Центральноукраїнський національний технічний унiверситет

Кафедра програмування комп’ютерних систем і мереж

Дисципліна: Базові методології та технології програмування

Звiт

з лабораторних робіт № 7 – 8

Тема: Програмування алгоритмів оюробки масивів даних та їх реалізація на мовах С та Паскаль

Мета роботи: Вивчити прийоми введення та виведення даних в одномірних та двомірних масивах. Засвоїти способи аналізу масивів даних та упорядкування даних у масивах

Виконав: ст. гр. КМ-20

Калиновський В.М.

Перевірила: асистент кафедри ПКСМ

Тесленко О.Є.

Варіант 3

Дата виконання: 07.11.2020

Дата оформлення: 08.11.2020

**Кропивницький 2020 р.**

**Завдання з 7 лабораторної:** Задано масив X {-12.4, 0, 12.3, -1, 0.01, 0, 123.3, -18.97, 0, 0.8, 10.01, -1.2, 11.2, 12.3, 13.6, 0, -5.2, 12.3, -111.3, 45.7, -6.7, 0, 2.1}; Скласти алгоритм та програму де передбачено наступне: введення масиву, поелементна обробка масива, виведення результату роботи програми – згідно третього варіанту моя програма повинна знаходити найбільший елемент і замінити його місцем з останнім.

**Завдання з 8 лабораторної:** Скласти алгоритм та програму де передбачено наступне: введення масиву, поелементна обробка масива, виведення результату роботи програми – згідно третього варіанту моя програма повинна підрахувати кількість нульових елементів у кожному стовпчику матриці .

**Робота.** На початку я вважав, що завдання буде складним, але після того, як я вник до завдання – я досить швидко розробив алгоритм програми. Аналогічно й до другого. Єдине, яку інформацію я шукав – як оголосити масив. Все інше я згадував сам. Колись давно я дивився відео, як написати програму яка сортує елементи у порядку зростання/спадання, та й недавно я бачив статтю, у якій було описано всі можливі алгоритми сортування даних. Я спробував згадати один з них. Якщо коротко описати – моя перша програма має порівнювати попередній елемент з наступнім і більший запам’ятовувати. Коли всі елементи масиву будуть перевірені – найбільше значеня має стати в кінці масиву, а останнє на місці найбільшого – таке собі часткове сортування. Після програма має вивести новий масив та число – яке вона виявила найбільшим. У другому завданні я представив матрицю, як набір масивів, де кожен масив є строками цієї матриці: 4 строки – 4 матриці. Стовпчики є елементами з однаковим номером у масиві, але знаходяться у різних масивах – перший стовпчик є набором чисел з 4 масивів, де кожне знаходиться на перших місцях але в різних масивах. А моя программа має брати ці самі значення й перевіряти їх. Моя програма бере всі перші елементи кожного масива й перевіряє – чи дорівнюють вони нулю. Скільки дорівнюють – стільки значення й записується до змінної, яка відповідає за зберігання кількості цих самих елементів. Потім бере другі елементи, в кінці треті. Так мають працювати мої програми. Тепер розберемося, які результати очікуються.

Перша програма у масиві X {-12.4, 0, 12.3, -1, 0.01, 0, 123.3, -18.97, 0, 0.8, 10.01, -1.2, 11.2, 12.3, 13.6, 0, -5.2, 12.3, -111.3, 45.7, -6.7, 0, 2.1} має віднайти число 123.3, так як воно є самим великим і поставити його на місце останнього 2.1, а це число поставити на місце 123.3. Новий масив має виглядати наступним чином: X {-12.4, 0, 12.3, -1, 0.01, 0, 2.1, -18.97, 0, 0.8, 10.01, -1.2, 11.2, 12.3, 13.6, 0, -5.2, 12.3, -111.3, 45.7, -6.7, 0, 123.3}.

Друга програма у матриці має у першому стовпичу {0, 0, 1, 2} порахувати кількість нульових елементів, потім у другому {-3, 0, 5, 6} і у третьому {0, 4, 0, 0}. Як ми самі бачимо: у першому програма має сказати: 2, у другому: 1, у третьому: 3.

Тепер, коли ми розібралися з потребою від програми, можна приступати до алгоритму. Алгоритмом першої програми є:

1. Задаю масив X {-12.4, 0, 12.3, -1, 0.01, 0, 123.3, -18.97, 0, 0.8, 10.01, -1.2, 11.2, 12.3, 13.6, 0, -5.2, 12.3, -111.3, 45.7, -6.7, 0, 2.1}
2. Задаю параметри i, k, n для циклів for
3. До змінної ZnachMass1 даю значення першого/попереднього елемента
4. До змінної ZnachMass2 буде надаватись другий/наступний елемент масива
5. Створюю перший цикл for, який буде перевіряти попереднє значення з наступним і у тому разі, коли наступне значення буде більше попереднього, воно записується у попереднє
6. Коли перевірка всіх елементів закінчилась, починається другий for, який призначений для заміни найбільшого й останнього елементів місцями: він перевіряє кожний елемент з найбільшим, і коли вони дорівнюють, програма розуміє, на якому місці знаходиться найбільше число й на його місце записує останнє
7. Після циклу програма поміщує на останнє місце найбільше число
8. Останнім циклом for програма виводить поелементно новий масив
9. В кінці програма друкує найбільше число, яке саме вона розмістила в кінці

Коди програми буде написано у таблицях 1 та 2, результати на рисунку 1. Друга програма написана наступним алгоритмом:

1. Задаються глобальні змінні які зберігають кількість нульових значень для кожного стовпчика
2. Задається параметр i
3. Задаються 4 масиви, які відображають строки матриці
4. Задається цикл for, який повторюється на кожному елементі строчки
5. Задаються умовні оператори if, які перевіряють кожний елемент з тим номером, залежно від того, на якому етапі знаходиться цикл for і рахують для кожного масива(строки матриці)
6. В кінці програма виводить результат

Коди програми буде написано у таблицях 3 та 4, результати на рисунку 2.

Таблиця1. Код першої програми на С

|  |
| --- |
| #include <stdio.h>  #include <Windows.h>  #define NumMass 23  int main() {  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  float MassivX[NumMass] = {-12.4, 0, 12.3, -1, 0.01, 0, 123.3, -18.97, 0, 0.8, 10.01, -1.2, 11.2, 12.3, 13.6, 0, -5.2, 12.3, -111.3, 45.7, -6.7, 0, 2.1};    int i;  int k;  int n;    float ZnachMass1 = MassivX[0];  float ZnachMass2;    for (i = 1; i < NumMass; i++) {  ZnachMass2 = MassivX[i];  if (ZnachMass1 < ZnachMass2) {  ZnachMass1 = ZnachMass2;  }  printf("Проверка: Первое значение: %.2lf; второе значение: %.2lf\n", ZnachMass1, ZnachMass2);  }  if (i == NumMass) {  for (k = 0; k < NumMass; k++) {  ZnachMass2 = MassivX[k];  if (ZnachMass2 == ZnachMass1) {  MassivX[k] = MassivX[22];  }  }  }  MassivX[22] = ZnachMass1;    printf("\nМассив: ");  for (n = 0; n < NumMass; n++) {  printf(" %.2lf,", MassivX[n]);  }  printf("\n\nПоследнее число - самое большое: %.2lf.\n", ZnachMass1);  system("pause");  } |

Таблиця 2. Код першої програми на Паскалі

|  |
| --- |
| **program** Kalinka\_ZadLab7;  MassivX:**array**[0..22] **of** real=(-12.4, 0, 12.3, -1, 0.01, 0, 123.3, -18.97, 0, 0.8, 10.01, -1.2, 11.2, 12.3, 13.6, 0, -5.2, 12.3, -111.3, 45.7, -6.7, 0, 2.1);  **var** ZnachMass1,ZnachMass2:real; i,k,n:integer;  **begin**  ZnachMass1:=MassivX[0];  **for** i=0 **to** 22 **do**  **begin**  ZnachMass2:=MassivX[i];  **if** ZnachMass1<ZnachMass2 **then** ZnachMass1=ZnachMass2;  writeln('Проверка: Первое значение: ', ZnachMass1 ,'; второе значение: ', ZnachMass2);  **end**;  **if** i=22 **then for** k=0 **to** 22 **do**  **begin**  ZnachMass2:=MassivX[k];  **if** ZnachMass2=ZnachMass1 **then** MassivX[k] = MassivX[23];  **end**;  MassivX[22]=ZnachMass1;  writeln;writeln('Массив: ');  **for** n=0 **to** 22 **do**  **begin**  writeln(MassivX[n], ' ');  **end**;  writeln;wriyeln;writeln('Последнее число - самое большое: ', ZnachMass1);  readkey;  **end**. |

Таблиця 3. Код другої програми на С

|  |
| --- |
| #include <stdio.h>  #include <Windows.h>  #define NumMass 3  int N1 = 0, N2 = 0, N3 = 0;  void NN(int i) {  if (i == 0) {  N1++;  }  else if (i == 1) {  N2++;  }  else {  N3++;  }  }  int main() {  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  int i = 0;    int Matrix1[NumMass] = { 0, -3, 0 };  int Matrix2[NumMass] = { 0, 0, 4 };  int Matrix3[NumMass] = { 1, 5, 0 };  int Matrix4[NumMass] = { 2, 6, 0 };  for (i; i < NumMass; i++) {  if (Matrix1[i] == 0) {  NN(i);  }  if (Matrix2[i] == 0) {  NN(i);  }  if (Matrix3[i] == 0) {  NN(i);  }  if (Matrix4[i] == 0) {  NN(i);  }  }  printf("Количество нолевых элементов в первом стобце: %i, во втором: %i, в третьем: %i.\n", N1, N2, N3);  system("pause");  } |

Таблиця 4. Код другої програми на Паскалі

|  |
| --- |
| **program** Kalinka\_ZadLab7;  Matrix1:**array**[0..2] **of** real=(0, -3, 0);  Matrix2:**array**[0..2] **of** real=(0, 0, 4);  Matrix3:**array**[0..2] **of** real=(1, 5, 0);  Matrix4:**array**[0..2] **of** real=(2, 6, 0);  **var** N1,N2,N3,i:integer;  **function** NN(i:integer):integer;  **begin**  **if** i=0 **then** N1:=N1+1;  **if** i=1 **then** N2:=N2+1;  **if** i=2 **then** N3:=N3+1;  **end**;  **begin**  **for** i=0 **to** 2 **do**  **begin**  **if** Matrix1[i]=0 **then** NN(i);  **if** Matrix2[i]=0 **then** NN(i);  **if** Matrix3[i]=0 **then** NN(i);  **if** Matrix4[i]=0 **then** NN(i);  **end**;  writeln('Количество нолевых элементов в первом столбце: ', N1, ', во втором: ', N2, ', в третьем: ', N3);  **end**. |

Рисунок 1. Результати першої програми на обох мовах

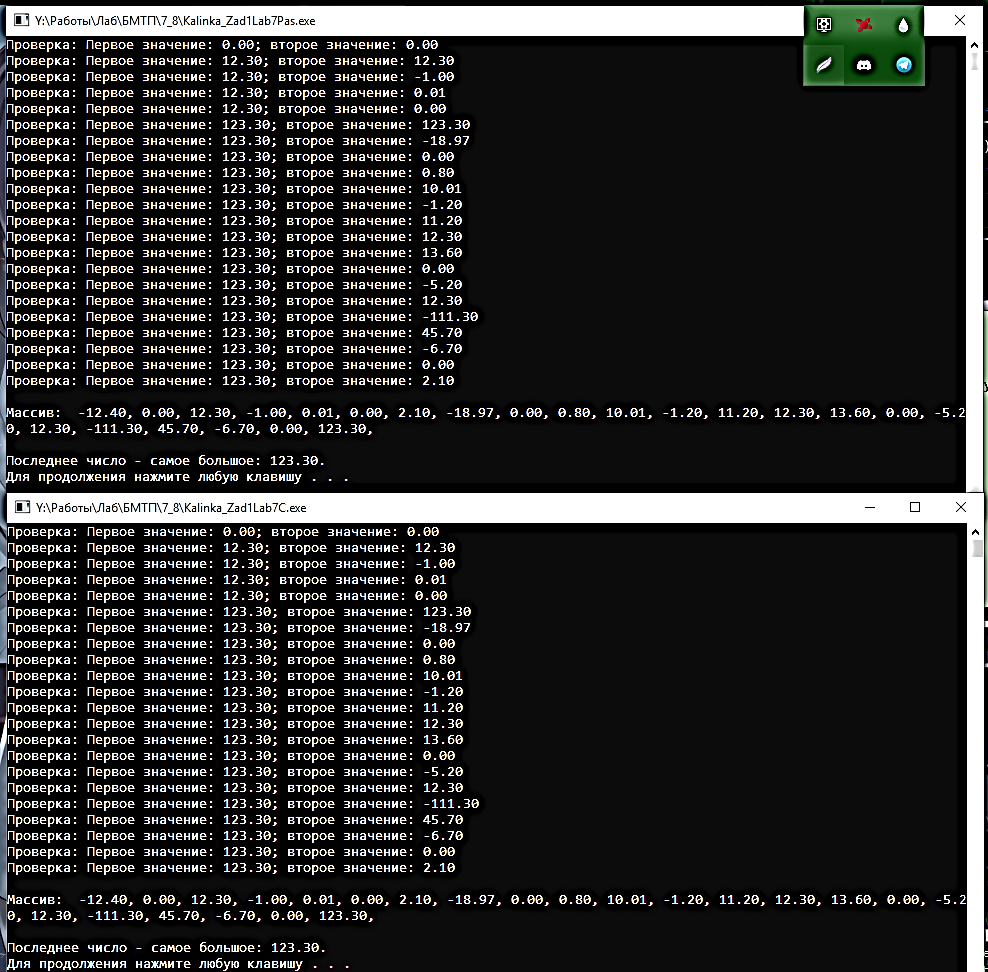
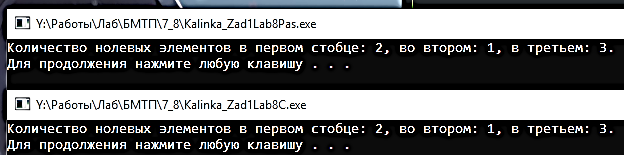


Рисунок 2. Результат другої прошрами на обох мовах



**Підсумуємо:** обидві програми виводять правильний результат – вони співпадають з тими результатами, які ми очікуємо від програм (у першому випадку – новий переписаний масив, у другому – правильно пораховану кількість нульових елементів), тому ми можемо сказати, що програми наші правильно написані й робочі, а їх основою можна користуватися для написання нових програм на цьому алгоритмі.

Відповіді на контрольні питання:

1. Що називається масивом?

Масив це набір із декількох елементів навідміну від змінної, яке зберігає лише одне значення елементу

1. Що таке ідентифікатор масиву?

Ідентифікатором масиву є кількість його елементів, яка зазначається місля його назви, чим він й відрізняється від змінної

1. Як ввести одномірний масив у діалозі?

Дописати ці значення після оголошення масива

1. Як вивести одновимірний масив у рядок та у стовпець?

У рядок – просто почергово брати кожний елемент і друкувати його, у стовпець – аналогічно попередньому за винятком того, що в кінці треба проставити оператор переносу курсора на новий рядок

1. Як вивести двовимірний масив у діалозі?

Кожна зі строк виміру виводиться на новій строчці

1. Як вивести двовимірний масив у формі таблиці?

Кожний елемент описувати по бокам символами, схожими на роздільні

1. Що таке розмірність масиву, як задається?

Я вважаю, що краще та легче кожний вимір масиву представляти окремим масивом й задавати його окремо

1. Як отримати доступ до елементів масиву, навести приклади?

Вказати конкретний адрес елемента, наприклад MassivX[4]

1. Як визначити добуток елементів кожного окремого стовпця матриці?

Вилучити всі перші елементи з мисивів та перемножити, потім другі, треті тощо, залежно від розмірів масив

1. Як визначити добуток елементів кожного окремого рядка матриці?

Кожний рядок представляється окремим масивом. З кожного масива вилучаються всі елементи й перемножаються. Потім другий, третій, і так далі в залежності від кількості строк головного масива

1. Які способи опису масивів вам відомі, навести приклади?

Зазначається тип масива, потім назва, розмірність й самі елементи

1. Які алгоритми сортування масиву вам відомі? Навести приклади фрагменту програми.

Почергова перебірка (спочатку шукає найбільший/найменший елемент масива, розміщує його спочатку/в кінці й потім повторюється не враховуючи використанні елементи й перші/останні номери до тих пір, поки не перебере всі елементи). У таблиці 5 наведено приклад.

Таблиця 5. Код сортувальної програми

|  |
| --- |
| #include <stdio.h>  #include <Windows.h>  #define NumMass 13  int main() {  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  int MyMassiv[NumMass] = { 3, 2, 9, 7, 14, 239, 814, -17, -1984, 0, 1, -2, 54 };  int i, k, n, m, o = NumMass, MassElement1 = MyMassiv[0], MassElement2;  for (k = 0; k < NumMass; k++) {  for (i = k; i < NumMass; i++) {  MassElement2 = MyMassiv[i];  if (MassElement1 < MassElement2) {  MassElement1 = MassElement2;  }  }  for (n = k; n < NumMass; n++) {  if (MassElement1 == MyMassiv[n]) {  MyMassiv[n] = MyMassiv[k];  MyMassiv[k] = MassElement1;  }  }  MassElement1 = -2000;  }  printf("\nРезультат: Новый отсортированный массив: ");  for (m = 0; m < NumMass; m++) {  printf("%i, ", MyMassiv[m]);  }  printf("\nСортровка завершена. ");  system("pause");  return 0;  } |

Рисунок 3. Результат сортувальної програми



Як ми бачимо: програма сортує масив { 3, 2, 9, 7, 14, 239, 814, -17, -1984, 0, 1, -2, 54 } і отримує відсортований на рисунку 3. До речі, цим алгоритмом можна користуватися для сортування любого масиву – достатньо всього переписати масив і у #define NumMass зазначити нову кількість елементів масиву

1. Як змінити сусідні елементи масиву місцями?

Вибрати ці елементи й перший переписати другим, а другий першим

1. Якій умові відповідають індекси діагональних елементів елементів матриці (головної та другорядної)?

Елемент кожного наступного рядка збільшується/зменшується на один в залежності від типу діагоналі: головна чи побічна відповідно

**Висновок:** Ми вивчили прийоми введення та виведення даних в одномірних та двомірних масивах, засвоїли способи аналізу масивів даних та упорядкування даних у масивах.

*Відразу відмічу, що всі блок-схеми до всіх задач написані від руки й будуть прикладені до звіту.*