

```

/*****
/* Program sortuje tablicę liczb całkowitych */
/* w porządku niemalejącym. */
/* Algorytm sortowania przez wstawianie. */
*****/
#include <iostream>
using namespace std;
int main()
{
    const int max_n = 100;
    int A[max_n], n;

    // wczytanie danych
    cout << "Ile liczb chcesz posortować? (max=100) ";
    cin >> n;
    cout << "Wprowadź liczby " << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "A[" << i << "] = ";
        cin >> A[i];
    }

    // wypisanie tabeli
    cout << "Tabela przed sortowaniem:" << endl;
    for (int i = 0; i < n; i++)
        cout << A[i] << " ";

    // sortowanie przez wstawianie
    int i, j, pom;
    for (i = 0; i < n-1; i++)
    {
        j = i + 1;
        pom = A[j];
        while ((j > 0) && (A[j-1] > pom))
        {
            A[j] = A[j-1];
            j--;
        }
        A[j] = pom;
    }

    // wypisanie posortowanej tabeli
    cout << "\nTabela po sortowaniu:" << endl;
    for (i = 0; i < n; i++)
        cout << A[i] << " ";
}

```

PRZYKŁAD DZIAŁANIA PROGRAMU:

```

Ile liczb chcesz posortować? (max=100) 6
Wprowadź liczby
A[0] = 5
A[1] = 2
A[2] = 8
A[3] = 0
A[4] = 3
A[5] = 6
Tabela przed sortowaniem:
5 2 8 0 3 6
Tabela po sortowaniu:
0 2 3 5 6 8

```

```

/*****
/* Program sortuje tablicę liczb całkowitych
/* w porządku niemalejącym.
/* Algorytm sortowania bąbelkowego.
*****/
#include <iostream>
using namespace std;
int main()
{
    const int max_n = 100;
    int A[max_n], n;

    // wczytanie danych
    cout << "Ile liczb chcesz posortować? (max=100) ";
    cin >> n;
    cout << "Wprowadź liczby " << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "A[" << i << "] = ";
        cin >> A[i];
    }

    // wypisanie tabeli
    cout << "Tabela przed sortowaniem:" << endl;
    for (int i = 0; i < n; i++)
        cout << A[i] << " ";

    // sortowanie bąbelkowe
    int i, j, pom;
    for (i = 1; i <= n; i++)
    {
        for (j = n-1; j >= i; j--)
        {
            if (A[j] < A[j-1])
            {
                pom = A[j-1];
                A[j-1] = A[j];
                A[j] = pom;
            }
        }
    }

    // wypisanie posortowanej tabeli
    cout << "\nTabela po sortowaniu:" << endl;
    for (i = 0; i < n; i++)
        cout << A[i] << " ";
}

```

PRZYKŁAD DZIAŁANIA PROGRAMU:

```

Ile liczb chcesz posortować? (max=100) 6
Wprowadź liczby
A[0] = 5
A[1] = 2
A[2] = 8
A[3] = 0
A[4] = 3
A[5] = 6
Tabela przed sortowaniem:
5 2 8 0 3 6
Tabela po sortowaniu:
0 2 3 5 6 8

```

```

/* Program zamienia miejscami elementy macierzy leżące po przeciwnej stronie */
/* prostej pionowej dzielącej macierz na dwie równe części. */
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main ()
{
    int A[10][10], n, m;
    int i, j, pom;
    cout << "Podaj liczbę wierszy (n<=10): ";
    cin >> n;
    cout << "Podaj liczbę kolumn (m<=10): ";
    cin >> m;
    //wypełnienie macierzy liczbami losowymi 0-9
    srand(time(NULL));
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            A[i][j] = rand()%10;
        }
    }
    //wyświetlenie tablicy
    cout << "Wylosowana tablica:" << endl;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cout << A[i][j] << " ";
        }
        cout << endl;
    }
    //zamiana elementów w tablicy
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m/2; j++)
        {
            pom = A[i][j];
            A[i][j] = A[i][(m-1)-j];
            A[i][(m-1)-j] = pom;
        }
    }
    //wyświetlenie tablicy zmodyfikowanej
    cout << "Tablica zmodyfikowana:" << endl;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cout << A[i][j] << " ";
        }
        cout << endl;
    }
}

```

PRZYKŁAD DZIAŁANIA PROGRAMU:

```

Podaj liczbę wierszy (n<=10): 3
Podaj liczbę kolumn (m<=10): 7
Wylosowana tablica:
8 4 2 5 6 5 2
4 6 4 2 2 3 4
8 8 7 0 9 3 9
Tablica zmodyfikowana:
2 5 6 5 2 4 8
4 3 2 2 4 6 4
9 3 9 0 7 8 8

```