Харківський національний університет імені В. Н. Каразіна Факультет комп'ютерних наук
Кафедра штучного інтелекту та програмного забезпечення

ЗВІТ
Практична робота №3
дисципліна: «Теорія алгоритмів»

Виконав: студент групи КС-22
Ковальов Андрій
Перевірив: викладач
Олешко Олег Іванович

Харків

Завдання.

На прямій дощечці вбиті гвіздки. Будь-які два гвіздки можна з'єднати ниточкою. Потрібно з'єднати деякі пари гвіздків ниточками так, щоб до кожного гвіздка була прив'язана хоча б одна ниточка, а сумарна довжина всіх ниточок була мінімальна.

Вхідні дані: У першому рядку вхідного потоку записане число N - кількість гвіздків (2 <= N <= 100). У наступному рядку записано N чисел - координати всіх гвіздків (невід'ємні цілі числа, що не перевершують 10000).

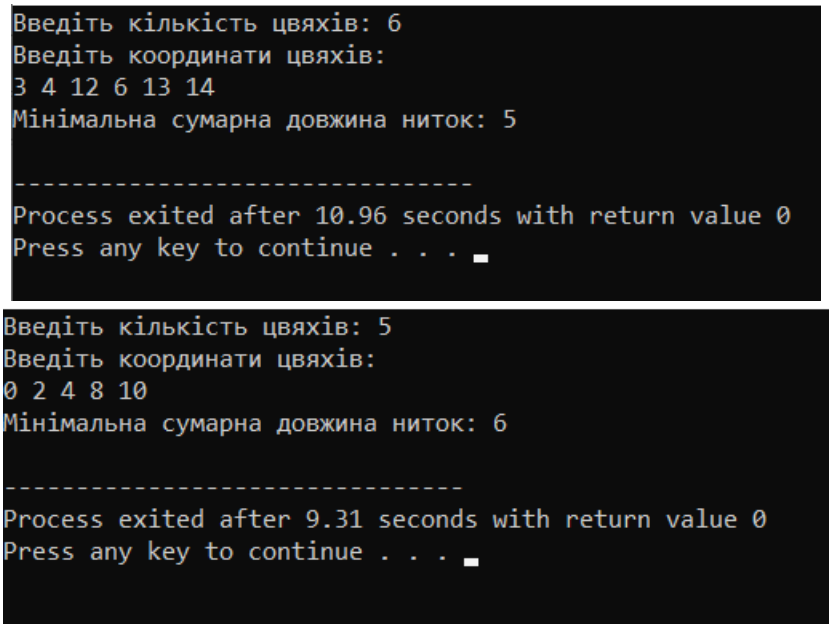
Вихідні дані: мінімальна сумарна довжина всіх ниточок

```
#include <stdio.h>

        // Функція сортування бульбашкою за зростанням
        int BubbleSort(int elem[], int k) {
            for (int b = 0; b < k - 1; b++) {
                for (int j = 0; j < k - b - 1; j++) {
                    if (elem[j] > elem[j + 1]) {
                        // Міняємо елементи місцями
                        int temp = elem[j];
                        elem[j] = elem[j + 1];
                        elem[j + 1] = temp;
                    }
                    else if (elem[j] == elem[j + 1])
                        return 0;  // Якщо зустрічаються однакові координати
                }
            }
            return 1;  // Якщо всі координати унікальні
        }

        int main() {
            int total;

        // Запитуємо кількість цвяхів
            printf("Введіть кількість цвяхів: ");
            scanf("%d", &total);

            int position[total];  // Масив для координат цвяхів
            int distance[total];  // Масив для мінімальних відстаней між нитками
```

```
        // Запитуємо координати цвяхів
            printf("Введіть координати цвяхів:\n");
            for (int b = 0; b < total; b++) {
                scanf("%d", &position[b]);
            }

        // Сортуємо масив координат за зростанням
            int result = BubbleSort(position, total);
            if (!result)
                return printf("Не може бути цвяхів з однаковими координатами.\n");  // Вив
помилку, якщо координати повторюються

        // Ініціалізуємо початкові значення для масиву відстаней
            distance[0] = 0;  // Перший цвях не має попередника
            distance[1] = position[1] - position[0];  // Для другого цвяха відстань — це різни
його координатою та першим
            distance[total - 2] = 0;  // Ініціалізація для передостаннього цвяха

        // Обчислюємо мінімальні відстані для всіх цвяхів
            for (int b = 2; b < total - 2; b++) {
                int next = position[b + 1] - position[b];
                int prev = position[b] - position[b - 1];

                if (next < prev || (next == prev && distance[b-1])) {
                    distance[b] = 0;
                    distance[b + 1] = next;
                }
                else if (next > prev || (next == prev && !distance[b-1])) {
                    distance[b] = prev;  // Якщо відстань до попереднього цвяха менша, встанов
її
                }
            }

        // Відстань для останнього цвяха — це різниця між останньою і передостан
координатами
            distance[total - 1] = position[total - 1] - position[total - 2];

        // Обчислюємо мінімальну сумарну довжину ниток
            int amount = 0;
```

```
    for (int b = 0; b < total; b++) {
        amount += distance[b];  // Додаємо всі відстані
    }

// Виводимо результат — мінімальну сумарну довжину ниток
    printf("Мінімальна сумарна довжина ниток: %d\n", amount);

    return 0;
}
```

Лістинг 1 – код програми



Скріншот 1 – виконання програми