

Лабораторна робота 7

Одновимірні масиви. Статичні й динамічні масиви. Функції

Варіант 1

Задайте два масиви $A(n)$ і $B(m)$. Виконайте такі завдання:

— визначте масив $C(n+m)$ як такий, що об'єднує масиви A і B . Упорядкуйте за збільшенням окремо елементи, що стоять на парних місцях, і елементи, що стоять на непарних місцях;

— замініть всі від'ємні елементи масиву A на корінь із максимального додатного елемента масиву A (обчислений раніше);

— змініть порядок розташування елементів у масиві B на зворотний.

Код

```
#include <iostream>
#include <Windows.h>
#include <cmath>
#include <algorithm>

using namespace std;

const int MAX_SIZE = 100;

void mergeArrays(int A[], int n, int B[], int m, int C[]) {
    int i = 0, j = 0, k = 0;

    while (i < n && j < m) {
        if (A[i] < B[j]) {
            C[k] = A[i];
            i++;
        }
        else {
            C[k] = B[j];
            j++;
        }
        k++;
    }

    while (i < n) {
        C[k] = A[i];
        i++;
        k++;
    }

    while (j < m) {
        C[k] = B[j];
        j++;
    }
}
```

```

        k++;
    }
}

void sortElements(int arr[], int size) {
    for (int i = 0; i < size; i += 2) {
        for (int j = i + 2; j < size; j += 2) {
            if (arr[i] > arr[j]) {
                std::swap(arr[i], arr[j]);
            }
        }
    }

    for (int i = 1; i < size; i += 2) {
        for (int j = i + 2; j < size; j += 2) {
            if (arr[i] > arr[j]) {
                std::swap(arr[i], arr[j]);
            }
        }
    }
}

void replaceNegativeElements(int arr[], int size, int maxPositive) {
    for (int i = 0; i < size; i++) {
        if (arr[i] < 0) {
            arr[i] = std::sqrt(maxPositive);
        }
    }
}

void reverseArray(int arr[], int size) {
    std::reverse(arr, arr + size);
}

int main() {
    int n, m;
    int A[MAX_SIZE], B[MAX_SIZE], C[MAX_SIZE];

    setlocale(LC_ALL, "Russian");
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);

    std::cout << "Введіть розмір масиву A: ";
    std::cin >> n;

    std::cout << "Введіть розмір масиву B: ";
    std::cin >> m;

    std::cout << "Введіть елементи масиву A: ";
    for (int i = 0; i < n; i++) {
        std::cin >> A[i];
    }

    std::cout << "Введіть елементи масиву B: ";
    for (int i = 0; i < m; i++) {
        std::cin >> B[i];
    }

    mergeArrays(A, n, B, m, C);

    sortElements(C, n + m);

    int maxPositive = 0;
    for (int i = 0; i < n; i++) {
        if (A[i] > 0 && A[i] > maxPositive) {
            maxPositive = A[i];
        }
    }
}

```

```
replaceNegativeElements(A, n, maxPositive);

reverseArray(B, m);

std::cout << "Результуючий масив C: ";
for (int i = 0; i < n + m; i++) {
    std::cout << C[i] << " ";
}
std::cout << std::endl;

std::cout << "Масив A після заміни від'ємних елементів: ";
for (int i = 0; i < n; i++) {
    std::cout << A[i] << " ";
}
std::cout << std::endl;

std::cout << "Масив B після зміни порядку розташування: ";
for (int i = 0; i < m; i++) {
    std::cout << B[i] << " ";
}
std::cout << std::endl;

return 0;
}
```