

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 2

з дисципліни “ Основи програмування ”

тема “Робота з одномірними та багатомірними масивами. Рядки.”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) I курсу  групи КП- 61  Корунська Анна Михайлівна  (*прізвище, ім’я, по батькові*)  варіант № 17 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Штрафні бали:   |  |  | | --- | --- | | **Термін здачі** | **Оформлення звіту** | |  |  | | Нараховані бали:   |  |  |  | | --- | --- | --- | | **Корект. програм (2 бала)** | **Відп. на теор. питання (1 бал)** | **Відп. на прогр. питання (2 бала)** | |  |  |  | | Сумарний бал:   |  | | --- | |  | |

Київ 2016

**Мета роботи**

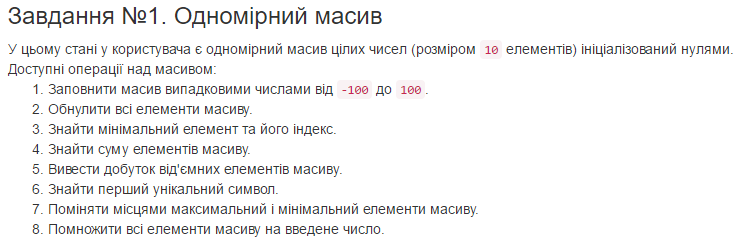
Навчитися працювати зі статичними масивами різних типів даних мови програмування С. Застосувати на практиці різні види циклічних конструкцій при роботі з одномірними та багатомірними масивами даних.

Вдосконалити вміння роботи з рядками у мові С.

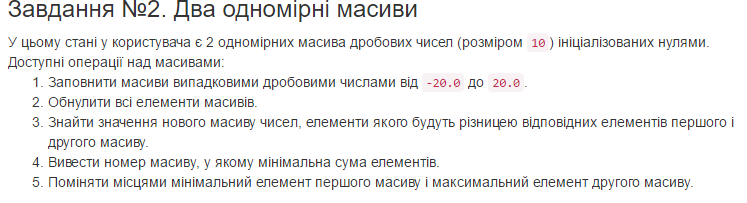
Навчитися оформлювати консольну програму для зручності роботи користувача.

**Постановка завдання**

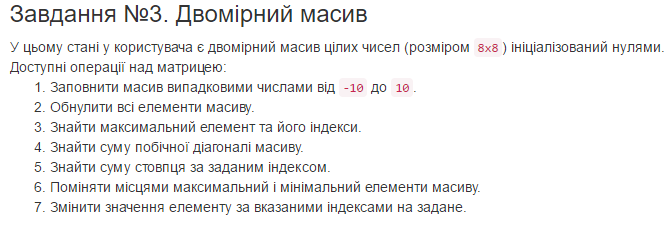


**Завдання 1.**

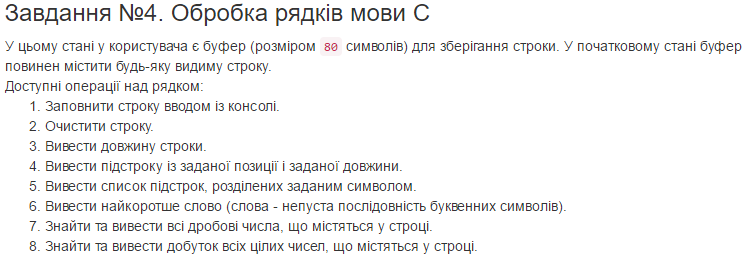
**Завдання 2**



**Завдання 3.**



**Завдання 4.**



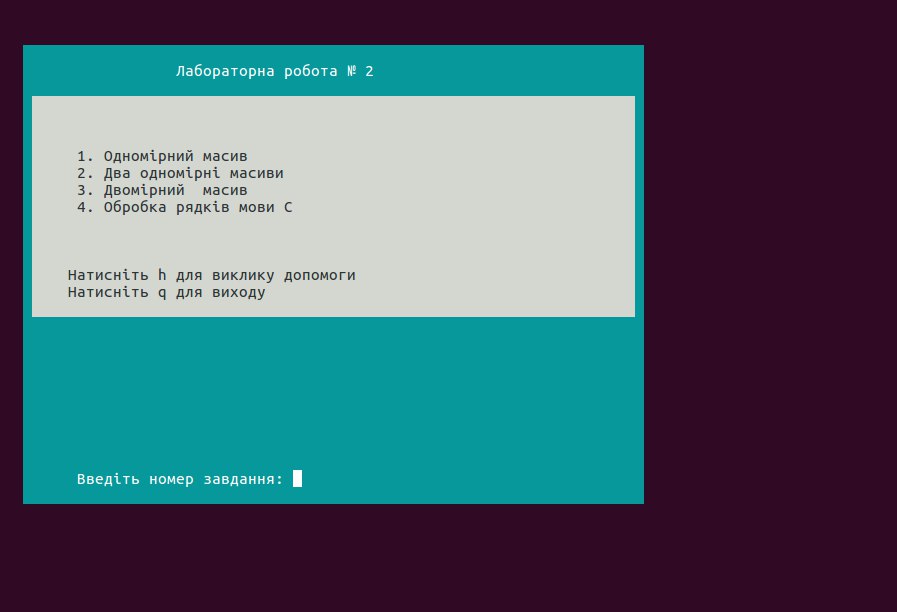
**Тексти коду програм**:

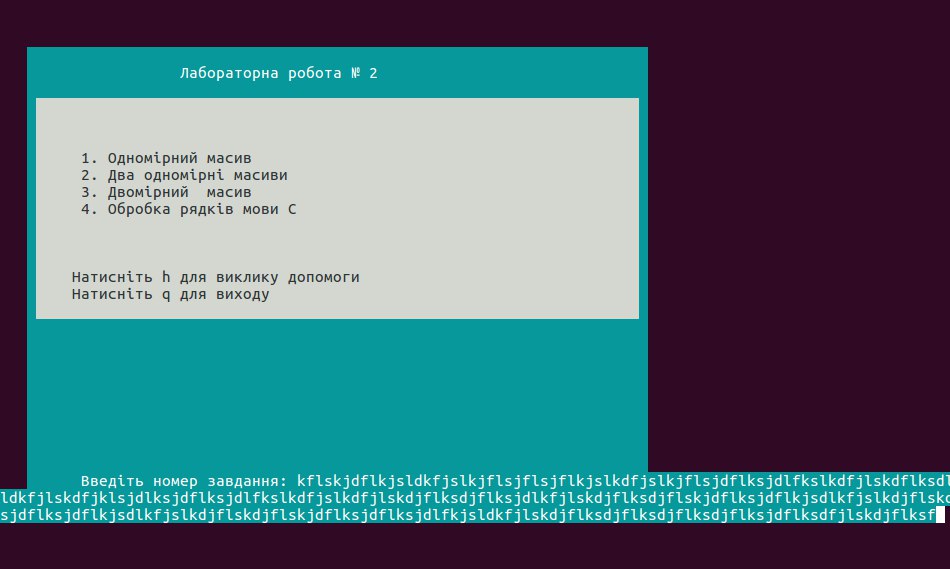
|  |
| --- |
| main.c |
| #include <stdio.h>  #include <progbase.h>  #include <pbconsole.h>  #include <ctype.h>  #include <stdlib.h>  #include <time.h>  #include <string.h>  struct menuBox {  int top;  int bottom;  int left;  int right;  int head;  int down;  };  void drawBox(struct menuBox);  void printMainMenu(struct menuBox box);  void mainMenu(void);  int taskOne(void);  int printGB(struct menuBox);  void clearScreen (struct menuBox box);  void copyArr(int arr[], int oldarr[]);  void printArr(int arr[], int oldarr[], struct menuBox);  void drawtOne(int arr[], int oldarr[], struct menuBox box);  int taskTwo(void);  void copyArr2 (double arr[], double oldarr[]);  void printArrs(double arr1[], double arr2[], double oldarr1[], double oldarr2[], struct menuBox box);  void drawtTwo(double arr1[], double arr2[], double oldarr1[], double oldarr2[], struct menuBox box);  void printHelp(struct menuBox box);  int taskThree(void);  void copyArr3 (int arr[8][8], int oldarr[8][8]);  void printArr3(int arr[8][8], int oldarr[8][8], struct menuBox box);  void drawtThree(int arr[8][8], int oldarr[8][8], struct menuBox box);  int taskFour (void);  void cleanStr(char str[], int size, int position);  void printStr(char str[80], struct menuBox box);  void drawtFour(char str[80], struct menuBox box);  int main(void){  mainMenu();  return 0;  }  void drawBox(struct menuBox box){  int x;  int y;  conReset();  for (x = 0; x <= 40; x++) {  for (y = 0; y <= 144; y++) {  conMove(x, y);  putchar(' ');  }  }  conSetAttr(BG\_CYAN);  for (x = box.top; x <= box.head; x++) {  for (y = box.left; y <= box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  conSetAttr(BG\_WHITE);  for (x = box.head; x < box.down; x++) {  for (y = box.left; y <= box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  conSetAttr(BG\_CYAN);  for (x = box.down; x <= box.bottom; x++) {  for (y = box.left; y <= box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  for (x = box.head; x < box.down; x++) {  conMove(x, box.right);  putchar(' ');  conMove(x, box.left);  putchar(' ');  }  conReset();  puts("");  }  int printGB(struct menuBox box){  int x;  int y;  conReset();  for (x = 0; x <= 40; x++) {  for (y = 0; y <= 140; y++) {  conMove(x, y);  putchar(' ');  }  }  conSetAttr(BG\_CYAN);  for (x = box.top; x <= box.bottom; x++) {  for (y = box.left; y <= box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  conMove((box.top + box.bottom)/2, box.left + 20);  printf("That's all)");  conReset();  conMove(box.bottom + 2, 0);  return 0;  }  void printHelp(struct menuBox box){  int x;  int y;  conSetAttr(BG\_BLACK);  for (x = box.down + 1; x < box.bottom - 2; x++) {  for (y = box.left + 1; y < box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  conMove((box.down + box.bottom)/2 - 3, box.left + 2);  printf("Для навігації використовуйте клавіші: \n\t m (повернення з меню завдань \  до головного меню), \n\t q (вихід із головного меню).");  conMove((box.down + box.bottom)/2, box.left + 2);  printf("Для вибору завдання введіть відповідну цифру, якою воно позначено.");  conMove((box.down + box.bottom)/2 + 2, box.left + 2);  printf("Приємного користування)");  conHideCursor();  conMove(box.bottom -1, box.left + 24);  conShowCursor();  conSetAttr(BG\_CYAN);  }  void clearScreen (struct menuBox box){  int x;  int y;  conReset();  for(x = box.down; x < box.bottom + 2; x++){  for(y = 0; y < box.left; y++){  conMove(x, y);  putchar(' ');  }  for(y = box.right + 1; y < 144; y++){  conMove(x, y);  putchar(' ');  }  }  for (x = box.bottom + 1; x <= box.bottom + 7; x++) {  for (y = 0; y <= 144; y++) {  conMove(x, y);  putchar(' ');  }  }  conSetAttr(BG\_CYAN);  }  void printMainMenu(struct menuBox box) {  drawBox(box);  conSetAttr(BG\_CYAN);  conMove((box.top + box.head)/2, box.left + 17);  printf("Лабораторна робота № 2");  conSetAttr(BG\_WHITE);  conSetAttr(FG\_BLACK);  conMove(box.head + 3, box.left + 3);  printf("\t 1. Одномірний масив \n\t 2. Два одномірні масиви \n\t 3. Двомірний\  масив \n\t 4. Обробка рядків мови С");  conMove(box.down - 3, 1);  printf("\tНатисніть h для виклику допомоги \n\tНатисніть q для виходу");  conReset();  conSetAttr(BG\_CYAN);  conMove(box.bottom -1, box.left);  printf("\t Введіть номер завдання: ");  }  void mainMenu(void) {  struct menuBox box = {4, 30, 4, 72, 7, 20};  int exit = 1;  drawBox(box);  printMainMenu(box);  while(exit) {  char ch;  ch = conGetChar();  printf("%c", ch);  if(isalpha(ch)){  if('q' == ch){  exit = printGB(box);  break;  }  if('h' == ch){  printHelp(box);  conSetAttr(BG\_CYAN);  conMove(box.bottom -1, box.left);  printf("\t Введіть номер завдання: ");  }  } else if(isdigit(ch)){  if('1' == ch) {  taskOne();  printMainMenu(box);  }  if('2' == ch) {  taskTwo();  printMainMenu(box);  }  if('3' == ch) {  taskThree();  printMainMenu(box);  }  if('4' == ch) {  taskFour();  printMainMenu(box);  }  } else {  drawBox(box);  printMainMenu(box);  conMove(box.bottom, box.left);  printf("\t Неправильно введений номер завдання");  conMove(box.bottom - 1, box.left);  printf("\t Введіть номер завдання: ");  }  }  }  int taskOne(void){  struct menuBox box = {4, 33, 4, 74, 7, 22};  int arr[10];  int oldarr[10];  int i;  srand(time(NULL));  for(i = 0; i < 10; i++){  arr[i] = 0;  oldarr[i] = 0;  }  drawBox(box);  conSetAttr(BG\_CYAN);  conMove((box.top + box.head)/2, box.left + 22);  printf("Завдання 1: одномірний масив");  conSetAttr(BG\_WHITE);  conSetAttr(FG\_BLACK);  conMove(box.head + 1, box.left + 3);  printf("\t1.Заповнити масив випадковими числами від -100 до 100.\n");  printf("\t2.Обнулити всі елементи масиву.\n");  printf("\t3.Знайти мінімальний елемент та його індекс.\n");  printf("\t4.Знайти суму елементів масиву.\n");  printf("\t5.Вивести добуток від'ємних елементів масиву.\n");  printf("\t6.Знайти перший унікальний символ.\n");  printf("\t7.Поміняти місцями максимальний і мінімальний елементи масиву.\n");  printf("\t8.Помножити всі елементи масиву на введене число.\n");  conMove(box.down - 3, 1);  printf("\tНатисніть h для виклику допомоги \n\tНатисніть m для повернення \  у головне меню");  drawtOne(arr, oldarr, box);  conSetAttr(BG\_CYAN);  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  while(1) {  char ch;  ch = conGetChar();  printf("%c", ch);  if(isalpha(ch)){  if('m' == ch){  return 0;  }  if('h' == ch) {  printHelp(box);  }  } else if(isdigit(ch)){  if('1' == ch) {  copyArr(arr, oldarr);  for(i = 0; i < 10; i++){  arr[i] = rand()%200 - 100;  }  drawtOne(arr, oldarr, box);  }  if('2' == ch) {  copyArr(arr, oldarr);  for(i = 0; i < 10; i++){  arr[i] = 0;  }  drawtOne(arr, oldarr, box);  }  if('3' == ch) {  int min = arr[1];  int posMin = 0;  copyArr(arr, oldarr);  for(i = 0; i < 10; i++){  if(arr[i] < min){  min = arr[i];  posMin = i;  }  }  drawtOne(arr, oldarr, box);  conMove(box.bottom - 4, box.left);  printf(" Індекс мінімального елемента: %i", posMin);  conMove(box.bottom - 3, box.left);  printf(" Значення мінімального елемента: %i", min);  conHideCursor();  conMove(box.bottom -1, box.left + 24);  conShowCursor();  }  if('4' == ch) {  int sum = 0;  copyArr(arr, oldarr);  for(i = 0; i < 10; i++){  sum += arr[i];  }  drawtOne(arr, oldarr, box);  conMove(box.bottom - 3, box.left);  conHideCursor();  printf(" Cума елементів масиву: %i", sum);  conMove(box.bottom -1, box.left + 24);  conShowCursor();  }  if('5' == ch) {  int prod = 1;  int flag = 0;  copyArr(arr, oldarr);  for(i = 0; i < 10; i++){  if(arr[i] < 0){  prod \*= arr[i];  flag = 1;  }  }  drawtOne(arr, oldarr, box);  conMove(box.bottom - 3, box.left);  conHideCursor();  if(flag == 1){  printf(" Добуток відʼємних елементів: %i", prod);  } else {  printf(" Добуток відʼємних елементів неможливо знайти");  }  conMove(box.bottom -1, box.left + 24);  conShowCursor();  }  if('6' == ch) {  int i = 0;  int j = 0;  int u;  int flag = 0;  for(i = 0; i < 10; i++){  u = arr[i];  for(j = i + 1; j < 10; j++){  if(arr[j] == arr[i]){  u = arr[i + 1];  } else {  flag = 1;  break;  }  }  if(flag) {  break;  }  }  drawtOne(arr, oldarr, box);  conMove(box.bottom - 3, box.left);  conHideCursor();  if(flag == 1){  printf(" Перший унікальний елемент: %i", u);  } else {  printf(" Унікальний елемент неможливо знайти");  }  conMove(box.bottom -1, box.left + 24);  conShowCursor();  }  if('7' == ch) {  int posMin = 0;  int posMax = 0;  int min = arr[0];  int max = arr[0];  int temp;  copyArr(arr, oldarr);  for(i = 1; i < 10; i++){  if(arr[i] < min){  min = arr[i];  posMin = i;  }  }  for(i = 1; i < 10; i++){  if(arr[i] > max){  max = arr[i];  posMax = i;  }  }  temp = arr[posMax];  arr[posMax] = arr[posMin];  arr[posMin] = temp;  drawtOne(arr, oldarr, box);  }  if('8' == ch) {  int val;  int i;  copyArr(arr, oldarr);  conMove(box.bottom -1, box.left);  printf(" Введіть значення, на яке домножаться елементи: ");  val = getInt();  for(i = 0; i < 10; i++){  arr[i] \*= val;  }  drawtOne(arr, oldarr, box);  }  if('0' == ch || '9' == ch) {  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  } else {  drawtOne(arr, oldarr, box);  conMove(box.bottom, box.left);  printf(" Неправильно введений номер операції");  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  }  }  void copyArr(int arr[], int oldarr[]) {  int i = 0;  for(; i < 10; i++){  oldarr [i] = arr [i];  }  }  void printArr(int arr[], int oldarr[], struct menuBox box){  int i;  conMove(box.down + 1, box.left);  conSetAttr(BG\_CYAN);  conSetAttr(FG\_WHITE);  for(i = 0; i < 10; i++){  if(arr[i] == oldarr[i]) {  printf(" %i", arr[i]);  } else {  conSetAttr(FG\_INTENSITY\_BLACK);  printf(" %i", arr[i]);  conSetAttr(FG\_WHITE);  }  }  }  void drawtOne(int arr[], int oldarr[], struct menuBox box){  int x;  int y;  clearScreen(box);  conSetAttr(BG\_CYAN);  for (x = box.down; x <= box.bottom; x++) {  for (y = box.left; y <= box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  printArr(arr, oldarr, box);  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  int taskTwo(void){  struct menuBox box = {4, 32, 4, 95, 7, 22};  double arr1[10];  double arr2[10];  double oldarr1[10];  double oldarr2[10];  int i;  srand(time(NULL));  for(i = 0; i < 10; i++){  arr1[i] = 0.0;  arr2[i] = 0.0;  oldarr1[i] = 0.0;  oldarr2[i] = 0.0;  }  drawBox(box);  conSetAttr(BG\_CYAN);  conMove((box.top + box.head)/2, box.left + 30);  printf("Завдання 2: два одномірні масиви");  conSetAttr(BG\_WHITE);  conSetAttr(FG\_BLACK);  conMove(box.head + 1, box.left + 3);  printf("\t1. Заповнити масиви випадковими дробовими числами від -20.0 до 20.0.\n");  printf("\t2. Обнулити всі елементи масивів.\n");  printf("\t3. Знайти значення нового масиву чисел, елементи якого будуть \  різницею відповідних \n\t елементів першого і другого масиву.\n");  printf("\t4. Вивести номер масиву, у якому мінімальна сума елементів.\n");  printf("\t5. Поміняти місцями мінімальний елемент першого масиву і \  максимальний елемент \n\t другого масиву.\n");  conMove(box.down - 3, 1);  printf("\tНатисніть h для виклику допомоги \n\tНатисніть m для повернення \  у головне меню");  printArrs(arr1, arr2, oldarr1, oldarr2, box);  conSetAttr(BG\_CYAN);  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  while(1) {  char ch;  ch = conGetChar();  printf("%c", ch);  if(isalpha(ch)){  if('m' == ch){  return 0;  }  if('h' == ch) {  printHelp(box);  }  } else if(isdigit(ch)){  if('1' == ch) {  copyArr2 (arr1, oldarr1);  copyArr2 (arr2, oldarr2);  for(i = 0; i < 10; i++){  arr1[i] = (rand()%400 - 200) / 10.0;  arr2[i] = (rand()%400 - 200) / 10.0;  }  drawtTwo(arr1, arr2, oldarr1, oldarr2, box);  }  if('2' == ch) {  copyArr2 (arr1, oldarr1);  copyArr2 (arr2, oldarr2);  for(i = 0; i < 10; i++){  arr1[i] = 0.0;  arr2[i] = 0.0;  }  drawtTwo(arr1, arr2, oldarr1, oldarr2, box);  }  if('3' == ch) {  double arr3[10];  copyArr2 (arr1, oldarr1);  copyArr2 (arr2, oldarr2);  for(i = 0; i < 10; i++){  arr3[i] = arr1[i] - arr2[i];  }  drawtTwo(arr1, arr2, oldarr1, oldarr2, box);  conMove(box.down + 3, box.left);  conSetAttr(BG\_CYAN);  conSetAttr(FG\_WHITE);  printf(" Масив 3: ");  for(i = 0; i < 10; i++){  printf(" %6.1f", arr3[i]);  }  conMove(box.bottom -1, box.left + 24);  }  if('4' == ch) {  double sum1 = 0;  double sum2 = 0;  int fin;  copyArr2 (arr1, oldarr1);  copyArr2 (arr2, oldarr2);  for(i = 0; i < 10; i++){  sum1 += arr1[i];  sum2 += arr2[i];  }  if(sum1 < sum2){  fin = 1;  } else if(sum2 > sum1){  fin = 2;  } else fin = 0;  if(fin != 0){  drawtTwo(arr1, arr2, oldarr1, oldarr2, box);  conMove(box.bottom - 5, box.left);  printf(" Сума елементів першого масиву: %.1f\tСума елементів другого \  масиву: %.1f", sum1, sum2);  conMove(box.bottom - 4, box.left);  printf(" Масив з мінімальною сумою елементів: масив %i", fin);  conMove(box.bottom -1, box.left + 24);  } else {  drawtTwo(arr1, arr2, oldarr1, oldarr2, box);  conMove(box.bottom - 4, box.left);  conHideCursor();  printf(" Cуми елементів масивів рівні");  conMove(box.bottom -1, box.left + 24);  conShowCursor();  }  }  if('5' == ch) {  double min1 = arr1[0];  int posMin = 0;  double max2 = arr2[0];  int posMax = 0;  double temp;  copyArr2 (arr1, oldarr1);  copyArr2 (arr2, oldarr2);  for(i = 0; i < 10; i++){  if(arr1[i] < min1){  min1 = arr1[i];  posMin = i;  }  if(arr2[i] > max2){  max2 = arr2[i];  posMax = i;  }  }  temp = arr2[posMax];  arr2[posMax] = arr1[posMin];  arr1[posMin] = temp;  drawtTwo(arr1, arr2, oldarr1, oldarr2, box);  }  if('6' == ch || '7' == ch || '8' == ch || '9' == ch || '0' == ch) {  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  } else {  drawtTwo(arr1, arr2, oldarr1, oldarr2, box);  conMove(box.bottom, box.left);  printf(" Неправильно введений номер операції");  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  }  }  void copyArr2 (double arr[], double oldarr[]) {  int i = 0;  for(; i < 10; i++){  oldarr [i] = arr [i];  }  }  void printArrs(double arr1[], double arr2[], double oldarr1[], double oldarr2[], struct menuBox box){  int i;  conSetAttr(BG\_CYAN);  conSetAttr(FG\_WHITE);  conMove(box.down + 1, box.left);  printf(" Масив 1: ");  for(i = 0; i < 10; i++){  if(arr1[i] == oldarr1[i]) {  printf(" %6.1f", arr1[i]);  } else {  conSetAttr(FG\_INTENSITY\_BLACK);  printf(" %6.1f", arr1[i]);  conSetAttr(FG\_WHITE);  }  }  conMove(box.down + 2, box.left);  printf(" Масив 2: ");  for(i = 0; i < 10; i++){  if(arr2[i] == oldarr2[i]) {  printf(" %6.1f", arr2[i]);  } else {  conSetAttr(FG\_INTENSITY\_BLACK);  printf(" %6.1f", arr2[i]);  conSetAttr(FG\_WHITE);  }  }  }  void drawtTwo(double arr1[], double arr2[], double oldarr1[], double oldarr2[], struct menuBox box) {  int x;  int y;  clearScreen(box);  conSetAttr(BG\_CYAN);  for (x = box.down; x <= box.bottom; x++) {  for (y = box.left; y <= box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  printArrs(arr1, arr2, oldarr1, oldarr2, box);  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  int taskThree(void){  struct menuBox box = {4, 37, 4, 80, 7, 22};  int arr[8][8];  int oldarr[8][8];  int i;  int j;  srand(time(NULL));  for(i = 0; i < 8; i++){  for(j = 0; j < 8; j++){  arr[i][j] = 0;  oldarr[i][j] = 0;  }  }  drawBox(box);  conSetAttr(BG\_CYAN);  conMove((box.top + box.head)/2, box.left + 25);  printf("Завдання 3: двомірний масив");  conSetAttr(BG\_WHITE);  conSetAttr(FG\_BLACK);  conMove(box.head + 1, box.left + 3);  printf("\t1. Заповнити масив випадковими числами від -10 до 10.\n");  printf("\t2. Обнулити всі елементи масиву.\n");  printf("\t3. Знайти максимальний елемент та його індекси.\n");  printf("\t4. Знайти суму побічної діагоналі масиву.\n");  printf("\t5. Знайти суму стовпця за заданим індексом.\n");  printf("\t6. Поміняти місцями максимальний і мінімальний елементи масиву.\n");  printf("\t7. Змінити значення елементу за вказаними індексами на задане.\n");  conMove(box.down - 3, 1);  printf("\tНатисніть h для виклику допомоги \n\tНатисніть m для повернення \  у головне меню");  printArr3(arr, oldarr, box);  conSetAttr(BG\_CYAN);  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  while(1) {  char ch;  ch = conGetChar();  printf("%c", ch);  if(isalpha(ch)){  if('m' == ch){  return 0;  }  if('h' == ch) {  printHelp(box);  }  } else if(isdigit(ch)){  if('1' == ch) {  copyArr3(arr, oldarr);  for(i = 0; i < 8; i++){  for(j = 0; j < 8; j++){  arr[i][j] = rand()%21 - 10;  }  }  drawtThree(arr, oldarr, box);  }  if('2' == ch) {  copyArr3(arr, oldarr);  for(i = 0; i < 8; i++){  for(j = 0; j < 8; j++){  arr[i][j] = 0;  }  }  drawtThree(arr, oldarr, box);  }  if('3' == ch) {  int max = arr[0][0];  int posi = 0;  int posj = 0;  copyArr3(arr, oldarr);  for(i = 0; i < 8; i++){  for(j = 0; j < 8; j++){  if(arr[i][j] > max){  max = arr[i][j];  posi = i;  posj = j;  }  }  }  drawtThree(arr, oldarr, box);  conMove(box.bottom - 4, box.left);  printf(" Значення максимального елемента: %i", max);  conMove(box.bottom - 3, box.left);  printf(" Позиція максимального елемента: [%i][%i]", posi, posj);  conMove(box.bottom -1, box.left + 24);  conShowCursor();  }  if('4' == ch) {  int sum = 0;  copyArr3(arr, oldarr);  for(i = 0; i < 8; i++){  for(j = 7; j != -1; j--){  if( i + j == 7){  sum += arr[i][j];  }  }  }  drawtThree(arr, oldarr, box);  conMove(box.bottom - 4, box.left);  printf(" Сума елементів побічної діагоналі: %i", sum);  conMove(box.bottom -1, box.left + 24);  conShowCursor();  }  if('5' == ch) {  int sum = 0;  int column = 0;  int i = 0;  int flag = 1;  copyArr3(arr, oldarr);  conMove(box.bottom, box.left);  printf(" (нумерація починається з нуля)");  conMove(box.bottom -1, box.left);  printf(" Введіть номер стовпця, суму якого бажаєте порахувати:");  column = getInt();  while(flag){  if(column > 0 && column < 7){  flag = 0;  } else{  conMove(box.bottom, box.left);  printf(" Введений номер стовпця є некоректним. Спробуйте ще раз.");  conMove(box.bottom -1, box.left);  printf(" Введіть номер стовпця, суму якого бажаєте порахувати: ");  column = getInt();  }  }  for(i = 0; i < 8; i++){  sum += arr[i][column];  }  drawtThree(arr, oldarr, box);  conMove(box.bottom - 4, box.left);  printf(" Сума елементів стовпця № %i: %i", column, sum);  conMove(box.bottom -1, box.left + 24);  conShowCursor();  }  if('6' == ch) {  int max = arr[0][0];  int posimax = 0;  int posjmax = 0;  int min = arr[0][0];  int posimin = 0;  int posjmin = 0;  int temp;  copyArr3(arr, oldarr);  for(i = 0; i < 8; i++){  for(j = 0; j < 8; j++){  if(arr[i][j] > max){  max = arr[i][j];  posimax = i;  posjmax = j;  }  if(arr[i][j] < min) {  min = arr[i][j];  posimin = i;  posjmin = j;  }  }  }  temp = arr[posimin][posjmin];  arr[posimin][posjmin] = arr[posimax][posjmax];  arr[posimax][posjmax] = temp;  drawtThree(arr, oldarr, box);  }  if('7' == ch) {  int i = 0;  int j = 0;  int flag = 1;  int val = 0;  copyArr3(arr, oldarr);  conMove(box.bottom - 1, box.left);  printf(" (нумерація починається з нуля)");  conMove(box.bottom - 2, box.left);  printf(" Введіть позицію елемента, значення якого бажаєте змінити: [ ] [ ]");  conMove(box.bottom - 2, box.left + 61);  conShowCursor();  i = getInt();  conMove(box.bottom - 2, box.left + 67);  conShowCursor();  j = getInt();  while(flag){  if(i >= 0 && i <= 7 && j >= 0 && j <= 7){  flag = 0;  } else{  conMove(box.bottom, box.left);  printf(" Введена позиція елемента є некоректною. Спробуйте ще раз.");  conMove(box.bottom - 1, box.left);  printf(" (нумерація починається з нуля)");  conMove(box.bottom - 2, box.left);  printf(" Введіть позицію елемента, значення якого бажаєте змінити: [ ] [ ]");  conMove(box.bottom - 2, box.left + 61);  conShowCursor();  i = getInt();  conMove(box.bottom - 2, box.left + 67);  conShowCursor();  j = getInt();  }  }  conMove(box.bottom - 1, box.left);  printf(" Введіть бажане значення елемента на заданій позиції: ");  val = getInt();  arr[i][j] = val;  drawtThree(arr, oldarr, box);  }  if('8' == ch || '9' == ch || '0' == ch) {  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  } else {  drawtThree(arr, oldarr, box);  conMove(box.bottom, box.left);  printf(" Неправильно введений номер операції");  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  }  }  void copyArr3 (int arr[8][8], int oldarr[8][8]) {  int i;  int j;  for(i = 0; i < 8; i++){  for(j = 0; j < 8; j++){  oldarr[i][j] = arr[i][j];  }  }  }  void printArr3(int arr[8][8], int oldarr[8][8], struct menuBox box){  int j;  int i;  conMove(box.down, 0);  conSetAttr(BG\_CYAN);  conSetAttr(FG\_WHITE);  printf("\n\t ");  for(i = 0; i < 8; i++){  for(j = 0; j < 8; j++){  if(arr[i][j] == oldarr[i][j]) {  printf(" %3i", arr[i][j]);  } else {  conSetAttr(FG\_INTENSITY\_BLACK);  printf(" %3i", arr[i][j]);  conSetAttr(FG\_WHITE);  }  }  printf("\n\t ");  }  }  void drawtThree(int arr [8][8], int oldarr [8][8], struct menuBox box) {  int x;  int y;  clearScreen(box);  conSetAttr(BG\_CYAN);  for (x = box.down; x <= box.bottom; x++) {  for (y = box.left; y <= box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  conMove(box.bottom -1, box.left);  printArr3(arr, oldarr, box);  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  int taskFour (void){  struct menuBox box = {4, 31, 4, 88, 7, 22};  char str [82] = {"Your new string"};  drawBox(box);  conSetAttr(BG\_CYAN);  conMove((box.top + box.head)/2, box.left + 25);  printf("Завдання 4: обробка рядків мови С");  conSetAttr(BG\_WHITE);  conSetAttr(FG\_BLACK);  conMove(box.head + 1, box.left + 3);  printf("\t1. Заповнити строку вводом із консолі.\n");  printf("\t2. Очистити строку.\n");  printf("\t3. Вивести довжину строки.\n");  printf("\t4. Вивести підстроку із заданої позиції і заданої довжини.\n");  printf("\t5. Вивести список підстрок, розділених заданим символом.\n");  printf("\t6. Вивести найкоротше слово.\n");  printf("\t7. Знайти та вивести всі дробові числа, що містяться у строці.\n");  printf("\t8. Знайти та вивести добуток всіх цілих чисел, що містяться у строці.");  conMove(box.down - 3, 1);  printf("\tНатисніть h для виклику допомоги \n\tНатисніть m для повернення \  у головне меню");  printStr(str, box);  conSetAttr(BG\_CYAN);  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  while(1) {  char ch;  ch = conGetChar();  printf("%c", ch);  if(isalpha(ch)){  if('m' == ch){  return 0;  }  if('h' == ch) {  printHelp(box);  }  } else if(isdigit(ch)){  if('1' == ch) {  drawtFour(str, box);  conMove(box.bottom - 6, box.left);  printf(" Введіть нову строку: \n");  conMove(box.bottom - 5, box.left + 1);  fgets(str, sizeof(str), stdin);  drawtFour(str, box);  }  if('2' == ch) {  int position = 0;  cleanStr(str, sizeof(str), position);  drawtFour(str, box);  }  if('3' == ch) {  int i = 0;  int l = 0;  for(i = 0; i < sizeof(str); i++) {  if(i == 0 && str[i] == '\0'){  l = 0;  break;  }  if(str[i] == '\0'){  l = i - 1;  break;  }  }  drawtFour(str, box);  conMove(box.bottom - 3, box.left);  printf(" Довжина поточної строки: %i ", l);  conMove(box.bottom -1, box.left + 24);  }  if('4' == ch) {  int position;  int len;  char pstr [81];  int i = 0;  conMove(box.bottom - 4, box.left);  printf(" Введіть позицію, з якою виведеться нова підстрока:");  position = getInt();  conMove(box.bottom - 3, box.left);  printf(" Введіть довжину нової підстроки: ");  len = getInt();  len = len + position;  for(i = 0; position <= len; position++, i++) {  pstr [i] = str [position - 1];  if( position == len){  pstr[i] = '\0';  }  }  drawtFour(str, box);  conMove(box.bottom - 4, box.left);  printf(" Нова підстрока: ");  puts(pstr);  conMove(box.bottom -1, box.left + 24);  }  if('5' == ch) {  char ch;  char pstr [81];  int i;  int j;  int count = 0;  conMove(box.bottom - 6, box.left);  printf(" Введіть символ, яким розібʼєте строку на підстроки: ");  ch = conGetChar();  printf("%c", ch);  drawtFour(str, box);  conMove(box.bottom - 4, box.left + 1);  printf("Нові підстроки: ");  conMove(box.bottom - 3, box.left + 2);  for(i = 0, j = 0; i < sizeof(str); i++, j++){  pstr[j] = str[i];  count++;  if(str[i] == ch) {  for(j = 0; j < count - 1; j++){  printf("%c", pstr[j]);  }  printf(",");  j = -1;  count = 0;  }  }  conMove(box.bottom -1, box.left + 24);  }  if('6' == ch) {  int i;  int minl = strlen(str);  int wordl = 0;  int wordIndex = 0;  int flag = 0;  drawtFour(str, box);  conMove(box.bottom - 4, box.left + 1);  printf("Слово найменшої довжини: ");  for(i = 0; str[i] != '\0'; i++) {  if(isalpha(str[i])) {  wordl++;  flag = 1;  } else if(flag) {  if(minl > wordl) {  minl = wordl;  wordIndex = i - wordl;  }  wordl = 0;  flag = 0;  }  }  for(i = wordIndex; i < wordIndex + minl && str[i] != '\0'; i++) {  putchar(str[i]);  }  conMove(box.bottom -1, box.left + 24);  }  if('7' == ch) {  int length = strlen(str);  int textIndex = 0;  int flag = 0;  int i;  int count;  drawtFour(str, box);  conMove(box.bottom - 4, box.left + 2);  for (textIndex = 0; textIndex < length; textIndex++) {  char ch = str[textIndex];  char nextCh = str[textIndex + 1];  if (isdigit(ch) || ('-' == ch && isdigit(nextCh))) {  double number = atof(str + textIndex);  printf("%f ", number);  for(count = 0, i = textIndex; isdigit(str[i]) || str[i] == '.'; i++ ){  count++;  }  textIndex += count - 1;  flag = 1;  }  }  conMove(box.bottom - 5, box.left + 1);  if(flag){  printf(" Дробові числа, що містяться в строці:");  } else {  printf(" Дана строка не містить дробових чисел");  }  conMove(box.bottom -1, box.left + 24);  }  if('8' == ch) {  int length = strlen(str);  int textIndex = 0;  int number;  int prod = 1;  int count;  int flag = 0;  for (textIndex = 0; textIndex < length; textIndex++) {  char ch = str[textIndex];  char nextCh = str[textIndex + 1];  if (isdigit(ch) || ('-' == ch && isdigit(nextCh))) {  number = atoi(str + textIndex);  count = printf("%d", number);  prod \*= number;  flag = 1;  textIndex += count - 1;  }  }  drawtFour(str, box);  conMove(box.bottom - 4, box.left + 1);  if(flag){  printf("Добуток цілих чисел у строці дорівнює: %i", prod);  } else {  printf(" Дана строка не містить цілих чисел");  }  conMove(box.bottom - 3, box.left + 2);  conMove(box.bottom -1, box.left + 24);  }  if('9' == ch || '0' == ch) {  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  } else {  drawtFour(str, box);  conMove(box.bottom, box.left);  printf(" Неправильно введений номер операції");  conMove(box.bottom -1, box.left);  printf(" Введіть номер операції: ");  }  }  }  void cleanStr(char str[], int size, int position) {  if(size > position){  str[position] = '\0';  } else {  str[size - 1] = '\0';  }  }  void printStr(char str[82], struct menuBox box){  int y;  conSetAttr(BG\_BLACK);  for (y = box.left + 2; y <= box.right - 2; y++) {  conMove(box.down + 1, y);  putchar(' ');  }  conMove(box.down + 1, box.left + 2);  conSetAttr(FG\_WHITE);  str[81] = '\0';  puts(str);  conSetAttr(BG\_CYAN);  }  void drawtFour(char str[82], struct menuBox box) {  int x;  int y;  clearScreen(box);  conSetAttr(BG\_CYAN);  for (x = box.down; x <= box.bottom; x++) {  for (y = box.left; y <= box.right; y++) {  conMove(x, y);  putchar(' ');  }  }  conMove(box.bottom - 1, box.left );  printStr(str, box);  conSetAttr(BG\_CYAN);  conMove(box.bottom - 1, box.left);  printf(" Введіть номер операції: ");  } |

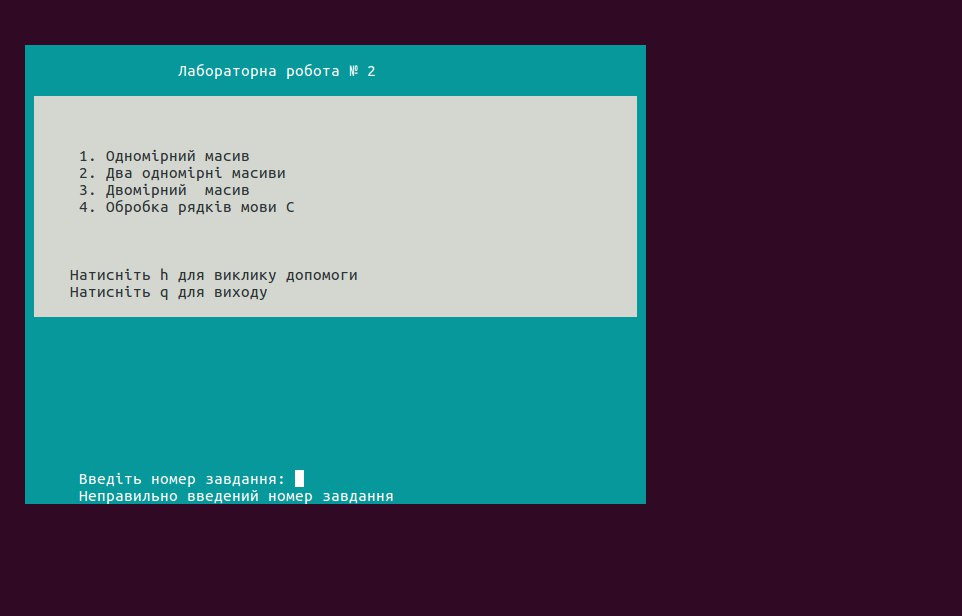
**Приклади результатів**

**Загальні риси роботи.**

При виклику програми відкривається головне меню.

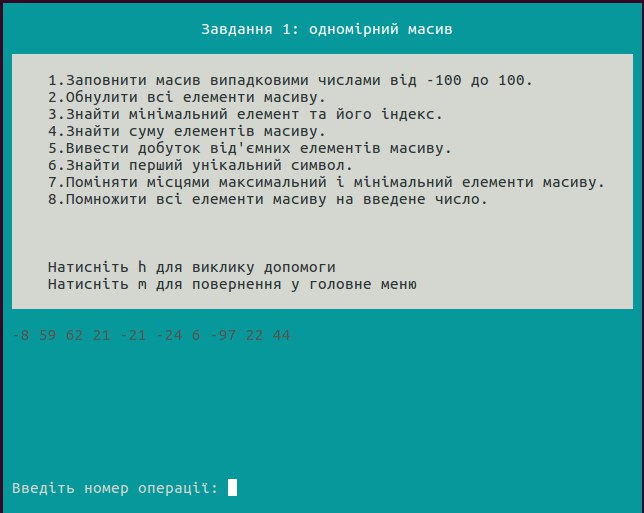


У випадку вводу неправильного номеру завдання або кодової клавіші вікно очиститься, а користувач побачить повідомлення про помилку: 

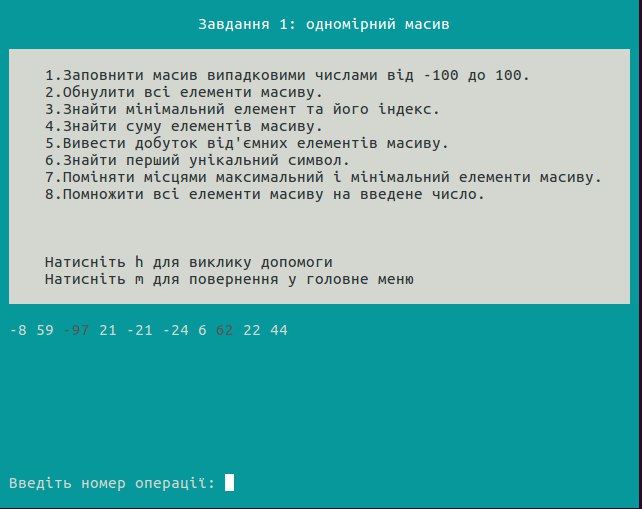


**Завдання 1.**

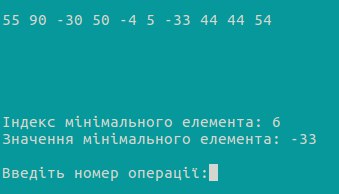
При переході до завдання 2 користувач має 8 дій с одномірним масивом на вибір:



Результат зміни місцями максимального і мінімального елементів:

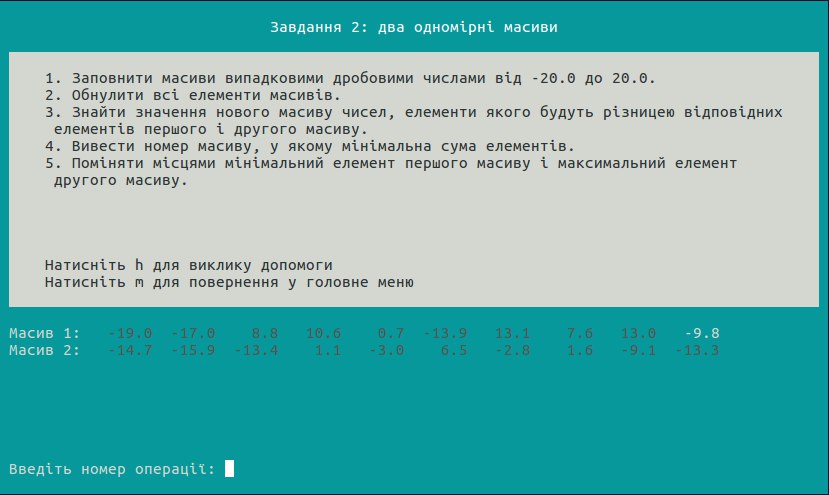


Приклад виводу результатів пошуку:

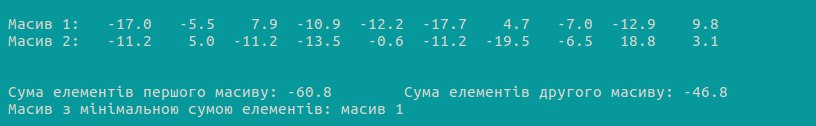


В головне меню користувач може повернутися, натистувши клавішу m, а з головного меню перейти у будь-яке інше меню завдання або завершити роботу програми.

**Завдання 2.**

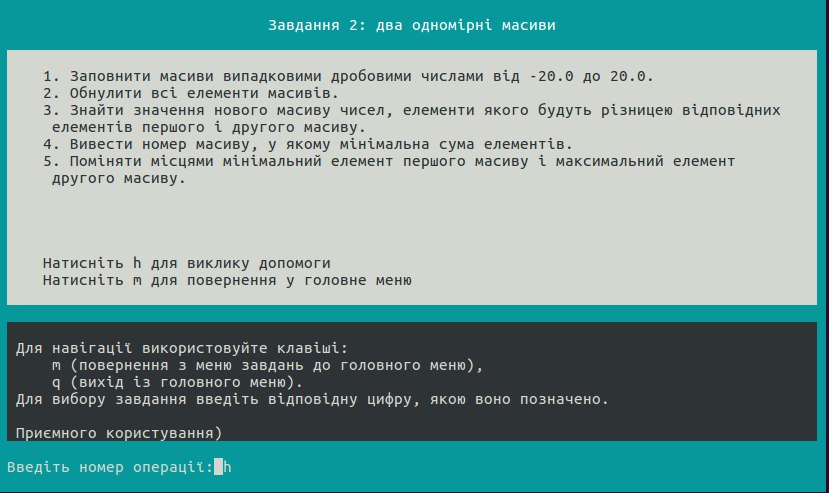
При переході у завдання два користувач має можливість працювати з двома одномірними масивами.

Приклад результату пошуку масиву з мінімальною сумою елементів:

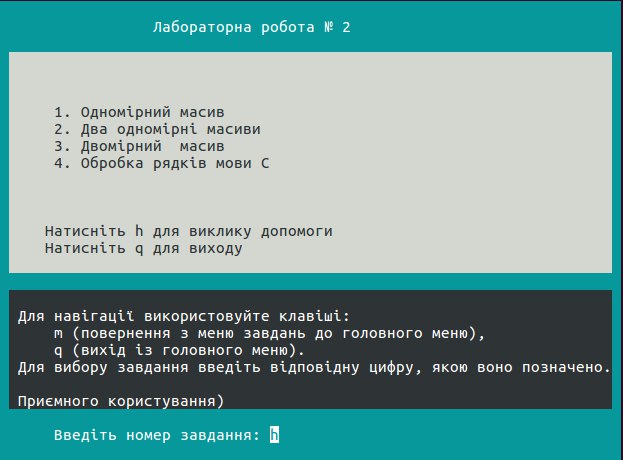


У випадку, коли сума елементів однакова, користувач отримає про це повідомлення:

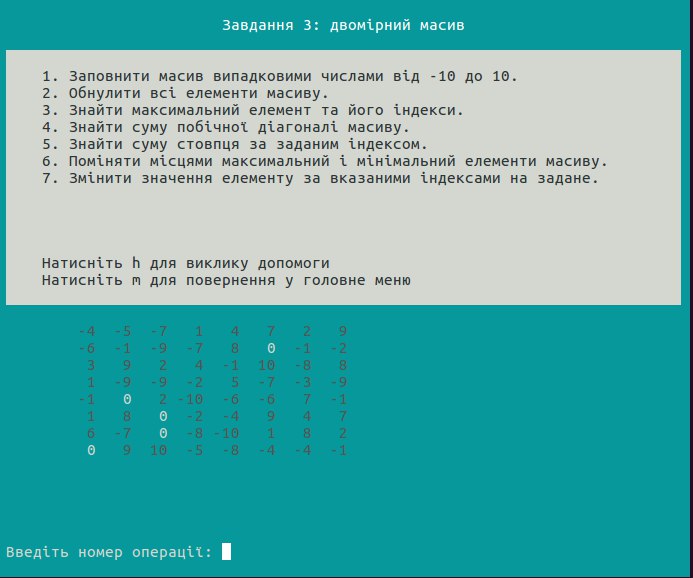


Як і у будь-якому стані програми, є можливість виклику вікна допомоги: 

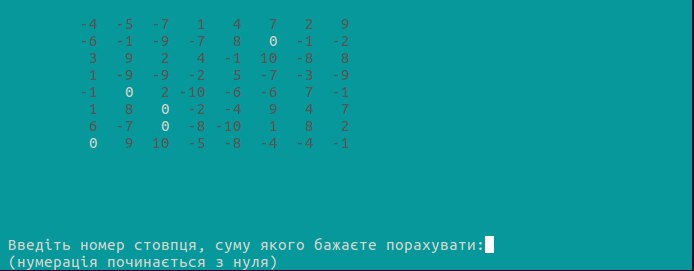
Приклад виклику вікна допомоги із головного меню:

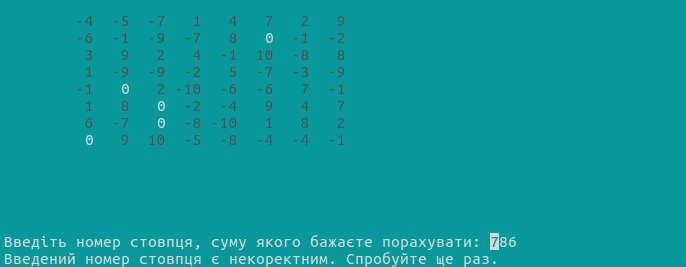


**Завдання 3.**

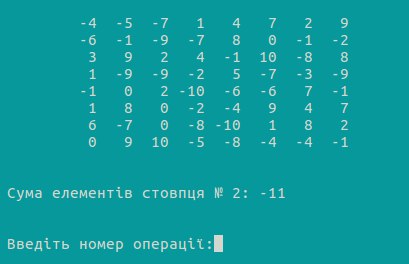
У завданні три користувач працює із квадратною матрицею, розміром 8\*8:

У випадку, коли користувач вводитиме недопустимі значення, виводитимуться відповідні повідомлення:



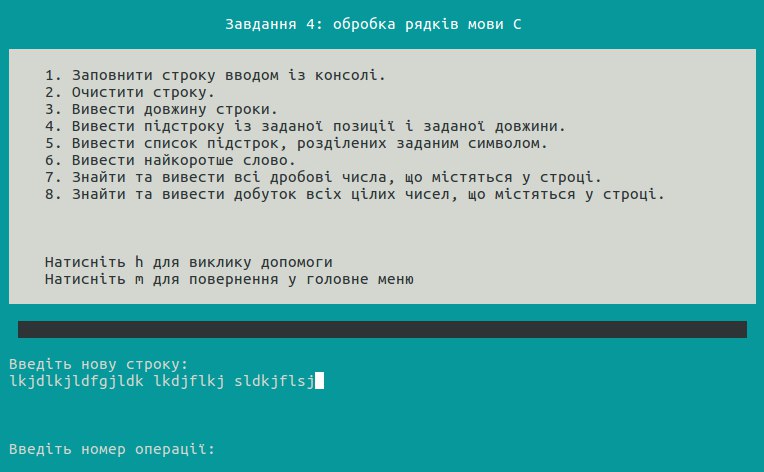


Приклад обрахунку суми елементів певного стовпця, обраного користувачем:



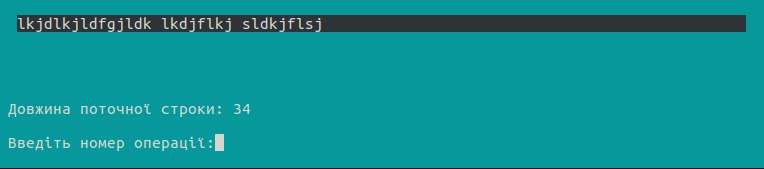
**Завдання 4.**

У завданні 4 користувач може вводити довільну строку розміром до 80 символів і виконувати з нею певні дії.

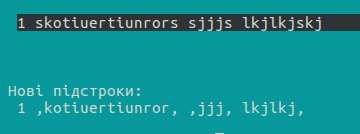




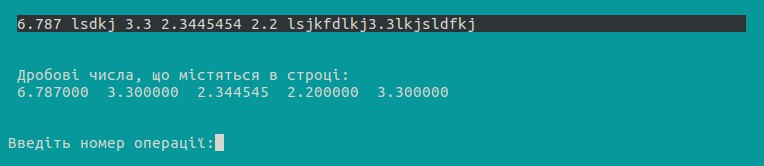
Приклад виводу результатів роботи: вивід довжини введеної строки:

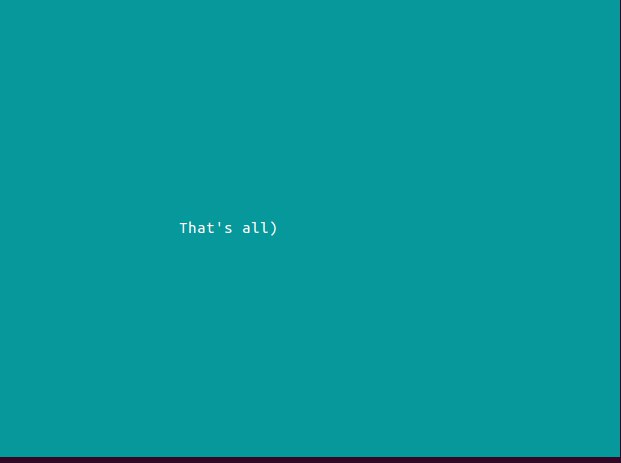


Приклад виводу підстрок основної строки, на які розбиває її символ:



Приклад знаходження та виводу дробових чисел у строці:



При завершенні роботи програми (перебуваючи в головному меню, користувач натискає q), виводиться прощальне вікно:

**Додаткове завдання.**

Також була реалізована підсвітка елементів, що змінюються.

Приклад 1: одномірний масив до та після операції "Поміняти місцями мінімальний та максимальний елементи масиву".

https://pp.vk.me/c638822/v638822471/ed50/ofhEHsXahtg.jpg

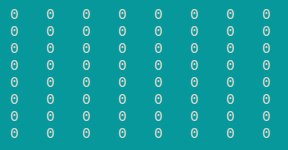
https://pp.vk.me/c638822/v638822471/ed57/3iMb82Zh1QM.jpg

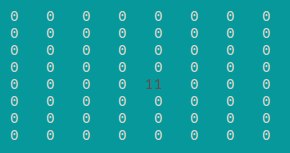
Приклад 2: два одномірні масиви до та після операції "Поміняти місцями мінімальний елемент першого масиву і максимальний елемент другого масиву."

https://pp.vk.me/c638822/v638822471/ed6f/67c7E8Z2DG4.jpg

https://pp.vk.me/c638822/v638822471/ed77/WCFsLiMEmrM.jpg

Приклад 3: квадратна матриця, заповнена нулями у першому стані, зі зміненим значенням елемента на позиції [4] [4] у другому.





**Висновки**

При виконанні даної лабораторної роботи була проведена робота зі статичними масивами різних видів за допомогою мови С. Також були покращені навички роботи зі строками.

Програма була оформлена для зручності роботи користувача. Завдяки циклічним конструкціям та конструкціям розгалуженна весь ввід користувача перевірявся на правильність, а також відбувалася обробка межових ситуацій і виключень.

Компіляція всього коду відбувалася за допомогою компілятора gcc.