

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 7 з дисципліни
«Основи програмування-1.
Базові конструкції»

«Одновимірні масиви»

Варіант 28

Виконав студент ІП-11 Сідак Кирил Ігорович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2021

Мета – визначити особливості обробки однорідних масивів.

Варіант 28

В масиві цілих чисел $V(n)$ знайти число, яке зустрічається найбільш часто.

Якщо таких чисел декілька, то визначити найменше із них.

Постановка задачі:

Щоб знайти кількість разів, які зустрічається певне число в масиві, треба пройти по елементам масиву і якщо даний елемент дорівнює взятому елементу, то збільшуємо частоту на 1. Таким чином, шляхом порівняння частоти та значення числа, знайдемо число, яке зустрічається найбільш часто і є мінімальним, якщо таких чисел декілька.

Програма на C++:

/*

Варіант

28

В масиві цілих чисел $V(n)$ знайти число, яке зустрічається найбільш часто. Якщо таких чисел декілька, то визначити найменше із них. */

```
#include <iostream>
```

```
#include <ctime>
```

```
using namespace std;
```

```
int* generate_array(int); // функція для генерації масиву заданої довжини
```

```
void display_array(int*, int); // функція для відображення масиву заданої довжини
```

```
int most_frequent_num(int*, int); /* функція для знаходження числа, яке зустрічається  
набільш часто в масиві, або найменшого з них, якщо їх декілька */
```

```
int main()
```

```
{
```

```
    int* array;
```

```
    int n, result;
```

```
    cout << "Enter the length of the array: ";
```

```
    cin >> n;
```

```
    array = generate_array(n);
```

```
    display_array(array, n);
```

```
    result = most_frequent_num(array, n);
```

```
    cout << "The most frequent number(the least one among the most frequent ones) is " << result << " ";
```

```
    delete []array;
```

```

        return 0;
    }

    int* generate_array(int n)
    {
        int num;
        int* array = new int[n];
        srand(time(NULL));
        for (int i = 0; i < n; ++i)
        {
            num = rand()% 10;
            array[i] = num;
        }
        return array;
    }

    void display_array(int *arr, int n) {
        if (n == 1) {
            cout << "[" << arr[n-1] << "]"\\n";
        }
        else {
            for (int i = 0; i < n; ++i) {
                if (i == 0)
                {
                    cout << "[" << arr[i] << " , ";
                } else if (i == n - 1) {
                    cout << arr[i] << "]"\\n";
                } else {
                    cout << arr[i] << " , ";
                }
            }
        }
    }

    int most_frequent_num(int* arr, int n)
    {
        int temp_num, temp_count, count;
        temp_num = arr[0];
        temp_count = 1;
        for (int i = 0; i < n; i++)
        {
            count = 1;
            for (int j = i + 1; j < n; ++j) {
                if (i != j) {
                    if (arr[i] == arr[j])
                    {

```

```
6 using namespace std;
7 int* generate_array(int); // функція для генерації масиву заданої довжини
8 void display_array(int*, int); // функція для відображення масиву заданої довжини
9 int most_frequent_num(int*, int); /* функція для знаходження числа, яке зустрічається
10 найбільш часто в масиві, або найменшого з них, якщо їх декілька */
11 int main()
12 {
13     int* array;
14     int n, result;
15     cout << "Enter the length of the array: ";
16     cin >> n;
17     array = generate_array(n);
18     display_array(array, n);
19     result = most_frequent_num(array, n);
20     cout << "The most frequent number(the least one among the most frequent ones) is " << result << ".";
21     delete []array;
22     return 0;
23 }
```

generate_array

Run: Lab7 x

/Users/kyryl/Desktop/Lab7/cmake-build-debug/Lab7

Enter the length of the array: 8

[0, 3, 4, 2, 8, 0, 8, 6]

The most frequent number(the least one among the most frequent ones) is 0.

Process finished with exit code 0

```
        count += 1;
    }
}

if ((temp_count == count && arr[i] < temp_num) || (temp_count < count))
{
    temp_num = arr[i];
    temp_count = count;
}

return temp_num;
}
```

Результат на C++:

Висновок

Отже, я дослідив особливості обробки однорідних масивів, створивши алгоритм, який використовує цикли для проходження по елементам масиву, для пошуку числа, яке зустрічається найбільш часто в масиві або мінімального з них, якщо їх декілька.

```
6 using namespace std;
7 int* generate_array(int); // функція для генерації масиву заданої довжини
8 void display_array(int*, int); // функція для відображення масиву заданої довжини
9 int most_frequent_num(int*, int); /* функція для знаходження числа, яке зустрічається
10 найбільш часто в масиві, або найменшого з них, якщо їх декілька */
11 int main()
12 {
13     int* array;
14     int n, result;
15     cout << "Enter the length of the array: ";
16     cin >> n;
17     array = generate_array(n);
18     display_array(array, n);
19     result = most_frequent_num(array, n);
20     cout << "The most frequent number(the least one among the most frequent ones) is " << result << ".";
21     delete []array;
22     return 0;
23 }
```

generate_array

Run: Lab7

/Users/kyryl/Desktop/Lab7/cmake-build-debug/Lab7

Enter the length of the array: 13

[3, 7, 9, 5, 4, 0, 5, 4, 1, 8, 1, 9, 9, 2, 2, 6]

The most frequent number(the least one among the most frequent ones) is 9.

Process finished with exit code 0