**ПРАКТИЧНЕ ЗАВДАННЯ № 7**

Напишіть програми, які будуть працювати, використовуючи різні шифри (алгоритми шифрування) тексту.

**Завдання № 1:** Шифр Цезаря

**Код програми:**

**#include <stdio.h>**

**#include <ctype.h> /\* islower, isupper \*/**

**void code\_letter(char arr[27], char ch, int code);**

**int main(void) {**

**char lower[27] = "abcdefghijklmnopqrstuvwxyz"; // with '\0'**

**char upper[27] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";**

**char ch;**

**int code;**

**printf("Enter code number: ");**

**scanf("%d", &code);**

**getchar(); // pass '\n'**

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

**ch = getchar();**

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

**while (ch != '\n') {**

**if (islower(ch)) {**

**code\_letter(lower, ch, code);**

**}**

**else if (isupper(ch)) {**

**code\_letter(upper, ch, code);**

**}**

**else { // numbers, space, symbols**

**putchar(ch);**

**}**

**ch = getchar();**

**}**

**printf("\n");**

**return 0;**

**}**

**void code\_letter(char arr[27], char ch, int code) {**

**int index;**

**for (int j = 0; j < 26; j++) { // letter search**

**if (arr[j] == ch) {**

**index = j + code;**

**if (index > 25) {**

**index -= 26;**

**}**

**putchar(arr[index]);**

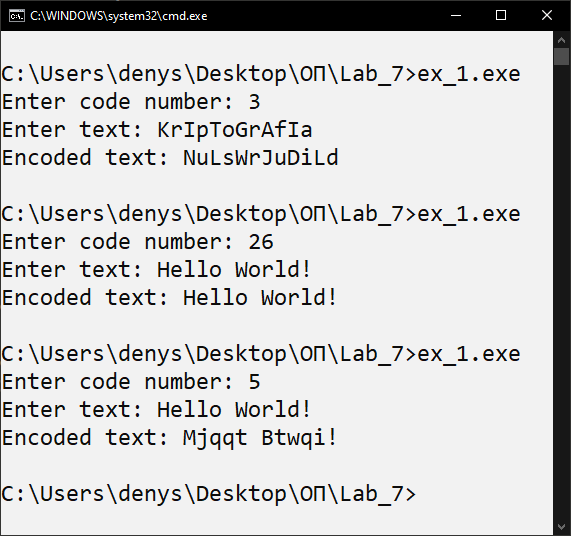
**break; // letter found**

**}**

**}**

**}**

**Результати роботи програми (скриншоти):**



**Завдання № 2:** Шифр Виженера

**Код програми:**

**#include <stdio.h>**

**#include <ctype.h> /\* tolower, islower, isupper \*/**

**#define MAX\_LEN 50 // code max length**

**void code\_letter(char arr[27], char ch, int code);**

**int main(void) {**

**char lower[27] = "abcdefghijklmnopqrstuvwxyz"; // with '\0'**

**char upper[27] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";**

**char ch;**

**char codex[MAX\_LEN];**

**int code; // current step in the codex**

**printf("Enter code word: ");**

**scanf("%s", codex);**

**getchar(); // pass '\n'**

**for (int i = 0; codex[i] != '\0'; i++) {**

**codex[i] = tolower(codex[i]);**

**for (int j = 0; j < 26; j++) {**

**if (lower[j] == codex[i]) {**

**codex[i] = j + 1; // letter -> code**

**break; // letter found**

**}**

**}**

**}**

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

**ch = getchar();**

**int k = 0;**

**code = codex[k];**

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

**while (ch != '\n') {**

**if (islower(ch)) {**

**code\_letter(lower, ch, code);**

**}**

**else if (isupper(ch)) {**

**code\_letter(upper, ch, code);**

**}**

**else { // numbers, space, symbols**

**putchar(ch);**

**}**

**// walk through the codex**

**k++;**

**if (codex[k] == '\0') {**

**k = 0;**

**}**

**code = codex[k];**

**ch = getchar();**

**}**

**printf("\n");**

**return 0;**

**}**

**void code\_letter(char arr[27], char ch, int code) {**

**int index;**

**for (int j = 0; j < 26; j++) { // letter search**

**if (arr[j] == ch) {**

**index = j + code;**

**if (index > 25) {**

**index -= 26;**

**}**

**putchar(arr[index]);**

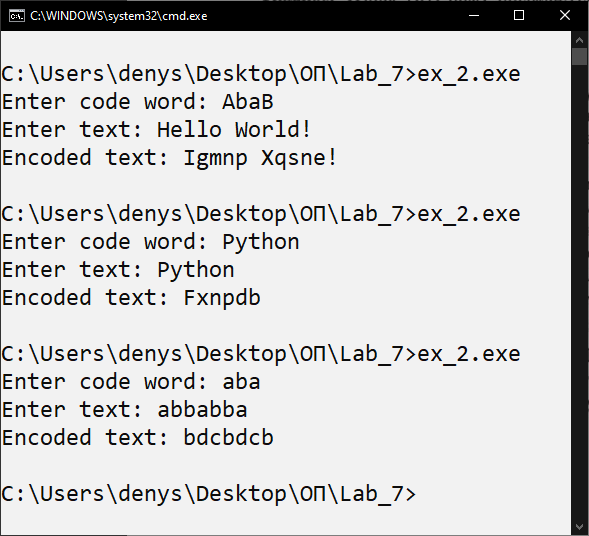
**break; // letter found**

**}**

**}**

**}**

**Результати роботи програми (скриншот):**



**Завдання № 3:** Шифр “Сциталь”

**Код програми:**

**#include <stdio.h>**

**#include <string.h>**

**#define MAX\_LEN 200 // text max length**

**int main(void) {**

**char text[MAX\_LEN];**

**int d, // diameter**

**len, // text length**

**index,**

**past = 0; // use the past value of the index**

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

**scanf("%d", &d);**

**getchar(); // pass '\n'**

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

**scanf("%[^\n]", text);**

**len = strlen(text);**

**char code\_text[len + 1]; // with '\0'**

**// fill the code\_text with null characters**

**for (int i = 0; i < len; i++)**

**code\_text[i] = '\0';**

**for (int k = 0; k < len; k++) { // move all characters**

**index = past + d;**

**if (index >= len)**

**index -= len;**

**while (code\_text[index] != '\0') { // meet the used place**

**index++;**

**if (index >= len)**

**index -= len;**

**}**

**code\_text[index] = text[k];**

**past = index;**

**}**

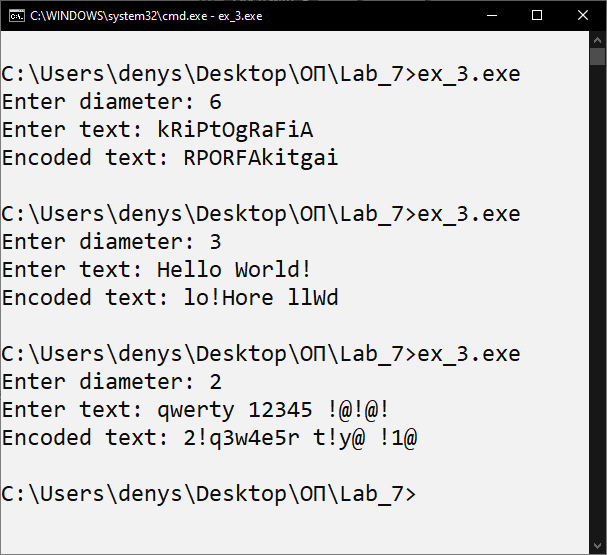
**code\_text[len] = '\0';**

**printf("Encoded text: %s\n", code\_text);**

**return 0;**

**}**

**Результати роботи програми:**



**Завдання № 4:** Напишіть програму, яка зберігатиме дуже довге значення числа. Наприклад факторіал числа 30.

**Код програми:**

**#include <stdio.h>**

**#include <ctype.h> /\* isdigit \*/**

**#include <stdbool.h>**

**#include <time.h> /\* clock \*/**

**#define DIMENSION 5 // the optimal dimension for a factorial from 0 to 10000**

**int main(void) {**

**int num;**

**printf("Enter an integer (from 0 to 10000): ");**

**scanf("%d", &num);**

**if (num < 0) {**

**puts("Error! Factorial of a negative number doesn't exist.");**

**}**

**else if (num == 0) {**

**puts("Factorial of 0 = 1");**

**}**

**else {**

**double time = clock();**

**int len = num \* DIMENSION; // space for digits in the array**

**int arr[len]; // the array for each individual digit**

**// fill the array with zeros**

**for (int i = 0; i < len; i++)**

**arr[i] = 0;**

**// split number into digits and put in the array**

**int copy\_num = num; // num will be changed**

**for (int p = len - 1; num; p--) {**

**arr[p] = num % 10;**

**num /= 10;**

**}**

**// multiplication in a column**

**for (int curr\_num = copy\_num - 1; curr\_num > 1; curr\_num--) {**

**int product, whole = 0, rest;**

**for (int j = len - 1; j > 0; j--) {**

**product = curr\_num \* arr[j] + whole;**

**whole = product / 10;**

**rest = product % 10;**

**arr[j] = rest;**

**}**

**}**

**// print the result without zeros**

**printf("Factorial of %d = ", copy\_num);**

**bool first\_natural = false; // met not zero**

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

**if (arr[k] != 0 && !first\_natural) { // meet first not zero digit**

**first\_natural = true;**

**}**

**if (first\_natural) {**

**printf("%d", arr[k]);**

**}**

**}**

**printf("\n");**

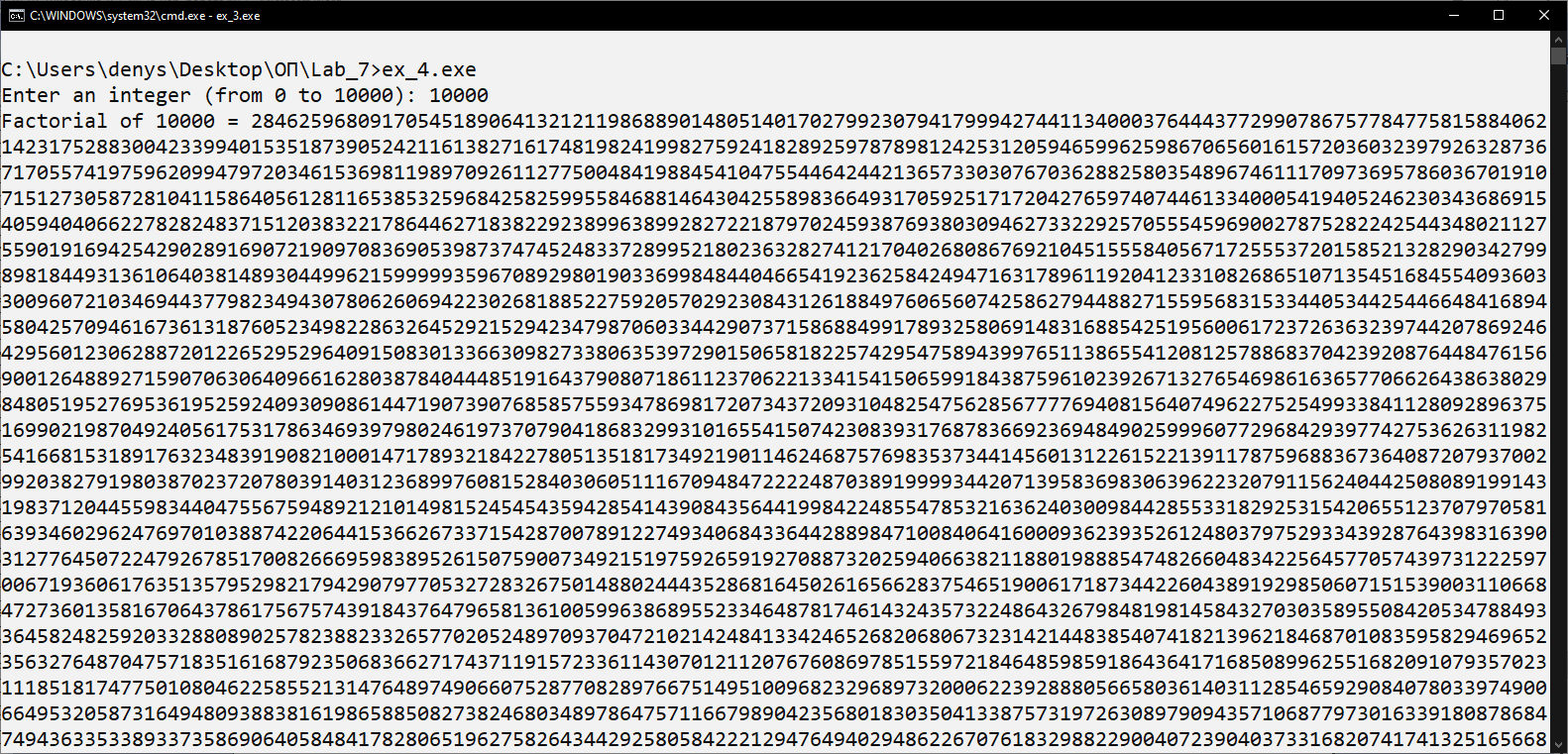
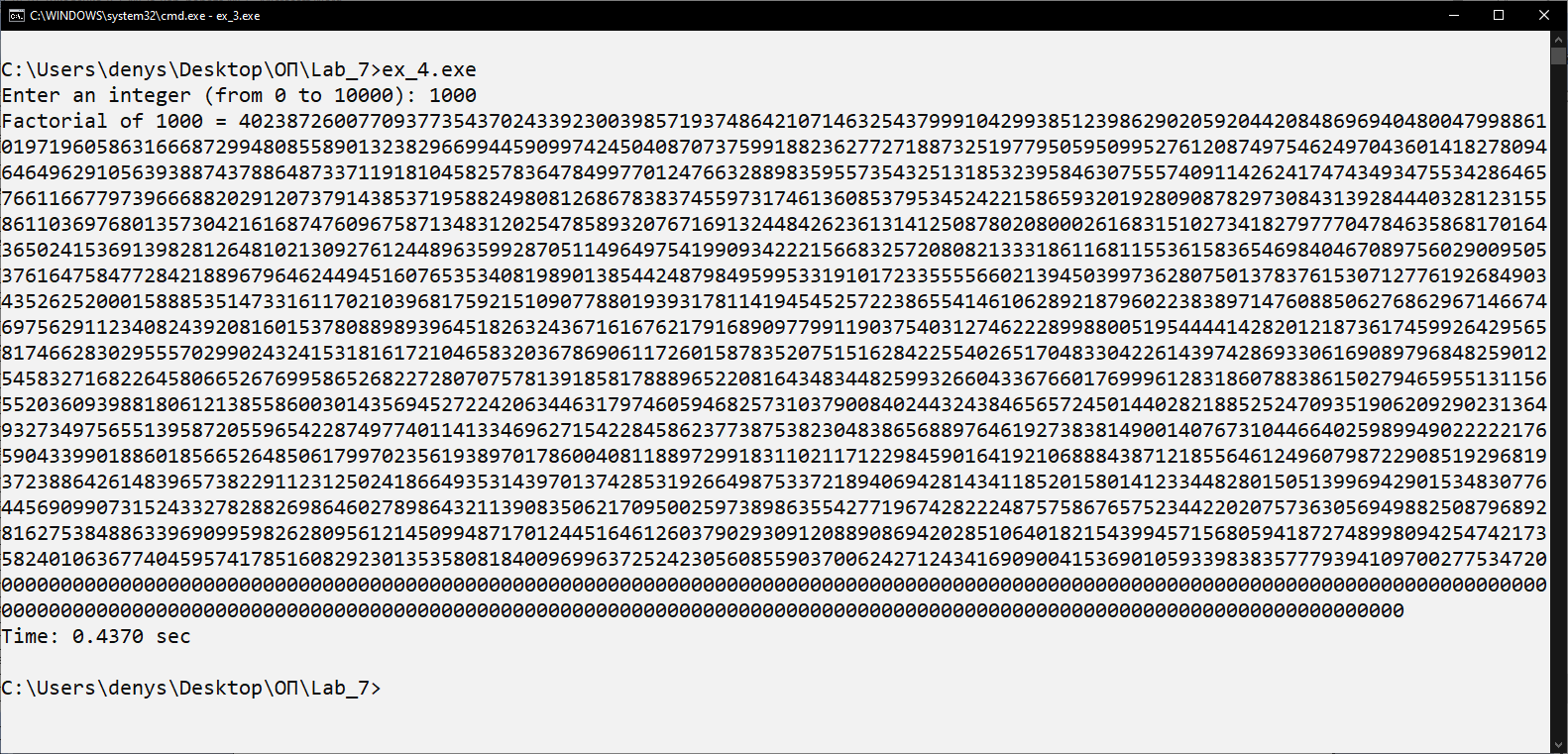
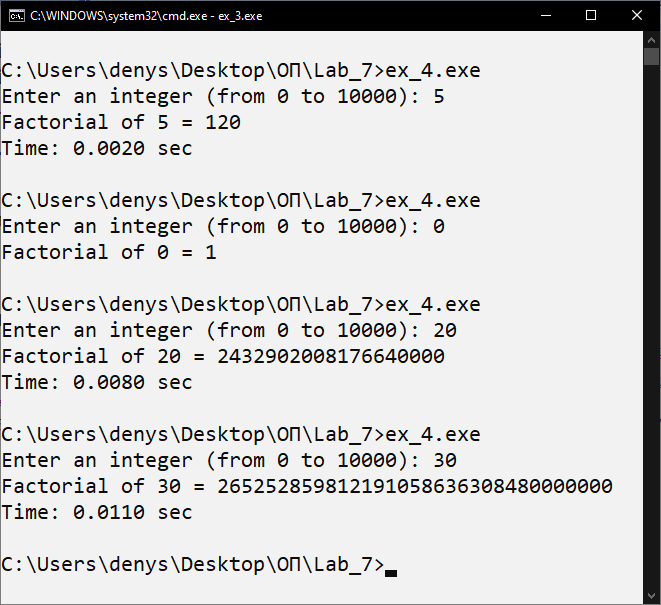
**printf("Time: %.4lf sec\n", (double)(clock() - time) / CLOCKS\_PER\_SEC);**

**}**

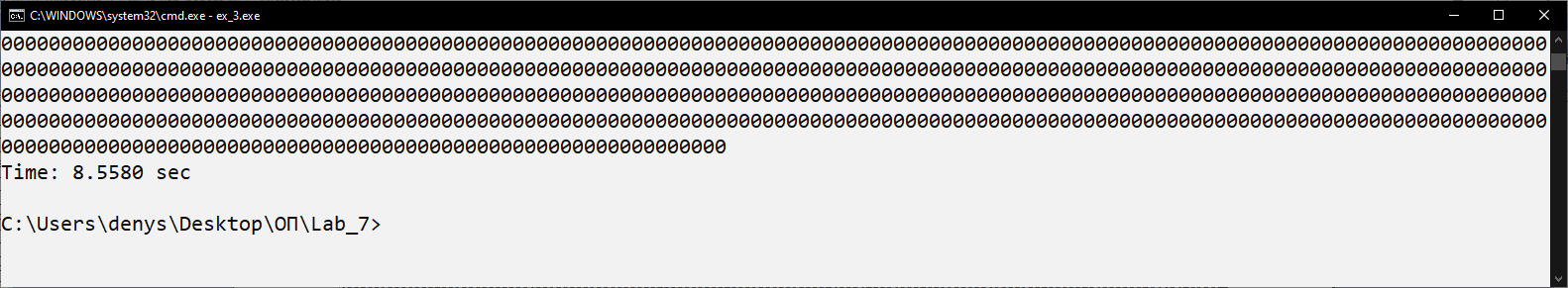
**return 0;**

**}**

**Результати роботи програми (скриншоти):**



**…**



**Коментар:**

Обчислений 1 000! джерело: <https://hi-news.ru/wp-content/uploads/2016/11/0-1j7qXnaj5JqMK78l-.jpg>

Обчислений 10 000! <https://image.winudf.com/v2/image/Y29tLmhtcHJvZHVjdGlvbnMuZmFjdG9yaWFsX3NjcmVlbl8wX3I1cGFvaXIx/screen-0.jpg?fakeurl=1&type=.jpg>

Даний алгоритм має малу часову складність (час виконання виводиться у кінці). Попередження про величину числа до 10 000 – умовне, програма може обчислити і більше.