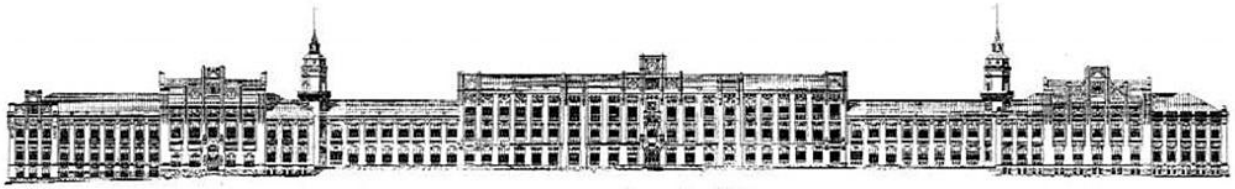

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Фізико-Технічний Інститут

Програмування

Лабораторна робота №4

Робота з динамічною пам’яттю

Варіант 4

Виконала:

Студентка 2 курсу ФТІ
групи ФФ-41

Щербина Софія

Перевірив:

_____

Мета: отримати навички роботи масивами, вказівниками, посиланнями та динамічним виділенням пам’яті.

Робоче завдання

1. Проаналізувати умови задач. Варіанти у Табл 4.1.
2. Розробити алгоритми та створити програми розв’язання задачі згідно з номером варіанту.
3. Підготувати відповіді на контрольні питання.
4. Результати роботи оформити протоколом. Приєднати до класу і захистити.

Завдання 1а.

Реалізувати завдання з використанням динамічного масиву. Для доступу до елементів масиву використати два підходи: через явне розіменування вказівника (не через імітацію індекса, а використовуючи переміщення вказівника безпосередньо) та через використання індексу. Якщо похідний масив буде іншого розміру, треба передбачити виділення під нього пам'яті потрібного розміру. Ввести з клавіатури кількість елементів у масиві Arr[N]. Створити динамічний масив розміром N. Забезпечити необхідну реакцію програми на некоректні ситуації. Масив створити з елементами цілого типу. Заповнити будь-яким методом, можна ввести з клавіатури, можна заповнити випадковими числами, можна прочитати з файла. Перевірити правильність виконання програми за допомогою кількох тестових варіантів.

4	Напишіть програму, яка створює новий масив з вхідного “викинувши” з нього всі парні числа.

Код реалізації завдання

```
#include <iostream>
// #include <ctime>
using namespace std;

int main() {
    int N;

    cout << "Enter number of elements: ";
    cin >> N;

    int* Arr = new int[N];

    srand(time(nullptr));
    cout << "Original array:";
    for (int i = 0; i < N; i++) {
        Arr[i] = rand() % 100;
        cout << Arr[i] << " ";
    }
    cout << endl;

// ......

    int countOdd = 0;
    for (int i = 0; i < N; i++) {
        if (Arr[i] % 2 != 0) countOdd++;
    }


    if (countOdd == 0) {
        cout << "No odd elements found." << endl;
        delete[] Arr;
        return 0;
    }

    int* NewArr = new int[countOdd];

    int* ptrArr = Arr;
    int* ptrNew = NewArr;
    for (int i = 0; i < N; i++) {
        if (*(ptrArr + i) % 2 != 0) {
            *ptrNew = *(ptrArr + i);
            ptrNew++;
        }
    }

    cout << "New array (odd elements only) - using index:";
    for (int i = 0; i < countOdd; i++) {
        cout << NewArr[i] << " ";
    }

    cout << "New array (odd elements only) - using pointer
arithmetic:";
    ptrNew = NewArr;
    for (int i = 0; i < countOdd; i++) {
        cout << *(ptrNew + i) << " ";
    }

    delete[] Arr;
    delete[] NewArr;

    return 0;
}
```

Результати виконання завдання програми

```
PS C:\prog_proj\lab_4> ./main.exe
Enter number of elements: 14
Original array: 41 67 34 0 69 24 78 58 62 64 5 45 81 27
New array (odd elements only) - using index:
41 67 69 5 45 81 27
New array (odd elements only) - using pointer arithmetic:
41 67 69 5 45 81 27
```

```
PS C:\prog_proj\lab_4> ./main.exe
Enter number of elements: 20
Original array: 41 67 34 0 69 24 78 58 62 64 5 45 81 27 61 91 95 42 27 36
New array (odd elements only) - using index:
41 67 69 5 45 81 27 61 91 95 27
New array (odd elements only) - using pointer arithmetic:
41 67 69 5 45 81 27 61 91 95 27
```

Висновки:

У цій лабораторній роботі було відпрацьовано роботу з динамічними масивами та вказівниками в C++. Програма формує новий масив, виключаючи парні числа, і демонструє два способи доступу до елементів: через індекси та через явне розіменування вказівника з його пересуванням. Робота закріпила навички виділення та звільнення пам’яті, обробки елементів масиву та використання вказівників для ефективної роботи з даними.