## ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №1

## « Алгоритми обробки масивів.»

## з курсу «Алгоритми і структури даних»

## студента групи ПА-19-2

## Філіпенка Дмитра Миколайовича

## кафедра комп’ютернихтехнологій, ДНУ

## 2019/2020 н.р.

1. Постановка задачі.

Завдання. Розробити програму, яка реалізує такі дії:

- створює статичний масив із 100 елементів;

- заповнює масив n випадковими цілими числами із діапазону, границі якого

ввести з клавіатури. Значення n (n<;=100) також ввести з клавіатури;

- виводить заповнену частину (із n елементів) на екран;

- виконує обробку масиву із n елементів згідно з завданням за варіантами.

Якщо в результаті обробки масиву після видалення елементів масив виявиться

порожнім, видати відповідне повідомлення.

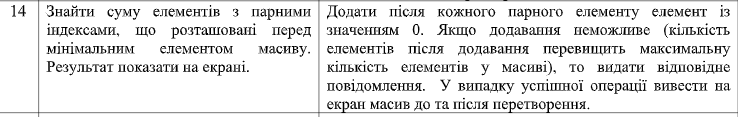
Якщо додавання елементів призведе до виходу за межі масиву (кількість елементів

перевищить 100), видати відповідне повідомлення.

Програма має бути оформлена таким чином, щоб всі функції та завдання можна

було б продемонструвати за один сеанс роботи програми. Для цього можна використати

текстове меню.



2. Опис розв’язку. За необхідністю навести блок-схему алгоритму.

Для цієї задачі я зробив 5 функцій щоб розбити програму на менші підпрограми та зменшити кількість повтору коду.

fill\_array – заповнення масиву

show\_array – відображення масиву

Task1 - задача №1

Task2 - задача №2

wait\_key – функція що очікує клавішу enter (використовується після виконання інших функцій)

Для заповнення масиву в функцію потрібно передати адресу масиву та кількість потрібних елементів. В ній також є перевірка на можливу кількість елементів, та min<max потім йде заповнення масиву до n-ого елементу .

Функція виводу приймає ті ж самі значення і в циклі відображує елементи масиву до n

Для задачі №1 використовується алгоритм пошуку мінімального елемента в масиві   
і його індексу.

Після того як пошук індексу завершився ми робимо ще один цикл for з 0 до знайденого індексу з кроком 2

І складуємо значення масиву з парними індексами.

\*примітка якщо масив пустий то ми отримаємо відповідне повідомлення

Для задачі №2

Можна використати уже готову функцію виводу і вивести значення масиву до зміни.  
Для того щоб не вискочити за межі масиву треба зробити перевірку n<N

Якщо ж ми вийдемо за цю межу то цикл зупиниться і буде повідомлення про переповнення масиву .

Щоб вставити значення 0 перед кожним парним елементом для початку треба зробити перевірку на парність Та не 0 значення (т.як якщо її не зробити то весь масив буде складатися з нулів)  
Якщо ми наштовхнулися на парний елемент ми беремо його індекс та робимо зсув всіх наступних елементів від кінця до цього індексу

І в «Порожню» ячейку вкладаємо значення 0

Після цього знову викликаємо нашу готову функцію show\_array Для перевірки результату.

3.Вихідний текст програми (з необхідними коментарями).

//

// main.cpp

// lab1 algoritms

//

// Created by Дима Филипенко on 14.02.2020.

// Copyright © 2020 Дима Филипенко. All rights reserved.

//

#include <iostream>

#include <ctime>

**using** **namespace** std;

**const** **int** N=100;

**bool** filled=**false**;

**bool** limit\_end=**false**;

**void** wait\_key(){

**do**

{

cout << '\n' << "Press a key to continue...";

} **while** (cin.get() != '\n');

getchar();

// clrscr();

// ClearScreen();

}

**void** fill\_array(**int**\* array,**int**& n){

**int** min,max;

**do**{

cout<<"Введите количество элементов в массиве: \nn=";cin>>n;

**if**(n<=0 || n>100){

cout<<"Ошибка - количество элементов может быть от 0 до 100 \n";

}

}**while**(n<=0 || n>100);

**do**{

cout<<"Введите диапазон чисел для заполнения данными \nmin=";cin>>min;

cout<<"max=";cin>>max;

**if**(max<min){

cout<<"Ошибка - min не может быть больше чем max\n";

}

}

**while**(max<min);

srand(time(0));

**int** x;

**for**(**int** i=0;i<n;i++){

x=rand()% (max-min+1) + min ;

array[i]=x ;

// cout<<x;

}

filled=**true**;

wait\_key();

}

**void** show\_array(**int** array[],**int**& n){

**if**(filled){

**for**(**int** i=0;i<n;i++){

cout<<"array["<<i<<"]="<<array[i]<<endl;

}

}**else**{

cout<<"Массив не заполнен";

}

// wait\_key();

}

**void** Task1(**int** array[],**int**& n){

**if**(filled){

**int** id=0;

**int** minimum=array[0];

**int** S=0;

**for**(**int** i=1;i<n;i++){

**if**(minimum>array[i]){

minimum=array[i];

id=i;

}

}

**if** (id==0){

cout<<"Нету суммы";

}**else**{

cout<<minimum<<" - минимальный элемент массива"<<endl;

**for**(**int** i=0;i<id;i+=2){

S+=array[i];

}

cout<<S<<" - сумма всех парных индексов перед минимальным элеметом"<<endl;

}

}**else**{

cout<<"Массив не заполнен";

}

wait\_key();

}

**void** Task2(**int** array[],**int**& n){

**if**(filled){

cout<<"До изменения"<<endl;

show\_array(array,n);

// int saved;

**for** (**int** i=0; i<n; i++) {

**if**(n<N){

**if**(array[i]%2==0 && array[i]!=0){

// saved=array[i+1];

n++;

**for**(**int** p=n; p>i+1; p--)

{

array[p]=array[p-1];

}

array[i+1]=0;

}

}**else**{

limit\_end = **true**;

**break**;

}

}

**if**(limit\_end){

cout<<"Массив не может дальше заполняться - достигнут предел в "<<N<<" Элементов";

}**else**{

cout<<"После изменения"<<endl;

show\_array(array,n);

}

}**else**{

cout<<"Массив не заполнен";

}

wait\_key();

}

**int** main(**int** argc, **const** **char** \* argv[]) {

**int** array[N];

**int** k,n;

**do**

{

cout<<"1. Заповнити масив \n";

cout<<"2. Вивести масив на екран \n";

cout<<"3. Завдання 1 \n";

cout<<"4. завдання 2 \n";

cout<<"5. Вихід \n";

cin>>k;

**if** (cin.fail() || k!=**int**(k)) {

cout << "Error" <<endl;

cin.clear();

cin.ignore(256,'\n');

//cin >> k;

// k=5;

}

**switch** (k)

{

**case** 1:

fill\_array(array,n);

**break**;

**case** 2:

show\_array(array,n);

wait\_key();

**break**;

**case** 3:

Task1(array,n);

**break**; // функція завдання 1

**case** 4:

Task2(array,n);

**break**; // функція завдання 2…

**default**:

**break**;

}

}

**while** (k!=5); //вихід

}

4. Опис інтерфейсу (керівництво користувача).

При запуску програми ви бачите меню з відповідними функціями .

1. Заповнити масив ";

2. Вивести масив на екран ";

3. Завдання 1 ";

4. завдання 2 ";

5. Вихід ";

П

Якщо ви оберете №1 вам потрібно буде вказати ціле значення довжину масиву 0<n<=100

Потім вказати min max – це діапазон чисел які випадково будуть заповнювати масив

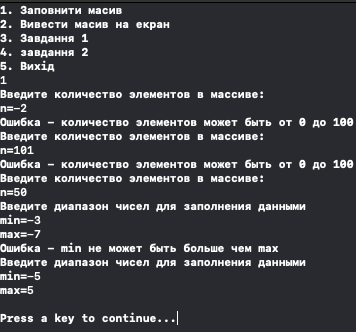
№2 показує заповнений масив, або повідомлення про те що він пустий

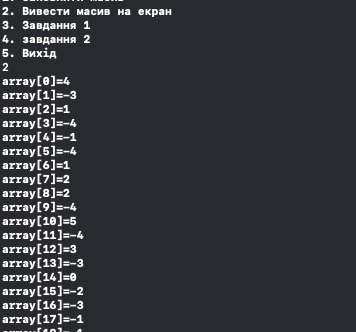
№3 Рахує суму елементів з парними індексами перед мінімальним елементом масиву

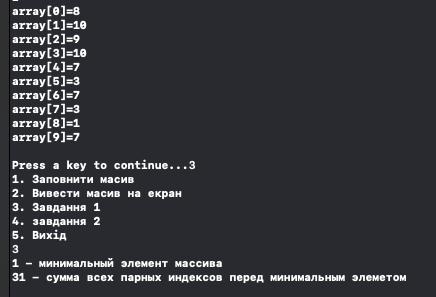
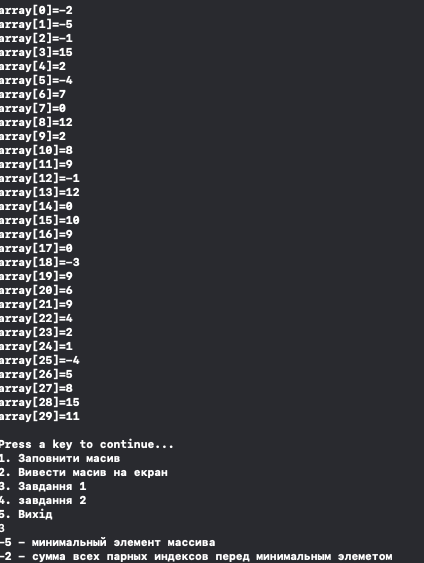
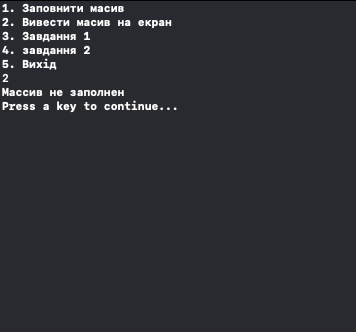
№4 додає після кожного елементу з парним значенням 0-вий елемент .

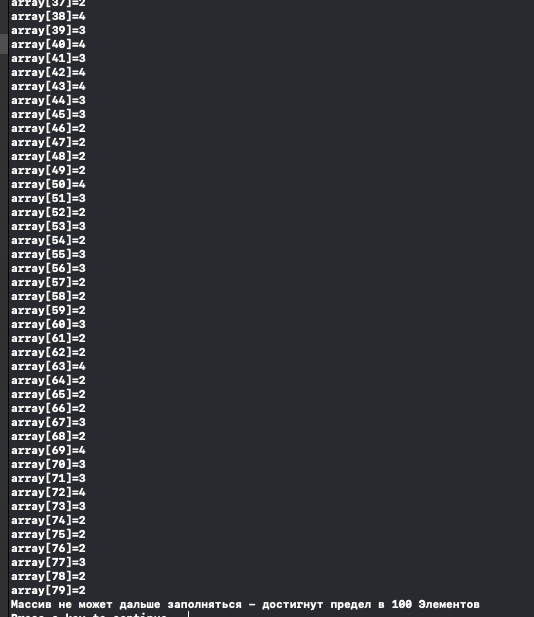
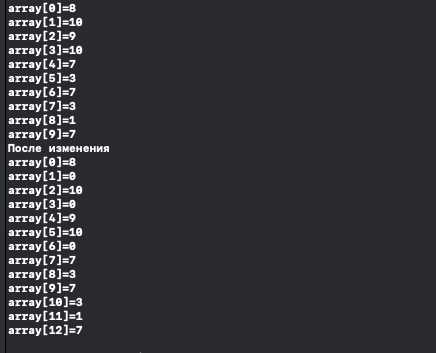
№5 завершує програму

5. Опис тестових прикладів та скриншоти їх реалізації.









6. Аналіз результатів. Висновки.

Для завдання 1 використовується алгоритм пошуку мінімального елементу.  
Для цього завдання можна було би не робити нову змінну, де зберігати значення мінімального елементу, а використовувати лише його індекс в масиві

В завданні 2 я зробив вкладений цикл, який здвигає масив на 1 значення.  
Хоча б можна було б уникнути квадратичної складності і використати лінійну,   
спочатку дізнатись кількість парних елементів, і зрозуміти чи вийдемо ми за межі чи ні.  
А потім в іншому циклі двигати його на 1 і присвоювати 0 після парних.

Перший алгоритм без виводу

T(n)=3+3(n-1)+n/2=3+3n-3+n/2=3n+n/2=3.5n;

O(n)=n

Другий алгоритм без виводу

Т(n)=n(3+n+1)=4n+n^2;

O(n)=n^2