

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  Щербина Вадим Олегович  (*прізвище, ім’я, по батькові*)  варіант №\_\_\_\_\_\_\_ |  | 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 2015

**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. Одномірний масив**

У цьому стані у користувача є одномірний масив цілих чисел (розміром 10 елементів) ініціалізований нулями.

Доступні операції над масивом:

1. Заповнити масив випадковими числами від -100 до 100.
2. Обнулити всі елементи масиву.
3. Знайти максимальний елемент та його індекс.
4. Знайти суму елементів масиву.
5. Вивести суму додатніх елементів масиву.
6. Знайти перший унікальний символ.
7. Поміняти місцями максимальний і мінімальний елементи масиву.
8. Збільшити всі елементи масиву на введене число.

### **Завдання №2. Два одномірні масиви**

У цьому стані у користувача є 2 одномірних масива дробових чисел (розміром 10) ініціалізованих нулями.

Доступні операції над масивами:

1. Заповнити масиви випадковими дробовими числами від -20.0 до 20.0.
2. Обнулити всі елементи масивів.
3. Знайти значення нового масиву чисел, елементи якого будуть добутком відповідних елементів першого і другого масиву.
4. Вивести номер масиву, у якому мінімальна сума елементів.
5. Поміняти місцями максимальний елемент першого масиву і мінімальний елемент другого масиву.

### **Завдання №3. Двомірний масив**

У цьому стані у користувача є двомірний масив цілих чисел (розміром 8х8) ініціалізований нулями.

Доступні операції над матрицею:

1. Заповнити масив випадковими числами від -10 до 10.
2. Обнулити всі елементи масиву.
3. Знайти мінімальний елемент та його індекси.
4. Знайти суму головної діагоналі масиву.
5. Знайти суму рядка за заданим індексом.
6. Поміняти місцями максимальний і мінімальний елементи масиву.
7. Змінити значення елементу за вказаними індексами на задане.

### **Завдання №4. Обробка рядків мови С**

У цьому стані у користувача є буфер (розміром 80 символів) для зберігання строки. У початковому стані буфер повинен містити будь-яку видиму строку.

Доступні операції над рядком:

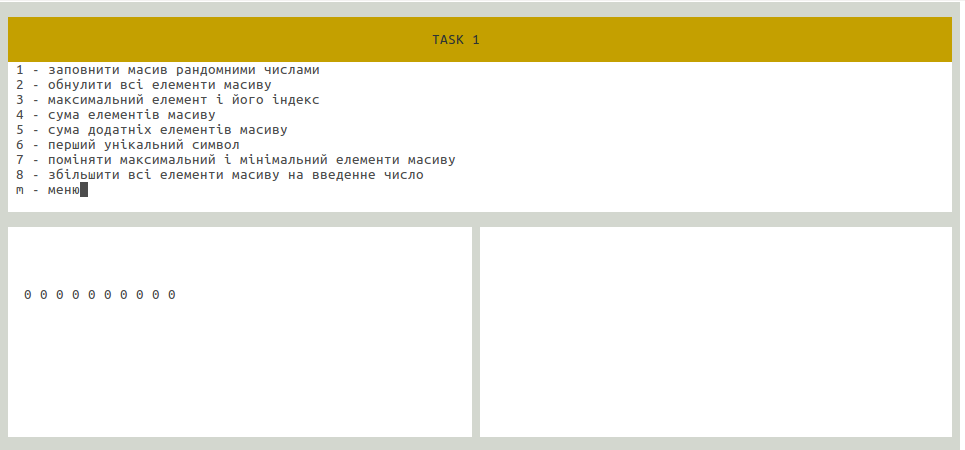
1. Заповнити строку вводом із консолі.
2. Очистити строку.
3. Вивести довжину строки.
4. Вивести підстроку із заданої позиції і заданої довжини.
5. Вивести список підстрок, розділених заданим символом.
6. Вивести найдовше слово (слова - непуста послідовність буквенних символів).
7. Знайти та вивести всі цілі числа, що містяться у строці.
8. Знайти та вивести суму всіх дробових чисел, що містяться у строці.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <time.h>  #include <pbconsole.h>  #include <math.h>  #include <ctype.h>  #include <string.h>  #include <progbase.h>  void line();  void clear();  void clear1();  void clear2();  void menu();  void masiv1(int A[10]);  void task2(double F[10],double D[10]);  void task3( int G[8][8]);  void clear3();  void line1();  void clear4();  void yellowline();  void task4();  void task41(char str[80]);  int main(void)  {  int A[10] = {0};  int random = 0;  double random1;  char numberstr[10];  int max = -110;  int sum = 0;  int sum2 = 0;  int unique = 0;  int j = 0;  int indexmax = 0;  int indexmin = 0;  int min = 0;  int temp = 0;  int b = 0;  int i = 0;  int k = 0;  double C[20] = {0.0};  double D[10]= {0.0};  double E[10]= {0.0};  double F[10]= {0.0};  double max1 = 0.0;  double min1 = 0.0;  int G[8][8]= {0};  int index1 = 0;  int index2 = 0;  int string = 0;  int indexm1 = 0;  int indexm2 = 0;  int size = 0;  char str[80] = {"Type string"};  int poselement=0;  int newstrlenth = 0;  int newstrsize = 0;  int count = 0;  int endword = 0;  int beginword = 0;  char sim;  double value = 0;  int flag = 0;  double sum3 = 0;  int flag1 =0;  conHideCursor();  menu();  while(1)  {  k = conGetChar();  conReset();  if(k == 27)  {  break;  }  switch(k)  {  case 49:  for (i = 0; i <= 9; i++)  {  A[i] = 0;  }  masiv1(A);  clear2();  yellowline();  conReset();  conSetAttr(BG\_YELLOW);  conSetAttr(FG\_BLACK);  conMove(3,55);  printf("TASK 1\n");  conReset();  conMove(5,3);  printf("1 - заповнити масив рандомними числами\n");  conMove(6,3);  printf("2 - обнулити всі елементи масиву\n");  conMove(7,3);  printf("3 - максимальний елемент і його індекс\n");  conMove(8,3);  printf("4 - сума елементів масиву\n");  conMove(9,3);  printf("5 - сума додатніх елементів масиву\n");  conMove(10,3);  printf("6 - перший унікальний символ");  conMove(11,3);  printf("7 - поміняти максимальний і мінімальний елементи масиву\n");  conMove(12,3);  printf("8 - збільшити всі елементи масиву на введенне число\n");  conMove(13,3);  printf("m - меню");  conReset();  while(1)  {  k = conGetChar();  conReset();  if(k == 109)  {  menu();  break;  }  switch(k)  {  case 49:  clear1();  conReset();  srand(time(0));  conMove(20, 2);  for (i = 0; i <= 9; i++)  {  random = rand() %200 -100;  A[i] = random;  printf(" %i", A[i]);  }  break;  case 50:  clear1();  conReset();  max = 0;  sum = 0;  sum2 = 0;  indexmax =0;  indexmin = 0;  temp = 0;  unique = 0;  conMove(20,2);  for(i = 0; i<=9; i++)  {  A[i] = 0;  printf(" %i", A[i]);  }  break;  case 51:  indexmax = 0;  max = 0;  clear();  masiv1(A);  conReset();  for(i = 0; i<=9; i++)  {  if(A[i] > max)  {  max = A[i];  indexmax = i;  }  }  conMove(18,62);  for (i = 0; i <= 9; i++)  {  if(A[i] == max)  {  conSetAttr(FG\_RED);  printf(" %i", A[i]);  conReset();  }  else  {  printf(" %i", A[i]);  }  }  conMove(20,62);  printf("максимальний елемент = %i\n\n", max);  conMove(21,62);  printf("індекс максимального елементa = %i\n",indexmax);  break;  case 52:  sum = 0;  clear();  masiv1(A);  conReset();  conMove(20,62);  for (i = 0; i <= 9; i++)  {  sum = sum + A[i];  }  printf("сума елементів масиву = %i\n", sum);  break;  case 53:  sum2 = 0;  clear();  masiv1(A);  conReset();  for(i = 0; i<=9; i++)  {  if(A[i]>0)  {  sum2 = sum2 + A[i];  }  }  if (sum2 > 0)  {  conMove(20,62);  printf("сума додатніх елементів = %i\n", sum2);  }  if (sum2 < 0)  {  conMove(20,62);  printf("Всі елементи відємні\n");  }  if(sum2 ==0)  {  conMove(20,62);  printf("Сума дорівнює 0");  }  conMove(18,62);  for (i = 0; i <= 9; i++)  {  if(A[i] >= 0)  {  conSetAttr(FG\_RED);  printf(" %i", A[i]);  conReset();  }  else  {  printf(" %i", A[i]);  }  }  break;  case 54:  clear();  clear1();  masiv1(A);  conReset();  unique= A[9];  conMove(20,62);  for(i =9; i >= 0; i--)  {  if(A[i] != unique)  {  unique = A[i];  }  }  printf("перший унікальний елемент = %i\n", unique);  conMove(20, 3);  for (i = 0; i <= 9; i++)  {  if(A[i] == unique)  {  conSetAttr(FG\_RED);  printf(" %i", A[i]);  conReset();  }  else  {  printf(" %i", A[i]);  }  }  break;  case 55:  clear();  masiv1(A);  conReset();  min = 110;  max = -110;  for(i = 0; i<=9; i++)  {  if(A[i] > max)  {  max = A[i];  indexmax = i;  }  }  for(i = 0; i<=9; i++)  {  if(A[i] < min)  {  min = A[i];  indexmin = i;  }  }  temp = A[indexmax];  A[indexmax] = A[indexmin];  A[indexmin] = temp;  conMove(20,62);  for(i = 0; i<=9; i++)  {  if(i == indexmax || i == indexmin)  {  conSetAttr(FG\_RED);  printf(" %i", A[i]);  conReset();  }  else  {  printf(" %i", A[i]);  }  }  break;  case 56:  clear();  masiv1(A);  conReset();  conMove(18,62);  printf("Введіть число на яке збільшаться елементи \n");  conMove(20,62);  fgets(numberstr, 10, stdin);  b = atoi(numberstr);  if(b != 0)  {  conMove(22,63);  for(i = 0; i<=9; i++)  {  A[i] = A[i] + b;  printf("%i ", A[i]);  }  }  else  {  conMove(22,63);  conSetAttr(BG\_RED);  printf("Введено не правильний символ");  conReset();  }  break;  }  }  break;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  case 50:  for (i = 0; i <= 9; i++)  {  D[i] = 0.0;  F[i] = 0.0;  }  clear();  clear1();  clear2();  task2(F,D);  yellowline();  conSetAttr(BG\_YELLOW);  conSetAttr(FG\_BLACK);  conMove(3,55);  printf("TASK 2\n");  conReset();  conMove(5,3);  printf("1 - заповнити масиви рандомними числами від -20 до 20");  conMove(6,3);  printf("2 - обнулити всі елементи масивів");  conMove(7,3);  printf("3 - новий масив елементами якого будуть добутки відповідних елементів двох масив");  conMove(8,3);  printf("4 - номер масива в якому мінімальна сума елементів");  conMove(9,3);  printf("5 - поміняти місцями максимальний елемент першого і мінімальний елемент другого масиву\n");  conMove(10,3);  printf("m - меню");  while(1)  {  k = conGetChar();  conReset();  if(k == 109)  {  menu();  break;  }  switch(k)  {  case 49:  clear1();  clear();  srand(time(0));  for (i = 0; i <= 19; i++)  {  random1 =(rand() % 400 - 200) / 10.0;  C[i] = random1;  }  conMove(20,3);  printf("F[10]= \n");  conMove(21,3);  for (i = 0, j = 0; i <= 9 && j <= 9; i++, j++)  {  F[j] = C[i];  printf(" %.1f", F[j]);  }  conMove(22,3);  printf("D[10]= \n");  conMove(23,3);  for (i = 10, j = 0; i <= 19 && j <= 9; i++, j++)  {  D[j] = C[i];  printf(" %.1f", D[j]);  }  break;  case 50:  clear1();  clear();  conMove(20,3);  printf("C[10]= \n");  conMove(21,3);  for (i = 0; i <= 9; i++)  {  F[i] = 0;  printf(" %1.f",F[i]);  }  conMove(22,3);  printf("D[10]= \n");  conMove(23,3);  for (i = 0; i <= 9; i++)  {  D[i] = 0;  printf(" %1.f",D[i]);  }  break;  case 51:  clear();  task2( F,D);  conMove(20,61);  printf("E[10]=\n");  conMove(21,61);  for(i = 0; i <=4; i ++)  {  E[i] = F[i] \* D[i];  printf(" %i) %.2f",i, E[i]);  }  conMove(22,61);  for(i = 5; i <=9; i ++)  {  E[i] = F[i] \* D[i];  printf(" %i) %.2f",i, E[i]);  }  break;  case 52:  clear();  task2( F,D);  sum = 0;  sum2 = 0;  for(i = 0; i <=9; i++ )  {  sum = sum + F[i];  sum2 = sum2 + D[i];  }  if(sum == sum2)  {  conMove(20,62);  printf("масиви мають однакову суму");  }  else  {  if (sum > sum2)  {  conMove(20,62);  printf("sum1 = %i", sum );  conSetAttr(FG\_RED);  conMove(21,62);  printf("sum2 = %i\n", sum2 );  conReset();  conMove(22,62);  printf("масив номер 2 має мінімальну суму елементів");  }  else  {  conMove(20,62);  conSetAttr(FG\_RED);  printf("sum1 = %i", sum );  conReset();  conMove(21,62);  printf("sum2 = %i", sum2 );  conMove(22,62);  printf("масив номер 1 має мінімальну суму елементів");  }  }  break;  case 53:  clear();  task2( F,D);  min1 = 30;  max1 = -30;  indexmax = 0;  indexmin = 0;  for(i = 0; i <= 9; i++)  {  if(F[i] > max1)  {  max1 = F[i];  indexmax = i;  }  }  for(i = 0; i <= 9; i++)  {  if(D[i] < min1)  {  min1 = D[i];  indexmin = i;  }  }  F[indexmax] = min1;  D[indexmin] = max1;  conMove(20,62);  printf("F[10]=\n");  conMove(21,62);  for(i = 0; i <= 9; i++)  {  if(i == indexmax)  {  conSetAttr(FG\_RED);  printf(" %.1f",F[i]);  conReset();  }  else  {  printf(" %.1f",F[i]);  }  }  conMove(22,62);  printf("D[10]=");  conMove(23,62);  for(i = 0; i <= 9; i++)  {  if(i == indexmin)  {  conSetAttr(FG\_RED);  printf(" %.1f", D[i]);  conReset();  }  else  {  printf(" %.1f", D[i]);  }  }  break;  }/\*2\*/  }/\*2\*/  break;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  case 51:  for (i = 0; i <= 7; i++)  {  for (j = 0; j <= 7; j++)  {  G[i][j]=0;  }  }  clear();  clear1();  clear2();  task3(G);  yellowline();  conSetAttr(BG\_YELLOW);  conSetAttr(FG\_BLACK);  conMove(3,55);  printf("TASK 3\n");  conReset();  conMove(5,3);  printf("1 - Заповнити масив випадковими числами від -10 до 10.");  conMove(6,3);  printf("2 - Обнулити всі елементи масиву.");  conMove(7,3);  printf("3 - Знайти мінімальний елемент та його індекси.");  conMove(8,3);  printf("4 - Знайти суму головної діагоналі масиву.");  conMove(9,3);  printf("5 - Знайти суму рядка за заданим індексом.");  conMove(10,3);  printf("6 - Поміняти місцями максимальний і мінімальний елементи масиву.");  conMove(11,3);  printf("7 - Змінити значення елементу за вказаними індексами на задане.");  conMove(12,3);  printf("m - Меню");  while(1)  {  k = conGetChar();  conReset();  if(k == 109)  {  menu();  break;  }  switch(k)  {  case 49:  clear1();  clear();  srand(time(0));  for (i = 0; i <= 7; i++)  {  conMove(18+i, 8);  for (j = 0; j <= 7; j++)  {  random = rand() %20 -10;  G[i][j] = random;  printf(" %\*i",3, G[i][j]);  }  }  break;  case 50:  clear1();  for (i = 0; i <= 7; i++)  {  conMove(18+i, 8);  for (j = 0; j <= 7; j++)  {  G[i][j] = 0;  printf(" %\*i",3, G[i][j]);  }  printf("\n");  }  break;  case 51:  clear();  min = 15;  for (i = 0; i <= 7; i++)  {  for (j = 0; j <= 7; j++)  {  if(G[i][j] < min)  {  min = G[i][j];  index1 = i;  index2 = j;  }  }  }  conMove(18, 64);  printf("min = %i", min);  conMove(20, 64);  printf("indexmin i = %i j = %i", index1, index2);  for (i = 0; i <= 7; i++)  {  conMove(18 + i,8);  for (j = 0; j <= 7; j++)  {  if(i == index1 && j == index2)  {  conSetAttr(FG\_RED);  printf(" %\*i",3, G[index1][index2]);  conReset();  }  else  {  printf(" %\*i",3, G[i][j]);  }  }  }  break;  case 52:  clear();  sum = 0;  for (i = 0, j = 0; i <= 7 && j <= 7; i++, j++)  {  sum = sum + G[i][j];  }  for (i = 0; i <= 7; i++)  {  conMove(18 + i,8);  for (j = 0; j <= 7; j++)  {  if(i == j)  {  conSetAttr(FG\_RED);  printf(" %\*i",3, G[i][j]);  conReset();  }  else  {  printf(" %\*i",3, G[i][j]);  }  }  }  conMove(18, 64);  printf("сума елементів по діагоналі дорівнює %i",sum);  break;  case 53:  clear();  sum = 0;  string = 0;  flag =0;  conMove(20, 64);  printf("введіть рядок =");  fgets(numberstr, 10, stdin);  for (k = 0; k < 10; k++)  {  if(isdigit(numberstr[k]))  {  string = atoi(numberstr);  flag = 1;  if(string < 8)  {  for( i = string, j = 0; j <= 7; j++)  {  sum = sum + G[i][j];  }  conMove(22, 64);  printf("сума елементів рядка = %i", sum);  }  }  }  for (i = 0; i <= 7; i++)  {  conMove(18 + i,8);  for (j = 0; j <= 7; j++)  {  if(i == string)  {  conSetAttr(FG\_RED);  printf(" %\*i",3, G[i][j]);  conReset();  }  else  {  printf(" %\*i",3, G[i][j]);  }  }  }  if(flag == 0 || string > 7)  {  conMove(22, 64);  printf("Введіть ще раз");  }  break;  case 54:  clear();  task3(G);  min = 30;  max = -30;  for (i = 0; i <= 7; i++)  {  for (j = 0; j <= 7; j++)  {  if(G[i][j] < min)  {  min = G[i][j];  index1 = i;  index2 = j;  }  }  }  for (i = 0; i <= 7; i++)  {  for (j = 0; j <= 7; j++)  {  if(G[i][j] > max)  {  max = G[i][j];  indexm1 = i;  indexm2 = j;  }  }  }  G[index1][index2] = max;  G[indexm1][indexm2] = min;  for (i = 0; i <= 7; i++)  {  conMove(18 + i, 64);  for (j = 0; j <= 7; j++)  {  if((i ==index1 && j ==index2) || (i == indexm1 && j == indexm2))  {  conSetAttr(FG\_RED);  printf(" %\*i",3, G[i][j]);  conReset();  }  else  {  printf(" %\*i",3, G[i][j]);  }  }  printf("\n");  }  break;  case 55:  task3(G);  clear();  flag = 0;  numberstr[10] = '\0';  conMove(18,64);  printf("введіть рядок в якому поміняти елемент ");  fgets(numberstr, 10, stdin);  for (k = 0; k < 10; k++)  {  if(isdigit(numberstr[k]))  {  flag = 1;  }  }  if (flag == 1)  {  index1 = atoi(numberstr);  }  if(index1 > 7 || flag == 0)  {  conMove(19,64);  conSetAttr(BG\_RED);  printf("Введіть ще раз");  conReset();  }  else  {  flag = 0;  conMove(19,64);  printf("введіть стопчик в якому поміняти елемент ");  fgets(numberstr, 10, stdin);  for (k = 0; k < 10; k++)  {  if(isdigit(numberstr[k]))  {  flag = 1;  }  }  if (flag == 1)  {  index2 = atoi(numberstr);  }  if(index2 > 7 || flag == 0)  {  conMove(20,64);  conSetAttr(BG\_RED);  printf("Введіть ще раз");  conReset();  }  else  {  flag = 0;  conMove(20,64);  printf("введіть значення нового елемента ");  fgets(numberstr, 10, stdin);  for (k = 0; k < 10; k++)  {  if(isdigit(numberstr[k]))  {  temp = atoi(numberstr);  flag =1;  }  }  if (flag ==0)  {  conMove(21,64);  conSetAttr(BG\_RED);  printf("Введіть ще раз");  conReset();  }  else  {  G[index1][index2]=temp;  for (i = 0; i <= 7; i++)  {  conMove(21 + i,64);  for (j = 0; j <= 7; j++)  {  if(i == index1 && j == index2)  {  conSetAttr(FG\_RED);  printf(" %\*i",3, G[i][j]);  conReset();  }  else  {  printf(" %\*i",3, G[i][j]);  }  }  }  }  }  }  break;  }/\*3\*/  }/\*3\*/  break;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  case 52:  task41(str);  clear();  clear1();  clear2();  clear3();  line1();  task4(str);  yellowline();  conSetAttr(BG\_YELLOW);  conSetAttr(FG\_BLACK);  conMove(3,55);  printf("TASK 4\n");  conReset();  conMove(5,3);  printf("1 - Заповнити строку вводом із консолі.");  conMove(6,3);  printf("2 - Очистити строку.");  conMove(7,3);  printf("3 - Вивести довжину строки.");  conMove(8,3);  printf("4 - Вивести підстроку із заданої позиції і заданої довжини.");  conMove(9,3);  printf("5 - Вивести список підстрок, розділених заданим символом.");  conMove(10,3);  printf("6 - Вивести найдовше слово (слова - непуста послідовність буквенних символів).");  conMove(11,3);  printf("7 - Знайти та вивести всі цілі числа, що містяться у строці.");  conMove(12,3);  printf("8 - Знайти та вивести суму всіх дробових чисел, що містяться у строці.");  conMove(13,3);  printf("m - Меню");  while(1)  {  k = conGetChar();  conReset();  if(k == 109)  {  menu();  break;  }  switch(k)  {  case 49:  clear1();  conMove(17,5);  printf("Введіть строку :");  clear3();  line1();  conMove(17,3);  fgets(str, 82, stdin);  conMove(17,3);  puts(str);  break;  case 50:  clear1();  clear4();  line1();  str[0] = '\0';  break;  case 51:  clear4();  line1();  task4(str);  size = strlen(str) -1;  if (size != -1)  {  conMove(21,5);  printf("довжина строки = %i", size);  }  else  {  conMove(22,5);  printf("довжина строки = 0");  }  break;  case 52:  clear4();  line1();  size = strlen(str) - 1;  task4(str);  conMove(21,5);  printf("введіть позицію з якої вивести строку");  fgets(numberstr, 10, stdin);  for (k = 0; k < 10; k++)  {  if(isdigit(numberstr[k]))  {  poselement = atoi(numberstr);  }  }  if(poselement > size)  {  conMove(22,5);  printf("Довжина строки менша за цю позицію");  }  else  {  conMove(22,5);  printf("введіть довжину строки");  fgets(numberstr, 10, stdin);  for (k = 0; k < 10; k++)  {  if(isdigit(numberstr[k]))  {  newstrlenth = atoi(numberstr);  }  }  if(newstrlenth > size)  {  conMove(22,5);  printf("Довжина строки менша за цю позицію");  }  else  {  newstrsize = poselement + newstrlenth;  for(i=5; poselement < newstrsize; poselement ++,i++)  {  conMove(23,i);  printf("%c", str[poselement]);  }  }  }  break;  case 53:  clear4();  line1();  task4(str);  conMove(21,5);  printf("введіть символ яким розділенні ");  sim = getchar();  size = strlen(str) - 1;  j = 0;  flag =0;  for(j = 0; j < size; j ++)  {  if(str[j] == sim)  {  flag =1;  }  }  if(flag == 1)  {  conMove(22,5);  for(i = 0; i < size; i++)  {  if (str[i] != sim)  {  printf("%c", str[i]);  }  else  {  printf("\t");  }  }  }  else  {  conMove(22,5);  printf("Не має даного символа");  }  break;  case 54:  clear4();  line1();  task4(str);  count = 0;  endword = 0;  beginword =0;  k =0;  flag = 0;  for(i = 0; i < strlen(str); i++)  {  if(isalpha(str[i]) !=0)  {  count++;  }  else  {  if(endword < count)  {  endword = count;  beginword = i - count;  endword += beginword;  flag =1;  }  count = 0;  }  }  if(flag ==0)  {  conMove(21, 5);  printf("Не має слів");  }  else  {  conMove(21, 5);  printf("Найдовше слово");  conMove(24, 5);  for(j = beginword; j < endword ; j++)  {  printf("%c",str[j]);  }  }  break;  case 55:  clear4();  line1();  task4(str);  size = strlen(str) - 1 ;  count =0;  j=0;  flag = 0;  conMove(22,5);  printf("Цілі числа в строкі");  for(i =0; i < size; i++)  {  char ch = str[i];  char nextCh = str[i + 1];  if (isdigit(ch) || (ch == '-' && isdigit(nextCh)))  {  flag = 1;  conMove(23, j+5);  count = printf("%i", atoi(str +i));  i +=count - 1;  j = j+3;  }  }  if(flag == 0)  {  conMove(23,5);  printf("немає цілих чисел");  }  break;  case 56:  clear4();  line1();  task4(str);  count = 0;  size = strlen(str)-1;  sum3 = 0;  flag = 0;  flag = 0;  for (i = 0; i < size; i++)  {  char ch = str[i];  char nextCh = str[i + 1];  if (isdigit(ch) || ( (ch == '.' ||ch == '-') && isdigit(nextCh) ))  {  for(k=0; k < 5; k++)  {  if(str[i+k] == '.' && isdigit(str[i+k+1]))  {  value = atof(str + i);  sum3 += value;  count ++;  flag = 0;  flag1 =1;  }  }  if (ch == '-')  {  i++;  }  while ((isdigit(str[i]) != 0 || (flag == 0 && str[i] == '.')) && i < size)  {  if (str[i] == '.')  {  flag = 1;  }  i++;  }  i = i - 1;  }  }  if(flag1 ==1)  {  conMove(23, 5);  printf("Cума дробових = %.10f", sum3);  }  else  {  conMove(23, 5);  printf("Немає дробових чисел");  }  break;  }  }  }  }  return 0;  }  void line()  {  int i=0;  conClear();  conSetAttr (BG\_WHITE);  for(i=0; i < 121; i++)  {  printf(" ");  conMove(1,i);  }  for(i = 0; i < 121; i++)  {  printf(" ");  conMove(15,i);  }  for(i = 0; i < 121; i++)  {  printf(" ");  conMove(30,i);  }  for(i = 0; i < 30; i++)  {  printf(" ");  conMove(i,60);  }  for(i = 0; i < 30; i++)  {  printf(" ");  conMove(i,120);  }  for(i = 0; i < 31; i++)  {  printf(" ");  conMove(i,1);  }  conReset();  }  void line1()  {  int i = 0;  for(i =0; i <120; i++)  {  conMove(20, i);  conSetAttr(BG\_WHITE);  printf(" ");  }  conReset();  }  void clear()  {  int i = 0;  int j = 0;  for(i = 17; i < 29; i ++)  {  for(j = 61; j < 120; j++)  {  conMove(i, j );  printf(" ");  }  }  }  void clear1()  {  int i = 0;  int j = 0;  for(i = 17; i < 29; i ++)  {  for(j = 2; j < 59; j++)  {  conMove(i, j );  printf(" ");  }  }  }  void clear2()  {  int i = 0;  int j = 0;  for(i = 2; i < 15; i ++)  {  for(j = 3; j < 117; j++)  {  conMove(i, j );  printf(" ");  }  }  }  void clear3()  {  int i = 0;  int j = 0;  for(i = 16; i < 30; i ++)  {  for(j = 3; j < 117; j++)  {  conMove(i, j );  printf(" ");  }  }  }  void clear4()  {  int i = 0;  int j = 0;  for(i = 17; i < 30; i ++)  {  for(j = 2; j < 119; j++)  {  conMove(i, j );  printf(" ");  }  }  }  void menu()  {  line();  clear2();  conMove(4,5);  printf("Task1 - нажміть 1\n");  conMove(5,5);  printf("Task2 - нажміть 2\n");  conMove(6,5);  printf("Task3 - нажміть 3\n");  conMove(7,5);  printf("Task4 - нажміть 4");  conMove(8,5);  printf("Exit - нажміть Esc");  }  void masiv1(int A[10])  {  int i = 0;  clear1();  conMove(20,3);  for (i = 0; i <= 9; i++)  {  printf(" %i", A[i]);  }  }  void task2(double F[10],double D[10])  {  int i =0;  clear1();  conMove(20,3);  printf("F[10]= \n");  conMove(21,3);  for (i = 0; i <= 9 ; i++)  {  printf(" %.1f", F[i]);  }  conMove(22,3);  printf("D[10]= \n");  conMove(23,3);  for (i = 0; i <= 9; i++)  {  printf(" %.1f", D[i]);  }  }  void task3( int G[8][8])  {  int i =0;  int j = 0;  clear1();  for (i = 0; i <= 7; i++)  {  conMove(18+i, 8);  for (j = 0; j <= 7; j++)  {  printf(" %\*i",3, G[i][j]);  }  }  }  void yellowline()  {  int i =0;  int j =0;  conSetAttr (BG\_YELLOW);  for(j=2; j < 5; j++)  {  for(i=2; i < 120; i++)  {  conMove(j,i);  printf(" ");  }  }  }  void task4(char str[80])  {  conMove(17,3);  puts(str);  }  void task41(char str[80])  {  str[0] = 'T';  str[1] = 'y';  str[2] = 'p';  str[3] = 'e';  str[4] = ' ';  str[5] = 's';  str[6] = 't';  str[7] = 'r';  str[8] = 'i';  str[9] = 'n';  str[10] = 'g';  } |

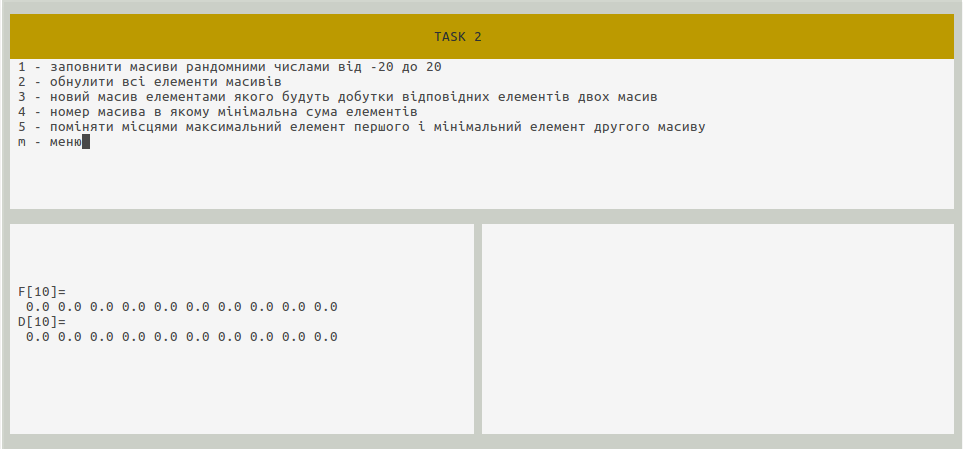
**Тексти коду програм**

**Приклади результатів**

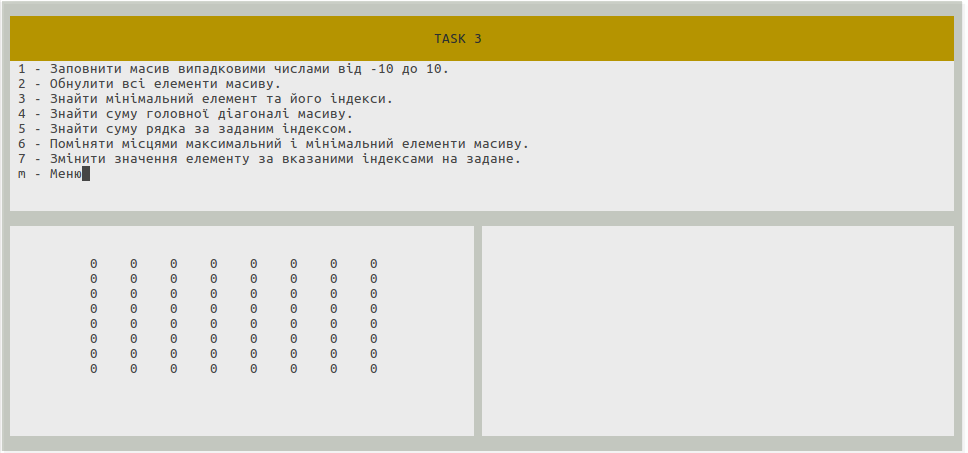
**Завдання 1.**



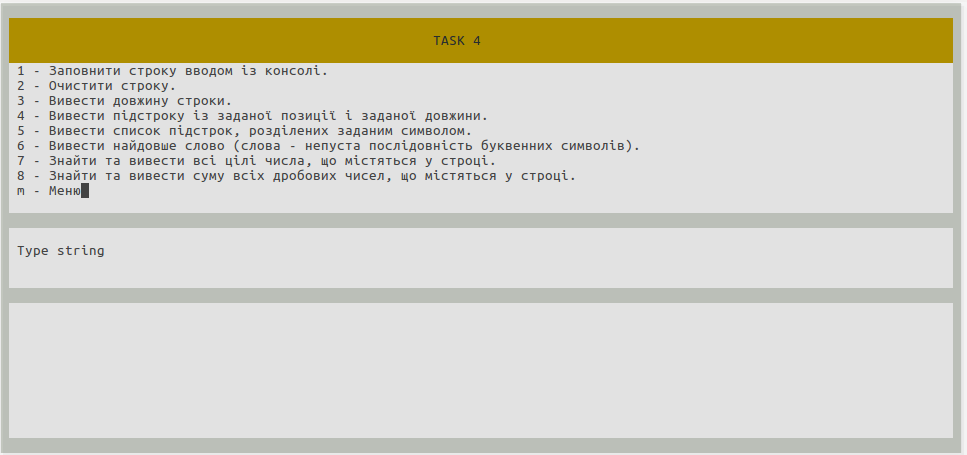
**Завдання 2.**



**Завдання 3.**



**Завдання 4.**



**Висновки**

Ми 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. Навчилися оформлювати консольну програму для зручності роботи користувача.