output(array, size);

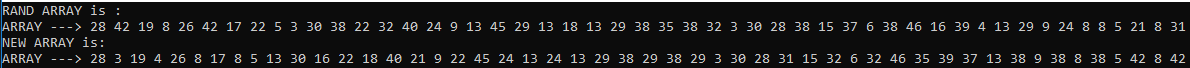
Process(array, size);

Output(array, size);

\_getch();

}

**Результати:**



**Контрольні запитання:**

**1. Що таке сортування, яка його основна мета?**- Сортування-це процес перегрупування заданої множини об’єктів, що приводить до їх впорядкованого розташування щодо ключа. Мета сортування- полегшити подальший пошук елементів.

**2. Чи впливає тип даних на час виконання алгоритму сортування?**- Так, впливає.

**3. Чому не існує універсального алгоритму сортування?**   
- Тому що для різних вхідних даних використовують різні, оптимальні для даної задачі, методи.

**4. Чим можна пояснити різноманітність алгоритмів сортування?**

- Різноманітністю задач, які необхідно вирішити за допомогою сортування.

Алгоритми відрізняються за наступними параметрами:

- час сортування;

- пам’ять;

- cтійкість ;

- природність поведінки

**5. Як визначити, яким алгоритмам сортування віддати перевагу при вирішення завдання?**   
- Потрібно оцінити виділення додаткової пам'яті, час на виконання алгоритму, формат представлення даних, стійкість, природність поведінки, сферу його застосування.

В залежності від кількості та попередньої відсортованості вхідних даних, достатньої швидкодії, необхідності стійкості. При вирішенні простих задач різниці у швидкодії майже немає. Класифікація за швидкодією:

За час O(n2): сортування вибором, сортування вставкою, сортування обміном, сортування методом бінарної вставки. За час O(n\*logn): пірамідальне сортування, швидке сортування, сортування злиттям, За час O(n) з використанням додаткової інформації про елементи: сортування підрахунком, сортування за розрядами, сортування комірками. За час O(n\*(logn)2): сортування злиттям модифіковане, сортування Шелла. За час O(nn!): Сортування перестановкою, Випадкове сортування.

**6. Які на сьогоднішній день найефективніші методи сортування?**  
- Найбільш поширеними алгоритмами для сортувань масивів,що є алгоритми: MergeSort (сортування злиттям), QuickSort (швидкого сортування або метод Хоара), Radixsort (байтове, радіксне або розподіляюче сортування), для прикладних задач, які застосовують елементи сортування дуже корисне HeapSort (пірамідальне сортування).

Швидке сортування (сортування Хоара) є найбільш ефективним алгоритмом з усіх відомих методів сортування, але всі вдосконалені методи мають один спільний недолік - невисоку швидкість роботи при малих значеннях n.

**7.Чому алгоритми швидкого сортування не дають великого виграшу при малих розмірах масивів?**   
- Швидке сортування по своїй суті нецікаве для малих масивів, тому що використовується рекурсивний метод, на обробку якого потрібен час, тому при сортуванні масивів малих розмірів це не є доцільним. Кожний прохід містить мінімум 256 ітерацій, тому при невеликих п загальний час виконання (n+m ) визначається константою m=256.

**8. За рахунок чого в алгоритмах швидкого сортування відбувається виграш при виконанні операцій порівняння і перестановок?**- Цей метод розглядає масив, як список значень. Спочатку виділяється середнє значення як сепаратор (фактор розбиття) списку. Список розбивається на два: в одному з них значення менше сепаратора, а в іншому - більше або рівні. Далі процедура сортування рекурсивно викликає саму себе для кожного з двох списків. Кожен раз при виклику сортування список елементів розбивається на два менших.

**9.  У чому переваги і недоліки по відношенню один до одного наступних алгоритмів сортування: пірамідальне сортування, сортування злиттям, сортування Шелла та сортування Хоара?**  
**Пірамідальне:**

Переваги: Має доведену оцінку гіршого випадку O (nlogn). Сортує на місці, тобто вимагає всього O (1) додаткової пам'яті.

Недоліки: Складний в реалізації. Нестійкий - для забезпечення стійкості потрібно розширювати ключ.

**Злиттям*:***

Переваги: При витраті пам'яті O (n) швидше (O (nlog n) з меншою константою) і не схильна деградації на невдалих даних (порівняно з пірамідальним сортуванням)

Недоліки: Через складність алгоритму виграш виходить тільки на великих n (порівняно з пірамідальним сортуванням)

**Сортування Шелла**

Переваги: На невеликих n (до декількох тисяч) швидше за сортування злиттям- відсутність потреби в пам'яті під стек; відсутність деградації при невдалих наборах даних - швидке сортування (Хоара) легко деградує до O (n ²), що гірше, ніж найгірше гарантований час для сортування Шелла.

Недоліки: сортування Шелла в багатьох випадках повільніше, ніж швидке сортування Хоара.

**Сортування Хоара:**

Переваги: Якщо розбиття збалансоване, то асимптотично алгоритм працює так само швидко як і алгоритм сортування злиттям. Простий в реалізації.

Недоліки: Нестійкий.

**10.Чим відрізняється внутрішнє сортування від зовнішнього?**  
- При внутрішньому сортуванні працюють з даними в оперативній пам’яті з довільним доступом, а при зовнішньому впорядковують інформацію, розташовану на зовнішніх носіях.