

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 №** **1**

з дисципліни “ Основи програмування ”

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

|  |  |  |
| --- | --- | --- |
| Виконав  студент I курсу  групи КП-61  Телефус Ілля Анатолійович  (*прізвище, ім’я, по батькові*)  варіант № 27 |  | 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**

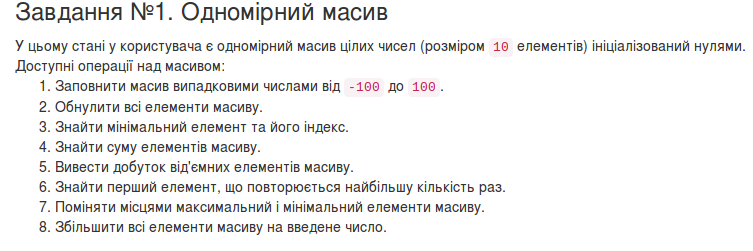
Навчитися працювати зі статичними масивами різних типів даних мови програмування С.

Застосувати на практиці різні види циклічних конструкцій при роботі з одномірними та багатомірними масивами даних.

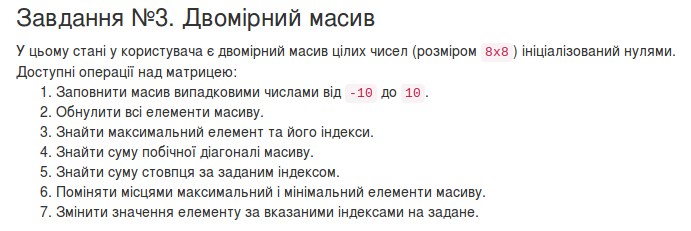
Вдосконалити вміння роботи з рядками у мові С.

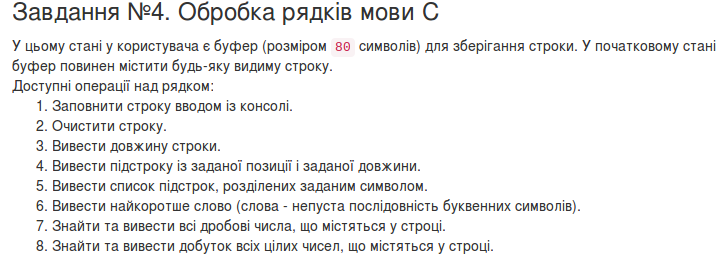
Навчитися оформлювати консольну програму для зручності роботи користувача.

**Постановка завдання**







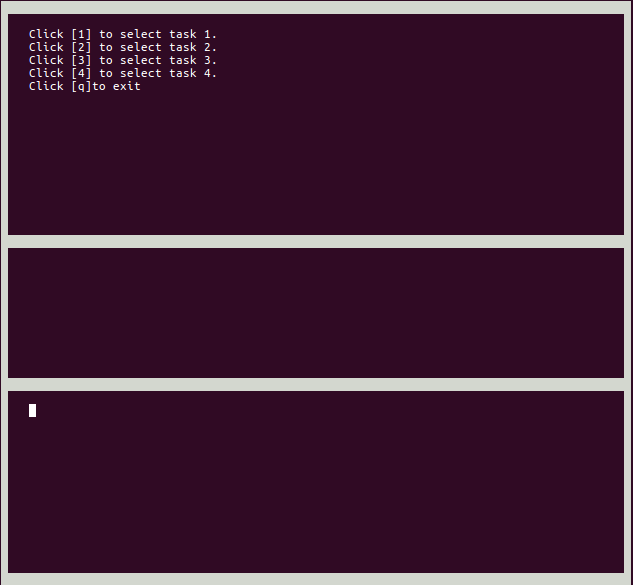


**Тексти коду програм**

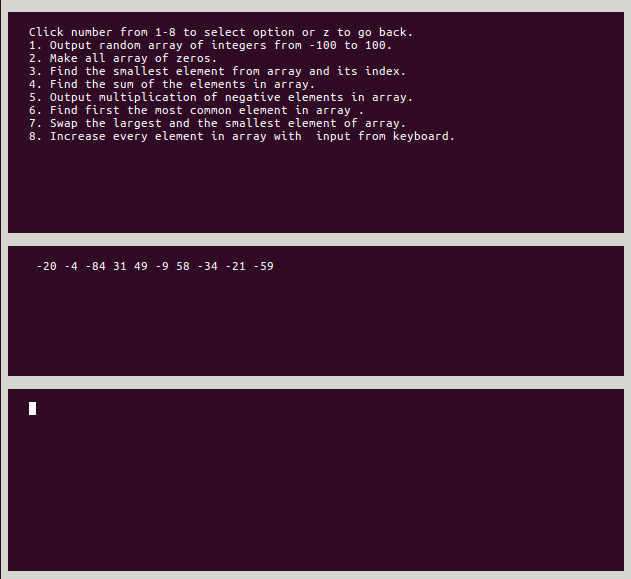
|  |
| --- |
| #include<math.h>  #include<stdlib.h>  #include<stdio.h>  #include<progbase.h>  #include<math.h>  #include<string.h>  #include<time.h>  #include<ctype.h>  #include<pbconsole.h>  #include<stdbool.h>  void bordergraphics();  void clean();  void menu();  void clean2();  void clean3();  int main(void) {  /\* reading the user input\*/  char option;  /\* controlling the while loop\*/  bool isRunning = true;  bool Running\_case = true;  int A[10] = {0};  double A\_doub2[10] = {0};  double A\_doub1[10] = {0};  double A\_doub\_devide[10] = {0};  int A\_two\_dim[8][8] = {0};  char string[80] = {"hi, dear user, I`m the string for task 4. Are you going to do something with me?"};  int i = 0;  int j = 0;  int size\_A = sizeof(A) / sizeof(A[0]);  int size\_A\_doub1 = sizeof(A\_doub1) / sizeof(A\_doub1[0]);  int size\_A\_doub2 = sizeof(A\_doub2) / sizeof(A\_doub2[0]);  int size\_A\_doub\_devide = sizeof(A\_doub\_devide) / sizeof(A\_doub\_devide[0]);  int length\_A\_two\_dim = 8;  int size\_string = 0;  /\*int size\_A\_two\_dim = 64;\*/  /\*int row;\*/  /\*int col;\*/  int key\_input = 0;  int key\_input\_y = 0;  int key\_input\_el = 0;  int sum = 0;  double sum\_doub1 = 0;  double sum\_doub2 = 0;  int multiply = 1;  double random;  int random\_3tsk;  int min\_arr = 0;  int index\_max\_arr = 0;  int index\_min\_arr = 0;  int index\_min\_x = 0;  int index\_min\_y = 0;  int index\_max\_x = 0;  int index\_max\_y = 0;  int max\_arr = 0;  double max\_arr\_d = 0;  double min\_arr\_d = 0;  int maxCount = 0;  int element;  int str\_input\_begin = 0;  int str\_input\_end = 0;  char character\_key;  int flag;  int length = 0;  int count = 0;  int word\_1 = 0;  int word\_2 = 0;  int count\_before = 0;  int count\_after = 0;  char maininput[10];  srand(time(0));  bordergraphics();  while (isRunning == true) {  /\*Clears the screen\*/  /\*Clears the keyboard buffer\*/  fflush(stdin);  /\*Outputs the options to console\*/  menu();  /\*Reads the user's option\*/  option = conGetChar(); /\*check\*/ /\*here i have changed getchar with conGetChar\*/  /\*Selects the course of action specified by the option\*/  switch (option) {  case '1':  conMove(3, 5);  puts("Click number from 1-8 to select option or z to go back.");  conMove(4, 5);  puts("1. Output random array of integers from -100 to 100.");  conMove(5, 5);  puts("2. Make all array of zeros.");  conMove(6, 5);  puts("3. Find the smallest element from array and its index.");  conMove(7, 5);  puts("4. Find the sum of the elements in array.");  conMove(8, 5);  puts("5. Output multiplication of negative elements in array.");  conMove(9, 5);  puts("6. Find first the most common element in array .");  conMove(10, 5);  puts("7. Swap the largest and the smallest element of array.");  conMove(11, 5);  puts("8. Increase every element in array with input from keyboard.");  conMove(32, 5);  /\*here commented that staff\*/  Running\_case = true;  while (Running\_case == true) {  option = conGetChar();  switch (option) {  case '1':  clean2();  conMove(21, 5);  for (i = 0; i < size\_A; i++) {  random = rand() % 200 - 100;  A[i] = random;  }  for (i = 0; i < size\_A; i++) {  printf(" %i", A[i]);  }  conReset();  conMove(32, 5);  break;  case '2':  clean2();  conMove(21, 5);  for (i = 0; i < size\_A; i++) {  A[i] = 0;  printf(" %i", A[i]);  }  conReset();  conMove(32, 5);  break;  case '3':  clean2();  conMove(21, 5);  min\_arr = A[1];  for (i = 0; i < size\_A; i++) {  if (A[i] < min\_arr) {  min\_arr = A[i];  index\_min\_arr = i;  }  }  conMove(21, 5);  printf("Here is the minimal element in array%i\n", min\_arr);  conMove(22, 5);  printf("Here is index of minimal element in array%i\n", index\_min\_arr);  conReset();  conMove(32, 5);  break;  case '4':  clean2();  for (i = 0; i < size\_A; i++) {  sum = sum + A[i];  }  conMove(21, 5);  printf("Here is sum of elements from array:%i\n", sum);  conReset();  conMove(32, 5);  break;  case '5':  clean2();  for (i = 0; i < size\_A; i++) {  if (A[i] < 0) {  multiply = multiply \* A[i];  }  }  for (i = 0; i < size\_A; i++) {  if (A[i] == 0) {  multiply = 0;  }  }  conMove(21, 5);  printf("Here is multiplication of negative elements from array:%i\n", multiply);  conReset();  conMove(32, 5);  break;  case '6':  clean2();  maxCount = 0;  element = 0;  for (i = 0; i < size\_A; i++) {  count = 0;  for (j = 0; j < size\_A; j++) {  if (A[i] == A[j]) {  count++;  }  if (maxCount < count) {  maxCount = count;  element = A[i];  }  }  }  conMove(21, 5);  printf("The most common element is: %i\n", element);  conReset();  conMove(32, 5);  break;  case '7':  clean2();  min\_arr = A[1];  for (i = 0; i < size\_A; i++) {  if (A[i] < min\_arr) {  min\_arr = A[i];  index\_min\_arr = i;  }  }  max\_arr = A[1];  for (i = 0; i < size\_A; i++) {  if (A[i] > max\_arr) {  max\_arr = A[i];  index\_max\_arr = i;  }  }  A[index\_max\_arr] = min\_arr;  A[index\_min\_arr] = max\_arr;  conMove(21, 5);  printf("Here we have swaped the minimum and maximum elements:\n");  conMove(22, 5);  for (i = 0; i < size\_A; i++) {  if (i == index\_max\_arr || i == index\_min\_arr) {  conSetAttr(FG\_RED);  printf(" %i", A[i]);  conReset();  } else {  printf(" %i", A[i]);  }  }  conReset();  conMove(32, 5);  break;  case '8':  clean2();  conMove(21, 5);  printf("Type in the integer number to increase every element of array with:\n");  conMove(32, 5);  flag = 0;  fgets(maininput, 10, stdin);  clean3();  for (i = 0; i < 10; i++) {  if (isdigit(maininput[i])) {  key\_input = atoi(maininput);  flag = 1;  }  }  if (flag == 0) {  conMove(23, 5);  printf("Error type number");  }  conMove(22, 5);  for (i = 0; i < size\_A; i++) {  A[i] = A[i] + key\_input;  printf(" %i", A[i]);  }  conReset();  conMove(32, 5);  break;  case 'z':  Running\_case = false;  clean();  menu();  break;  }  }  clean2();  break;  case '2':  clean();  conMove(3, 5);  puts("Click number from 1-5 to select option or z to go back.");  conMove(4, 5);  puts("1. Output random arrays of doubles from -20 to 20.");  conMove(5, 5);  puts("2. Make all arrays of zeros.");  conMove(6, 5);  puts("3. Find the array whose each element is dividation of elements from 1 and 2 arrays.");  conMove(7, 5);  puts("4. Find array whose sum of elements is bigger.");  conMove(8, 5);  puts("5. Swap the smallest element from 1 array and the biggest from 2 array.");  conMove(32, 5);  conReset();  Running\_case = true;  while (Running\_case == true) {  option = conGetChar(); /\*made congetchar instead of getchar\*/  switch (option) {  case '1':  clean2();  conMove(22, 5);  printf("Here are 2 arrays of random double numbers:\n");  srand(time(0));  conMove(23, 5);  for (i = 0; i < size\_A\_doub1; i++) {  random = -20 + rand() / (float)(RAND\_MAX - 1) + rand() % 40;  A\_doub1[i] = random;  printf(" %.2f", A\_doub1[i]);  }  printf("\n");  conMove(24, 5);  for (i = 0; i < size\_A\_doub2; i++) {  random = -20 + rand() / (float)(RAND\_MAX - 1) + rand() % 40;  A\_doub2[i] = random;  printf(" %.2f", A\_doub2[i]);  }  conReset();  conMove(32, 5);  break;  case '2':  clean2();  conMove(22, 4);  printf("Here are 2 arrays of zeroes:\n");  conMove(23, 4);  for (i = 0; i < size\_A\_doub1; i++) {  A\_doub1[i] = 0;  printf(" %.2f", A\_doub1[i]);  }  printf("\n");  conMove(24, 4);  for (i = 0; i < size\_A\_doub2; i++) {  A\_doub2[i] = 0;  printf(" %.2f", A\_doub2[i]);  }  conReset();  conMove(32, 5);  break;  case '3':  clean2();  conMove(22, 4);  printf("Here is an array whose each element is dividation of elements from 1 and 2 arrays:\n");  for (i = 0; i < size\_A\_doub1; i++) {  }  for (i = 0; i < size\_A\_doub2; i++) {  }  conMove(23, 4);  for (i = 0; i < size\_A\_doub\_devide; i++) {  A\_doub\_devide[i] = A\_doub1[i] / A\_doub2[i];  printf(" %.2f", A\_doub\_devide[i]);  }  conReset();  conMove(32, 5);  break;  case '4':  clean2();  conMove(22, 5);  printf("Here we gonna find array whose sum of elements is bigger.");  conMove(23, 5);  printf("Here is an array whose sum of elements is bigger:");  for (i = 0; i < size\_A\_doub1; i++) {  sum\_doub1 = sum\_doub1 + A\_doub1[i];  }  for (i = 0; i < size\_A\_doub2; i++) {  sum\_doub2 = sum\_doub2 + A\_doub2[i];  }  if (sum\_doub1 > sum\_doub2) {  conMove(24, 5);  for (i = 0; i < size\_A\_doub1; i++) {  printf(" %.2f", A\_doub1[i]);  }  conReset();  conMove(25, 5);  printf("it`s number is 1");  } else {  conMove(24, 5);  for (i = 0; i < size\_A\_doub1; i++) {  printf(" %.2f", A\_doub2[i]);  }  conMove(25, 5);  printf("its number is 2");  }  conReset();  conMove(32, 5);  break;  case '5':  clean2();  min\_arr\_d = 20;  index\_max\_arr = 0;  index\_min\_arr = 0;  min\_arr = A\_doub1[0];  conMove(21, 5);  printf("Here we swaped the smallest element from 1 array and the biggest from 2 array:\n");  for (i = 0; i < size\_A\_doub1; i++) {  if (A\_doub1[i] < min\_arr\_d) {  min\_arr\_d = A\_doub1[i];  index\_min\_arr = i;  }  }  max\_arr\_d = -20;  for (i = 0; i < size\_A\_doub2; i++) {  if (max\_arr\_d < A\_doub2[i]) {  max\_arr\_d = A\_doub2[i];  index\_max\_arr = i;  }  }  A\_doub2[index\_max\_arr] = min\_arr\_d;  A\_doub1[index\_min\_arr] = max\_arr\_d;  conMove(22, 5);  for (i = 0; i < size\_A\_doub1; i++) {  printf(" %.2f", A\_doub1[i]);  }  clean2();  conMove(21, 5);  for (i = 0; i < size\_A\_doub2; i++) {  if (i == index\_max\_arr) {  conSetAttr(FG\_RED);  printf(" %.2f", A\_doub2[i]);  conReset();  } else {  printf(" %.2f", A\_doub2[i]);  }  }  conMove(22, 5);  for (i = 0; i < size\_A\_doub1; i++) {  if (i == index\_min\_arr) {  conSetAttr(FG\_RED);  printf(" %.2f", A\_doub1[i]);  conReset();  } else {  printf(" %.2f", A\_doub1[i]);  }  }  conReset();  conMove(32, 5);  break;  case 'z':  Running\_case = false;  clean();  menu();  break;  }  }  clean2();  break;  case '3':  conMove(3, 5);  puts("Click number from 1-8 to select option or z to go back.");  conMove(4, 5);  puts("1. Output random 2 dimensional array of integers from -8 to 8.");  conMove(5, 5);  puts("2. Make all array of zeros.");  conMove(6, 5);  puts("3. Find the biggest element from array and its index.");  conMove(7, 5);  puts("4. Find the sum of the elements in pobichna diagonal.");  conMove(8, 5);  puts("5. Find the sum of the elements in chosen column (you have to choose).");  conMove(9, 5);  puts("6. Find the biggest element in array.");  conMove(10, 5);  puts("7. Swap the largest and the smallest element of array.");  conMove(11, 5);  puts("8. Change the element with inputed cordinates with inputed from keyboard.");  conMove(32, 5);  conReset();  /\*option = getchar();\*/  /\*removed option = getchar()\*/  Running\_case = true;  while (Running\_case == true) {  option = conGetChar(); /\*made congetchar instead of getchar\*/  switch (option) {  case '1':  clean2();  srand(time(0));  for (i = 0; i < length\_A\_two\_dim; i++) { /\*there isnt right output there and next when 2dim array\*/  conMove(21 + i, 5);  for (j = 0; j < length\_A\_two\_dim; j++) {  random\_3tsk = rand() % 20 - 10;  A\_two\_dim[i][j] = random\_3tsk;  printf(" %\*i", 3, A\_two\_dim[i][j]);  }  }  conReset();  conMove(32, 5);  break;  case '2':  clean2();  for (i = 0; i < length\_A\_two\_dim; i++) {  conMove(21 + i, 5);  for (j = 0; j < length\_A\_two\_dim; j++) {  A\_two\_dim[i][j] = 0;  printf(" %\*i", 3, A\_two\_dim[i][j]);  }  }  conReset();  conMove(32, 5);  break;  case '3':  clean2();  max\_arr = 0;  index\_max\_arr = 0;  index\_min\_arr = 0;  max\_arr = A\_two\_dim[0][0];  for (i = 0; i < length\_A\_two\_dim; i++) {  for (j = 0; j < length\_A\_two\_dim; j++) {  if (max\_arr < A\_two\_dim[i][j]) {  max\_arr = A\_two\_dim[i][j];  index\_max\_arr = i;  index\_min\_arr = j;  }  }  }  conMove(21, 5);  printf("here is the maximal element in array %i", max\_arr);  conMove(22, 5);  printf("here is index of minimal element in array(x and y cordinates): %i %i", index\_min\_arr, index\_max\_arr);  conReset();  conMove(32, 5);  break;  case '4':  clean2();  sum = 0;  for (i = 0, j = 0; i < length\_A\_two\_dim && j < length\_A\_two\_dim; i++, j++) {  sum += A\_two\_dim[j][length\_A\_two\_dim - 1 - i];  }  conMove(21, 5);  printf("Here is sum of pobichna diagonal:%i\n", sum);  conReset();  conMove(32, 5);  break;  case '5':  clean2();  key\_input = 0;  sum = 0;  conMove(32, 5);  printf("Enter number of row sum of which you wanna count\n");  conMove(33, 5);  flag = 0;  fgets(maininput, 10, stdin);  clean3();  for (i = 0; i < 10; i++) {  if (isdigit(maininput[i])) {  key\_input = atoi(maininput);  if (key\_input < 8) {  flag = 1;  for (j = 0; j < length\_A\_two\_dim; j++) {  sum += A\_two\_dim[j][key\_input];  }  conMove(21, 5);  printf("Here is sum of chosen column:%i\n", sum);  conReset();  conMove(32, 5);  }  }  }  if (flag == 0) {  conMove(23, 5);  printf("Error type number");  }  /\*scanf("%i", &key\_input);\*/  /\*for(j = 0; j < length\_A\_two\_dim; j++){  sum += A\_two\_dim[j][key\_input];  }  conMove(21, 5);  printf("Here is sum of chosen column:%i\n", sum);  conReset();  conMove(32, 5);\*/  break;  case '6':  clean2();  max\_arr = -10;  conMove(32, 5);  printf("Here we gonna find the biggest element in array\n");  A\_two\_dim[0][0] = max\_arr;  clean2();  for (i = 0; i < length\_A\_two\_dim; i++) {  for (j = 0; j < length\_A\_two\_dim; j++) {  if (A\_two\_dim[i][j] > max\_arr) {  max\_arr = A\_two\_dim[i][j];  }  }  }  conMove(21, 5);  printf("The biggest element is: %i", max\_arr);  conReset();  conMove(32, 5);  break;  case '7':  clean2();  conMove(20, 5);  printf("Swap the biggest and the smallest elements:%i\n", multiply);  max\_arr = -10;  for (i = 0; i < 8; i++) {  conMove(21 + i, 5);  clean2();  for (j = 0; j < 8; j++) {  if (A\_two\_dim[i][j] > max\_arr) {  max\_arr = A\_two\_dim[i][j];  index\_max\_x = i;  index\_max\_y = j;  }  }  }  min\_arr = 10;  for (i = 0; i < 8; i++) {  for (j = 0; j < 8; j++) {  if (A\_two\_dim[i][j] < min\_arr) {  min\_arr = A\_two\_dim[i][j];  index\_min\_x = i;  index\_min\_y = j;  }  }  }  A\_two\_dim[index\_min\_x][index\_min\_y] = max\_arr;  A\_two\_dim[index\_max\_x][index\_max\_y] = min\_arr;  for (i = 0; i < length\_A\_two\_dim; i++) {  conMove(21 + i, 5);  for (j = 0; j < length\_A\_two\_dim; j++) {  if ((i == index\_max\_x || j == index\_max\_y) && (i == index\_min\_x || j == index\_min\_y)) {  conSetAttr(FG\_RED);  printf(" %\*i", 3, A\_two\_dim[i][j]);  conReset();  } else {  printf(" %\*i", 3, A\_two\_dim[i][j]);  }  }  printf("\n");  }  conReset();  conMove(32, 5);  break;  case '8':  clean2();  conMove(21, 5);  printf("Type in cordinates (x then y) of element that you wanna change and value of new one\n");  conMove(32, 5);  flag = 0;  fgets(maininput, 10, stdin);  clean3();  for (i = 0; i < 10; i++) {  if (isdigit(maininput[i])) {  key\_input = atoi(maininput);  flag = 1;  }  }  if (flag == 0) {  conMove(23, 5);  printf("Error (you`ve entered not a number) press any key from 1-8");  }  break;  /\*scanf("%i", &key\_input);\*/  conMove(33, 5);  flag = 0;  fgets(maininput, 10, stdin);  clean3();  for (i = 0; i < 10; i++) {  if (isdigit(maininput[i])) {  key\_input\_y = atoi(maininput);  flag = 1;  }  }  if (flag == 0) {  conMove(23, 5);  printf("Error (you`ve entered not a number) press any key from 1-8");  }  /\*scanf("%i", &key\_input\_y);\*/  conMove(34, 5);  flag = 0;  fgets(maininput, 10, stdin);  clean3();  for (i = 0; i < 10; i++) {  if (isdigit(maininput[i])) {  key\_input\_el = atoi(maininput);  flag = 1;  }  }  if (flag == 0) {  conMove(23, 5);  printf("Error (you`ve entered not a number) press any key from 1-8");  }  /\*scanf("%i", &key\_input\_el);\*/  conMove(22, 5);  clean2();  for (i = 0; i < length\_A\_two\_dim; i++) {  conMove(21 + i, 5);  for (j = 0; j < length\_A\_two\_dim; j++) {  A\_two\_dim[key\_input][key\_input\_y] = key\_input\_el;  printf(" %\*i", 3, A\_two\_dim[i][j]);  }  printf("\n");  }  conReset();  conMove(32, 5);  break;  case 'z':  Running\_case = false;  clean();  menu();  break;  }  }  clean2();  break;  case '4':  clean();  conMove(3, 5);  puts("Click number from 1-8 to select option or z to go back.");  conMove(4, 5);  puts("1. Type in 80 characters.");  conMove(5, 5);  puts("2. Make the string empty.");  conMove(6, 5);  puts("3. Output string length.");  conMove(7, 5);  puts("4. Here we output string from start position and string length.");  conMove(8, 5);  puts("5. Here we output string pieces devided by inputed character.");  conMove(9, 5);  puts("6. Output the longest word in the string.");  conMove(10, 5);  puts("7. Output all double values from the string.");  conMove(11, 5);  puts("8. Output multiplication of integers from the string.");  conMove(32, 5);  conReset();  Running\_case = true;  while (Running\_case == true) {  option = conGetChar();  switch (option) {  case '1':  clean2();  conMove(21, 5);  printf("Type in 80 characters:\n");  conMove(32, 5);  fgets(string, 82, stdin);  clean2();  conMove(21, 5);  printf("here is your string:");  conMove(22, 5);  puts(string);  clean3();  conReset();  conMove(32, 5);  break;  case '2':  clean2();  conMove(21, 5);  printf("Here we make our string empty:\n");  strcpy(string, " ");  conMove(32, 5);  puts(string);  clean3();  conReset();  conMove(32, 5);  break;  case '3':  clean2();  conMove(21, 5);  printf("Here we output the string length.\n");  size\_string = strlen(string) - 1;  conMove(22, 5);  printf("String length is: %i\n", size\_string);  clean3();  conReset();  conMove(32, 5);  break;  case '4':  clean2();  i = 0;  conMove(21, 5);  printf("Type in start position of string and its length.\n");  conMove(32, 5);  flag = 0;  fgets(maininput, 10, stdin);  clean3();  clean2();  for (i = 0; i < 10; i++) {  if (isdigit(maininput[i])) {  str\_input\_begin = atoi(maininput);  flag = 1;  }  }  if (flag == 0) {  conMove(23, 5);  printf("Error type number");  }  /\*scanf("%i", &str\_input\_begin);\*/  conMove(32, 5);  flag = 0;  fgets(maininput, 10, stdin);  clean3();  clean2();  for (i = 0; i < 10; i++) {  if (isdigit(maininput[i])) {  str\_input\_end = atoi(maininput);  flag = 1;  }  }  if (flag == 0) {  conMove(23, 5);  printf("Error type number");  }  /\*scanf("%i", &str\_input\_end);\*/  conMove(21, 5);  printf("Here is your string:\n");  conMove(22, 5);  for (i = str\_input\_begin; i < str\_input\_end; i++) {  printf("%c", string[i]);  }  clean3();  conReset();  conMove(32, 5);  break;  case '5':  clean2();  conMove(21, 5);  printf("Type in word which will devide string in pieces.\n");  character\_key = getchar();  conMove(22, 5);  for (i = 0; i < strlen(string); i++) {  if (character\_key != string[i]) {  printf("%c", string[i]);  } else {  printf("\n");  }  }  clean3();  conReset();  conMove(32, 5);  break;  case '6':  clean2();  conMove(21, 5);  printf("Here we gonna output the longest word.\n");  conMove(22, 5);  printf("The longest word is:\n");  for (i = 0; i < strlen(string); i++) {  if (isalpha(string[i])) {  word\_1++;  } else {  if (word\_2 < word\_1) {  word\_2 = word\_1;  count = i - word\_1;  word\_2 += count;  }  word\_1 = 0;  }  }  conMove(23, 5);  for (i = count; i < word\_2; i++) {  printf("%c", string[i]);  }  clean3();  conReset();  conMove(32, 5);  break;  case '7':  clean2();  conMove(21, 5);  printf("Here we gonna output all double values from the string.\n");  for (i = 0; i < strlen(string); i++) {  if (isdigit(string[i]) != 0 && string[i + 1] == '.' && isdigit(string[i + 2]) != 0) {  count\_before = i;  count\_after = i + 2;  while (isdigit(string[count\_before - 1]) != 0) {  count\_before--;  }  conMove(22, 5);  if (string[count\_before - 1] == '-') {  printf("-");  }  while (isdigit(string[count\_after]) != 0) {  count\_after++;  }  conMove(32, 6);  for (i = count\_before; i < count\_after; i++) {  printf("%c", string[i]);  }  printf(" ");  }  }  clean3();  conReset();  conMove(32, 5);  break;  case '8':  clean2();  conMove(21, 5);  printf("Here we gonna output multiplication of integers from the string.\n");  flag = 0;  multiply = 1;  length = strlen(string);  for (i = 0; i < length; i++) {  char ch = string[i];  char nextCh = string[i + 1];  conMove(22, 5);  if (isdigit(ch) || (ch == '-' && isdigit(nextCh))) {  flag = 1;  count = printf("%i", atoi(string + i));  printf(" ");  multiply \*= atoi(string + i);  i += count - 1;  }  }  conMove(23, 5);  if (flag == 0) {  printf("there is no numbers iin the string!");  }  conMove(24, 5);  printf("\nHere is multiply %i\n", multiply);  clean3();  conReset();  conMove(32, 5);  break;  case 'z':  Running\_case = false;  clean();  menu();  break;  }  }  break;  case 'q':  /\*Exits the system\*/  isRunning = false;  return 0;  default:  conMove(21, 5);  printf("Hehehe, type in a number)");  conReset();  break;  }  }  return 0;  }  void bordergraphics() {  int col = 0;  int row = 0;  conClear();  conSetAttr(BG\_WHITE);  for (col = 0; col < 90; col++) {  conMove(1, col);  printf(" ");  conMove(19, col);  printf(" ");  conMove(30, col);  printf(" ");  conMove(45, col);  printf(" ");  }  for (row = 0; row < 45; row++) {  conMove(row, 1);  printf(" ");  }  for (row = 0; row < 46; row++) {  conMove(row, 90);  printf(" ");  }  conReset();  }  void clean() {  int x = 0;  int y = 0;  for (x = 3; x < 89; x++) {  for (y = 2; y < 19; y++) {  conMove(y, x);  puts(" ");  }  }  }  void clean2() {  int x = 0;  int y = 0;  for (x = 3; x < 90; x++) {  for (y = 20; y < 30; y++) {  conMove(y, x);  puts(" ");  }  }  }  void clean3() {  int x = 0;  int y = 0;  for (x = 3; x < 90; x++) {  for (y = 31; y < 45; y++) {  conMove(y, x);  puts(" ");  }  }  }  void menu() {  conMove(3, 5);  puts("Click [1] to select task 1.");  conMove(4, 5);  puts("Click [2] to select task 2.");  conMove(5, 5);  puts("Click [3] to select task 3.");  conMove(6, 5);  puts("Click [4] to select task 4.");  conMove(7, 5);  puts("Click [q]to exit");  conMove(32, 5);  conReset();  } |

**Приклади результатів**

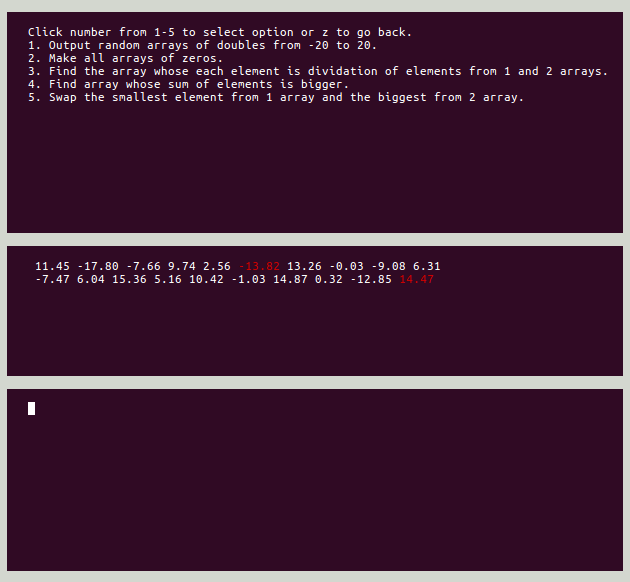
**Меню.**



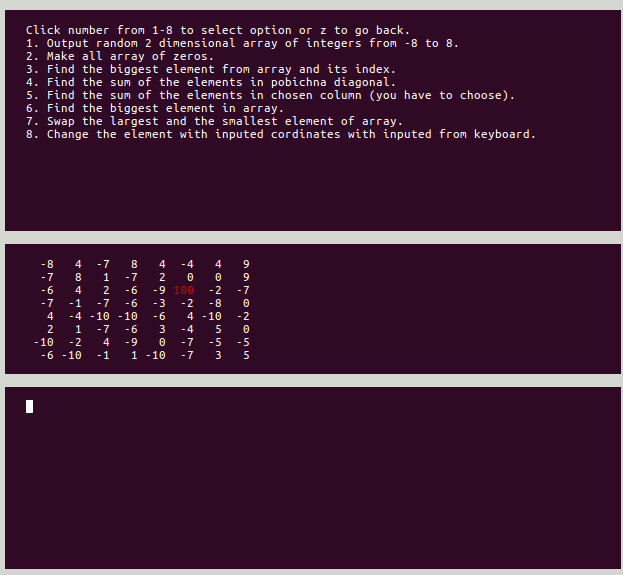
**Завдання 1.**



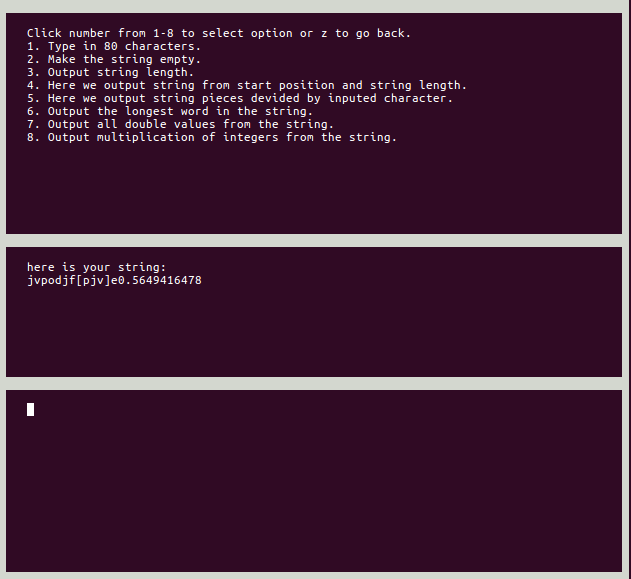
**Завдання 2.**

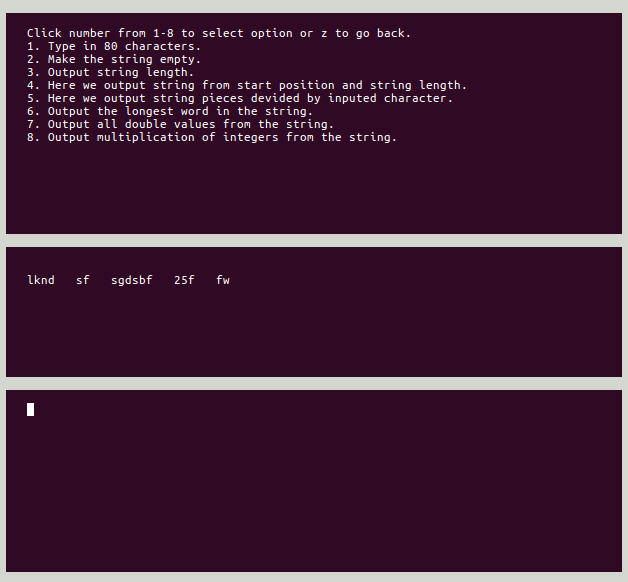


**Завдання 3.**



**Завдання 4.**





**Висновки**

Виконавши дану лабораторну роботу ми навчилися працювати з масивами та строками в мові програмування С.

Застосувати на практиці різні види циклічних конструкцій при роботі з одномірними та багатомірними масивами даних, а також вдосконалили уміння аналізувати введені дані з строки .

Вдосконалити вміння роботи з рядками у мові С.

Створили консольну програму з доволі зручним інтерфейсом.

В цілому, мета досягнена.