**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу  
Кафедра системного проектування**

**Звіт**

**про виконання лабораторної роботи №12   
з дисципліни “Алгоритми та програмування”**

Виконав:  
студент I курсу, групи ДА-12  
Кракович Павло Дмитрович

Прийняв:

к.т.н., доцент Безносик О. Ю.

Київ – 2022

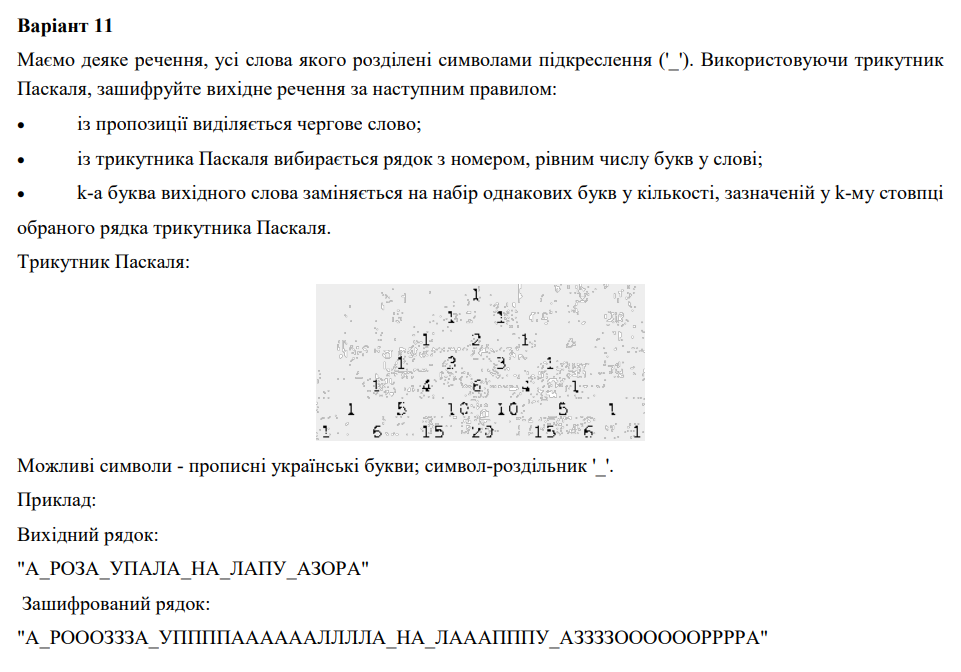
1. Скласти алгоритм і програму рішення запропонованої задачі (рис. 1)

Рис. 1

1. Аналіз умови задачі.

Для успішного виконання завдання потрібно згенерувати значення трикутника паскаля для кожної букви у слові та потім створити новий рядок циклом та за допомогою значень, які ми отримали з трикутника паскаля повторювати букви потрібну кількість разів, щоб отримати зашифрований рядок.

1. Код (рис. 2)

#include <iostream>  
  
using namespace std;  
  
void triangleOfPascal(int tail, int head, int \*arr, int &position);  
void fillEncodedString (int \*arr,string& str,string& encodedString,int lengthOfString);  
void encode(string& str,string& encodedString,int lengthOfString);  
void print(string encodedString);  
  
int main()  
{  
 string str,encodedString;  
 cout << "Type text to encode" << endl;  
 cin >> str;  
 int lengthOfString = str.length();  
 encode(str,encodedString,lengthOfString);  
 cout << "Encoded text:" << endl;  
 print(encodedString);  
}  
  
void triangleOfPascal(int tail, int head, int \*arr, int &position) {  
 int x = 1;  
 for (int k = 0; k <= tail-head-1; k++) {  
 arr[position]=x;  
 x = x \* ((tail-head-1) - k) / (k + 1);  
 position++;  
 }  
 arr[position]=0;  
 position++;  
  
 // Генерую трикутник паскаля та зберігаю його значення у масив.  
 // Основа ідея - Для кожного символу у строці, за індексом я паралельно зберігаю кількість повторів, які потрібно буде зробити для  
 // кодування  
}  
  
void encode(string& str,string& encodedString,int lengthOfString) {  
 int head=0,tail=0,position = 0;  
 int arr[100];  
  
 for (int i=0; i <= lengthOfString; i++) {  
 if (str[i]=='\_' || i == lengthOfString) {  
 triangleOfPascal(tail,head,arr,position);  
 head = i+1;  
 tail++;  
 }  
 else {  
 tail++;  
 }  
 }  
 // Тут я пройшов по всій строці та вже сгенерував дзеркальний масив чисел, з яким буду працювати  
  
 fillEncodedString(arr,str,encodedString,lengthOfString); // Викликаю функцію для заповнення нової строки  
}  
  
void fillEncodedString (int\* arr,string& str,string& encodedString,int lengthOfString) {  
 for (int i=0; i < lengthOfString; i++) {  
 if (arr[i]==0) {  
 encodedString.push\_back('\_');  
 }  
 for (int j=0; j < arr[i]; j++) {  
 encodedString.push\_back(str[i]);  
 }  
 } // Заповнив нову строку  
}  
  
void print(string encodedString) {  
 int lengthOfEncodedString = encodedString.length();  
 for (int i=0; i < lengthOfEncodedString; i++) {  
 cout << encodedString[i];  
 }  
 // Друкую нову строку із зашифрованим текстом  
}

Рис. 2

1. Результат роботи програми: (рис. 3)

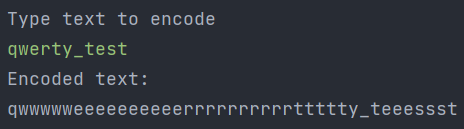


Рис. 7

Рис. 3

Висновок

В ході виконання лабораторної роботи №12 я створив функцію, яка отримує значення «трикутника Паскаля» для кожної букви у рядку, який буду шифрувати. Потім, я записав ці значення в окремий масив, з яким буду працювати. Останнім кроком шифрування я створив цикл, який проходитиме по кожному символу оригінального рядка та за допомогою функції push\_back() з бібліотеки string зміг облегшити собі роботу та не додавати зайву логіку заповнення нового рядка. На кожному кроці потрібний символ додається до нового рядка.

Виконуючи цю роботу, я покращив свої навички роботи з рядками, вивчив синтаксис і семантику стандартних методів по роботі з рядками.