**Report:**

Caesar cipher: <https://www.youtube.com/watch?v=Y1gZ7z3X-po>

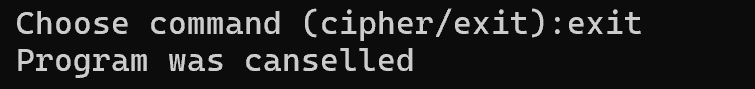
About static library: <https://www.geeksforgeeks.org/how-to-create-static-library-in-cpp/>

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис



Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

//коли ми в run time, що з цього коду маємо зробити, коли створюємо бібліотеку і хочемо використати з нею якусь ф-ю:  
//крок 1: знайти цю бібліотеку (завантажити бібліотеку, яка лежить в певному місці (ЧАС!)  
//крок 2: знайти ф-ю і завантажити її з бібліотеки  
//крок 3: викликати цю ф-ю  
//крок 4: вивантажуємо бібліотеку назад, коли ми використали бібліотеку і вона нам більше не потрібна в оперативній пам'яті (ЧАС!)

The explanation in another example:

#include <windows.h>//використовуємо платформно залежний код  
using namespace std;  
  
int main()  
{  
 //крок 1  
 HINSTANCE handle = LoadLibrary(TEXT("C:\\KSE\\Programming\_paradigms\\Additional\\swap.dll"));//завантажуємо бібліотеку і вказуємо її назву, яку будуть шукати в тій директорії, де зараз знаходиться executable file  
 //якщо в різних бібліотеках, то можемо поставити відносний\абсолютний шлях  
 //handle це бібліотека, яку завантажили; (handle, "swap") -- з цієї бібліотеки завантажити цю ф-ю  
 //(TEXT("swap.dll") -- кастить до типу, який by default has 2 bytes на 1 символ ---- локалізація?; альтернатива TEXT це L ------- змінюємо к-сть байт, які відповідають за один символ  
 //HINSTANCE -- тип даних Windows  
 if (handle == nullptr || handle == INVALID\_HANDLE\_VALUE) //якщо handle invalid (бібліотека не існує\неправильного формату)  
 {  
 cout << "Lib not found" << endl;  
 return -1;  
 }  
  
 //крок 1:  
 typedef void (\*function\_ptr)(string, char, char);//вказівник на ф-ю, який потім можемо використовувати, як виклик ф-ї  
 //спеціальний тип -- typedef; function\_ptr вказуватиме на ф-ю, яка вертає void та приймає (string, char, char)  
  
 //крок 2:  
 //створюємо змінну -- вказівник на ф-ю func; GetProcAddress -- завантажуємо ф-ю з бібліотеки. для цього маємо знати назву ф-ї як вона називається в бібліотеці  
 function\_ptr func = (function\_ptr)GetProcAddress(handle, "swap");  
 //(function\_ptr) -- cast  
 if (func == nullptr)//якщо з якоїсь причини не знайшли  
 {  
 cout << "Proc not found" << endl;  
 return -1;  
 }  
  
 //крок 3:  
 func("aabbccc", 'a', 'b');//користуємося цією ф-єю як зазвичай  
  
 //крок 4:  
 FreeLibrary(handle);  
  
 return 0;  
}

About static library: <https://www.geeksforgeeks.org/how-to-create-static-library-in-cpp/>

<https://medium.com/@dreamanip/c-static-library-and-dynamic-library-d7803ae8eef5>

<https://domiyanyue.medium.com/c-development-tutorial-4-static-and-dynamic-libraries-7b537656163e>

Code in console to compite dynamic library:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

About header: <https://www.geeksforgeeks.org/header-files-in-c-cpp-and-its-uses/>

#ifndef YOUR\_NAME\_INCLUDE

#define YOUR\_NAME\_INCLUDE

/\* Your function statement here \*/

#endif