**ПРАКТИЧНЕ ЗАВДАННЯ № 9**

**Завдання № 1:** Напишіть програму, в якій реалізуйте виконання поставлених задач з

використанням вказівників.

1. Знаходження транспонованої матриці будь-якого розміру.

2. Знаходження суми двох матриць (враховуйте всі необхідні перевірки)

Розмірність та елементи матриць мають задаватися з клавіатури.

**1)**

**Код програми:**

**#include <stdio.h>**

**void size\_entry(int \*h, int \*len);**

**void elem\_entry(int h, int len, int(\*matrix)[len]);**

**void show\_matrix(int h, int len, int(\*matrix)[len]);**

**int main(void) {**

**int h, len;**

**size\_entry(&h, &len);**

**int a[h][len];**

**int transpose[len][h];**

**puts(" Enter elements of matrix:");**

**elem\_entry(h, len, a);**

**// finding the transpose of matrix**

**for (int row = 0; row < h; row++) {**

**for (int col = 0; col < len; col++) {**

**\*(\*(transpose + col) + row) = \*(\*(a + row) + col);**

**}**

**}**

**puts("\n Entered Matrix:");**

**show\_matrix(h, len, a);**

**puts("\n Transpose of Matrix:");**

**show\_matrix(len, h, transpose);**

**return 0;**

**}**

**void size\_entry(int \*h, int \*len) {**

**do {**

**fseek(stdin, 0, SEEK\_END); // buffer cleaning**

**printf("Enter size with whitespace: ");**

**} while (scanf("%d %d", h, len) != 2); // until it's entered correctly**

**}**

**void elem\_entry(int h, int len, int(\*matrix)[len]) {**

**for (int row = 0; row < h; row++) {**

**for (int col = 0; col < len; col++) {**

**do {**

**fseek(stdin, 0, SEEK\_END); // buffer cleaning**

**printf(" Enter elem a%d%d: ", row + 1, col + 1);**

**} while (!scanf("%d[^\n]", \*(matrix + row) + col));**

**}**

**}**

**}**

**void show\_matrix(int h, int len, int(\*matrix)[len]) {**

**for (int row = 0; row < h; row++) {**

**for (int col = 0; col < len; col++) {**

**printf("%3d", \*(\*(matrix + row) + col));**

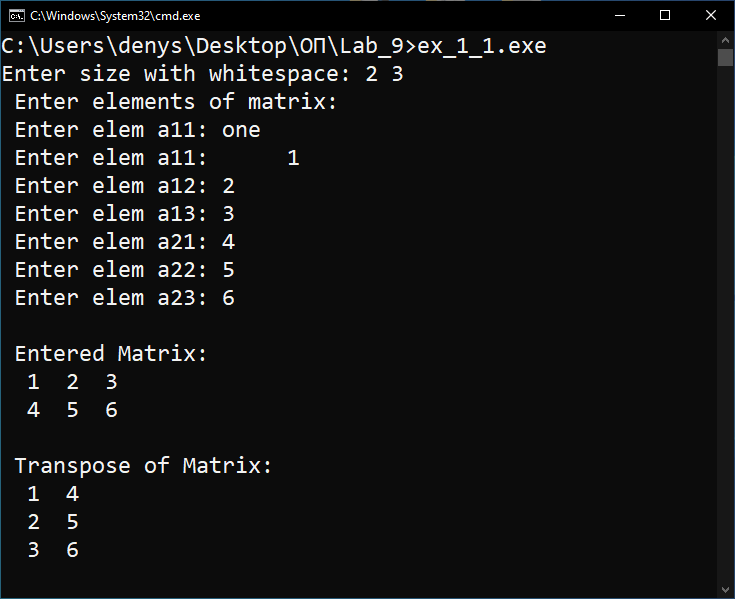
**}**

**putchar('\n');**

**}**

**}**

**Результати роботи програми (скриншот):**



**2)**

**Код програми:**

**#include <stdio.h>**

**void size\_entry(int \*h, int \*len);**

**void elem\_entry(int h, int len, int(\*matrix)[len]);**

**void show\_matrix(int h, int len, int(\*matrix)[len]);**

**int main(void) {**

**int h, len;**

**size\_entry(&h, &len);**

**int a[h][len],**

**b[h][len],**

**sum[h][len];**

**puts(" Enter elements of first matrix:");**

**elem\_entry(h, len, a);**

**puts(" Enter elements of second matrix:");**

**elem\_entry(h, len, b);**

**// adding corresponding elements of two arrays**

**for (int row = 0; row < h; row++) {**

**for (int col = 0; col < len; col++) {**

**\*(\*(sum + row) + col) = \*(\*(a + row) + col) + \*(\*(b + row) + col);**

**}**

**}**

**puts("\n Entered First Matrix:");**

**show\_matrix(h, len, a);**

**puts("\n Entered Second Matrix:");**

**show\_matrix(h, len, b);**

**puts("\n Sum Of Matrix:");**

**show\_matrix(h, len, sum);**

**return 0;**

**}**

**void size\_entry(int \*h, int \*len) {**

**do {**

**fseek(stdin, 0, SEEK\_END); // buffer cleaning**

**printf("Enter size with whitespace: ");**

**} while (scanf("%d %d", h, len) != 2); // until it's entered correctly**

**}**

**void elem\_entry(int h, int len, int(\*matrix)[len]) {**

**for (int row = 0; row < h; row++) {**

**for (int col = 0; col < len; col++) {**

**do {**

**fseek(stdin, 0, SEEK\_END); // buffer cleaning**

**printf(" Enter elem a%d%d: ", row + 1, col + 1);**

**} while (!scanf("%d[^\n]", \*(matrix + row) + col));**

**}**

**}**

**}**

**void show\_matrix(int h, int len, int(\*matrix)[len]) {**

**for (int row = 0; row < h; row++) {**

**for (int col = 0; col < len; col++) {**

**printf("%3d", \*(\*(matrix + row) + col));**

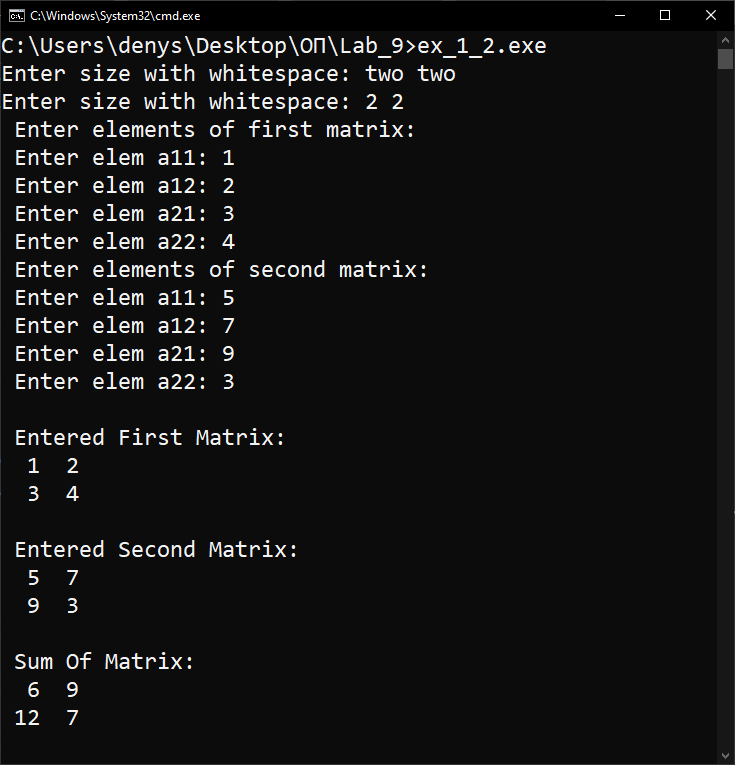
**}**

**putchar('\n');**

**}**

**}**

**Результати роботи програми (скриншот):**



**Завдання № 2:** Напишіть програму, в якій реалізуйте виконання поставлених задач з використанням вказівників.

1. На вхід в програму має подаватись рядок із букв і символів, після чого символи повинні бути видалені, а рядок виводитиметься на екран уже тільки з буквами.

2. З клавіатури зчитуватиметься декілька слів (враховуйте перевірки, щоб унеможливити введення цифр чи інших символів), після чого вони виводитимуться на екран у відсортованому в алфавітному порядку вигляді (можна використовувати введення тільки латинських літер).

**1)**

**Код програми:**

**#include <stdio.h>**

**#include <ctype.h>**

**#define SIZE 100**

**int main(void) {**

**char text[SIZE];**

**printf("Enter text: ");**

**gets(text);**

**int i = 0;**

**while (\*(text + i) != '\0') {**

**if (isdigit(\*(text + i)) || ispunct(\*(text + i))) {**

**for (int j = i; \*(text + j) != '\0'; j++) {**

**\*(text + j) = \*(text + j + 1);**

**}**

**i--;**

**}**

**i++;**

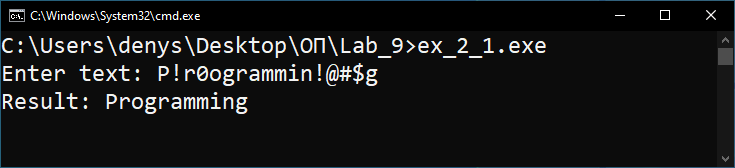
**}**

**printf("Result: %s\n", text);**

**return 0;**

**}**

**Результати роботи програми (скриншот):**



**2)**

**Код програми:**

**#include<stdio.h>**

**#include <string.h>**

**#include <ctype.h>**

**#define N 10**

**#define L 50**

**int isletters(char \*arr);**

**int main(void) {**

**char str[N][L], temp[L];**

**printf("Enter %d words:\n", N);**

**int k = 0;**

**while (k < N) {**

**scanf("%s[^\n]", \*(str + k));**

**if (!isletters(\*(str + k))) {**

**puts(" Wrong! Try again!");**

**k--;**

**}**

**k++;**

**}**

**for (int i = 0; i < N - 1; ++i) {**

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

**if (strcmp(\*(str + i), \*(str + j)) > 0) {**

**strcpy(temp, \*(str + i));**

**strcpy(\*(str + i), \*(str + j));**

**strcpy(\*(str + j), temp);**

**}**

**}**

**}**

**puts("\nIn lexicographical order:");**

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

**puts(\*(str + i));**

**}**

**return 0;**

**}**

**int isletters(char \*arr) {**

**for (int i = 0; \*(arr + i) != '\0'; i++) {**

**char elem = \*(arr + i);**

**if ((elem < 'a' || elem > 'z') && (elem < 'A' || elem > 'Z')) {**

**return 0;**

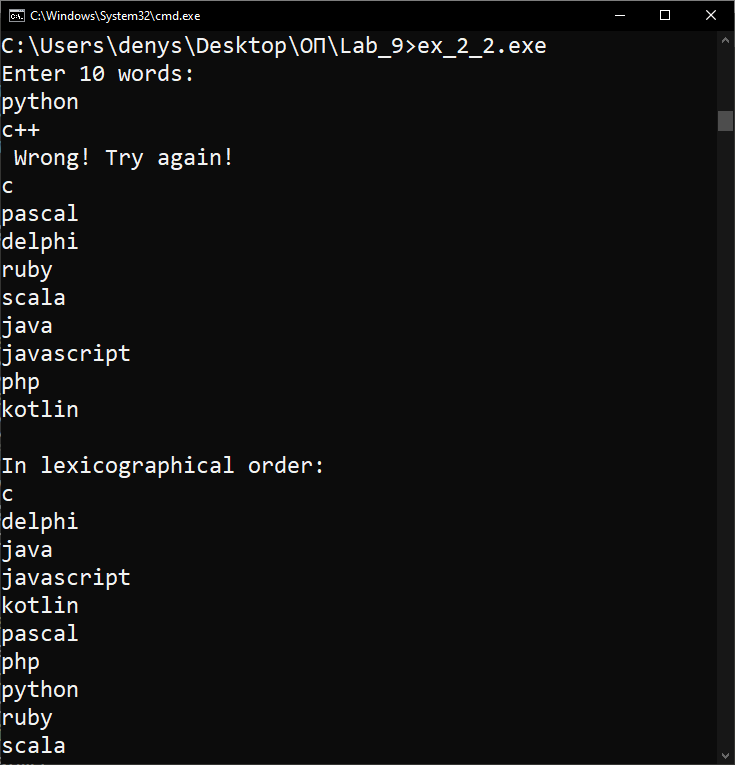
**}**

**}**

**return 1;**

**}**

**Результати роботи програми (скриншот):**



**Завдання № 3:** Напишіть програму, в якій на вхід буде прийматись рядок із декількох чисел (чисел має бути не менше 10, тільки цілочисельного типу), після чого реалізуйте сортування введеного масиву наступними методами в такому порядку:

1) алгоритм сортування методом Бульбашки (Bubble sort);

2) тасування чисел, використовуючи шафл (Shuffle method);

3) сортування методом Шелла (Shell sort);

4) знову перетасувати через шафл;

5) сортування методом Хоара (Hoor sort).

**Код програми:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <time.h>**

**#define SWAP(a, b) {int tmp; tmp=a; a=b; b=tmp;}**

**#define DIM 10**

**void bubble\_sort(int a[], int dim);**

**void shuffle(int a[], size\_t n);**

**void sort\_shell(int a[], int dim);**

**void sort\_hoor(int a[], int left, int right);**

**void showdata(int a[], int dim);**

**int main(void) {**

**int data[DIM];**

**int count = 0;**

**do { // until it's entered correctly**

**printf("Enter %d numbers: ", DIM);**

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

**count += scanf("%d[^\n]", data + i);**

**}**

**fseek(stdin, 0, SEEK\_END); // buffer cleaning**

**} while (count < 10);**

**puts(" Bubble sort");**

**bubble\_sort(data, DIM);**

**showdata(data, DIM);**

**puts(" Shuffle method");**

**shuffle(data, DIM);**

**showdata(data, DIM);**

**puts(" Shell sort");**

**sort\_shell(data, DIM);**

**showdata(data, DIM);**

**puts(" Shuffle method");**

**shuffle(data, DIM);**

**showdata(data, DIM);**

**puts(" Hoor sort");**

**sort\_hoor(data, 0, DIM - 1);**

**showdata(data, DIM);**

**return 0;**

**}**

**void bubble\_sort(int a[], int dim) {**

**for (int i = 0; i < dim - 1; i++) {**

**for (int j = 0; j < dim - i - 1; j++) {**

**if (\*(a + j) > \* (a + j + 1)) {**

**int temp = \*(a + j);**

**\*(a + j) = \*(a + j + 1);**

**\*(a + j + 1) = temp;**

**}**

**}**

**}**

**}**

**void shuffle(int a[], size\_t n) {**

**if (n > 1) {**

**srand((unsigned)time(NULL));**

**for (size\_t i = 0; i < n - 1; i++) {**

**size\_t j = i + rand() / (RAND\_MAX / (n - i) + 1);**

**int temp = \*(a + j);**

**\*(a + j) = \*(a + i);**

**\*(a + i) = temp;**

**}**

**}**

**}**

**void sort\_shell(int a[], int dim) {**

**for (int gap = dim / 2; gap > 0; gap /= 2) {**

**for (int i = gap; i < dim; i++) {**

**for (int j = i - gap; j >= 0 && \*(a + j) > \* (a + j + gap); j -= gap) {**

**SWAP(\*(a + j), \*(a + j + gap));**

**}**

**}**

**}**

**}**

**void sort\_hoor(int a[], int left, int right) {**

**int i = left, j = right, step = -1, condition = 1;**

**if (left >= right) return; // nothing to sort**

**do {**

**if (condition == (\*(a + i) > \* (a + j))) {**

**SWAP(\*(a + i), \*(a + j)); // swap values**

**SWAP(i, j); // swap indexes**

**step \*= -1;**

**condition = !condition;**

**}**

**j += step;**

**} while (j != i);**

**sort\_hoor(a, left, i - 1);**

**sort\_hoor(a, i + 1, right);**

**}**

**void showdata(int a[], int dim) {**

**printf("Data: ");**

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

**printf("%d ", \*(a + i));**

**}**

**putchar('\n');**

**}**

**Результати роботи програми (скриншот):**

