

# Кодировки

# Задание

Дан файл `Lopatin.txt` с орфографическим словарем русского языка. Найдите в этом файле все слова, которые имеют заданную длину (это значение пользователь задает в консоли) и содержат две гласные. Результат запишите в файл `result.txt`.

# Кодировки в Windows

## ANSI

- 1 байт на символ
- мультибайтовая (многобайтовая) кодировка
- позволяет работать с латиницей
- тип **char**

## Unicode

- 2 байта на символ
- позволяет работать с символами разных языков мира
- тип **wchar\_t**
- перед строкой нужен префикс **L**

# Типы для строк

<b>TCHAR</b>	<code>char</code> или <code>wchar_t</code>	символ; зависит от настроек проекта
<b>TCHAR*</b>	<code>char*</code> или <code>wchar_t*</code>	указатель на строку; зависит от настроек проекта
<b>LPSTR</b>	<code>char*</code>	указатель на ANSI-строку
<b>LPCSTR</b>	<code>const char*</code>	указатель на константную ANSI-строку
<b>LPWSTR</b>	<code>wchar_t*</code>	указатель на Unicode-строку
<b>LPCWSTR</b>	<code>const wchar_t*</code>	указатель на константную Unicode-строку
<b>LPTSTR</b>	<code>char*</code> или <code>wchar_t*</code>	указатель на строку; зависит от настроек проекта
<b>LPCTSTR</b>	<code>const char*</code> или <code>const wchar_t*</code>	указатель на константную строку; зависит от настроек проекта

**LP** – длинный указатель

**C** – константа

**W** – Unicode

**T** – тип **TCHAR**

# Функции для работы со строками `wchar_t`

- тип `wchar_t` – функции с префиксом **WCS**:  
`wcsncpy`, `wcslen`, `wcscat` и др.

```
void f(){  
    wchar_t* ws1 = L"One. ";  
    wchar_t* ws2 = L"Two. ";  
    wchar_t* ws3 = L"Three";  
    const size_t len = wcslen(ws1) + wcslen(ws2) + wcslen(ws3) + 1;  
    wchar_t* wsres = new wchar_t[len];  
    wcscpy(wsres, ws1);  
    wcscat(wsres, ws2);  
    wcscat(wsres, ws3);  
    ...  
}
```

# Функции для работы со строками TCHAR

- функции начинаются с префикса **\_tcs**:  
\_tcscpy, \_tcslen, \_tcscat и др.

```
void f(){  
    TCHAR ts1[] = L"One. ";  
    TCHAR ts2[] = L"Two. ";  
    TCHAR ts3[] = L"Three";  
    const size_t len = _tcslen(ts1) + _tcslen(ts2) + _tcslen(ts3) + 1;  
    TCHAR* tsres = new TCHAR[len];  
    _tcscpy(tsres, ts1);  
    _tcscat(tsres, ts2);  
    _tcscat(tsres, ts3);  
    ...  
}
```

# Макросы для TCHAR

- описаны в файле "tchar.h"

```
#ifndef _UNICODE
#define _tcslen wcslen
#else
#define _tcslen strlen
#endif
```

```
#define _T(x)    __T(x) // строка зависит от компиляции
#define _TEXT(x) __T(x) // строка зависит от компиляции
```

```
TCHAR* s = _TEXT("Какая-то строка");
MessageBox(NULL, s, _TEXT("Example"), MB_ICONERROR);
```

# Конвертация строк (1)

```
#include "stdlib.h"
size_t wcstombs(char* outstr, const wchar_t* instr, size_t n);
```

- конвертирует первые n символов строки из Unicode в ANSI (нужно включать '\0')

```
void f(){
    setlocale(LC_ALL, "Russian");
    wchar_t str1[] = L"Строка";
    size_t len = wcstombs(NULL, str1, 0) + 1;
    char* str2 = new char[len];
    int count = wcstombs(str2, str1, len);
    // ...
    delete[] str2;
}
```



# Конвертация строк (2)

```
#include "stdlib.h"
```

```
size_t mbstowcs(wchar_t* outstr, const char* instr, size_t n);
```

- конвертирует первые n символов строки из ANSI в Unicode (нужно включать '\0')

```
void f(){  
    setlocale(LC_ALL, "Russian");  
    char str1[] = "Строка";  
    size_t len = mbstowcs(NULL, str1, 0) + 1;  
    wchar_t* str2 = new wchar_t[len];  
    int count = mbstowcs(str2, str1, len);  
    ...  
    delete[] str2;  
}
```

# Конвертация строк (3)

- **MultiByteToWideChar()** – в Unicode
- **WideCharToMultiByte()** – из Unicode
- функции, специфичные для Windows

```
int MultiByteToWideChar(  
    UINT CodePage,  
    DWORD dwFlags,  
    LPCSTR lpMultiByteStr,  
    int cbMultiByte,  
    LPWSTR lpWideCharStr,  
    int cchWideChar  
);
```

```
int WideCharToMultiByte(  
    UINT CodePage,  
    DWORD wFlags,  
    LPCWSTR lpWideCharStr,  
    int cchWideChar,  
    LPSTR lpMultiByteStr,  
    int cbMultiByte,  
    LPCSTR lpDefaultChar,  
    LPBOOL lpUsedDefaultChar );
```

# Пример

```
void f() {  
    char str[] = "My new string";  
    size_t len = MultiByteToWideChar(CP_ACP, 0, str, -1, NULL, 0);  
    wchar_t* str1 = new wchar_t[len];  
    MultiByteToWideChar(CP_ACP, 0, str, -1, str1, len);  
    MessageBox(NULL, str1, L"Example 3", MB_ICONERROR);  
    delete[] str1;  
  
    wchar_t str2[] = L"И еще одна строка";  
    len = WideCharToMultiByte(CP_ACP, 0, str2, -1, NULL, 0,  
                               0, 0);  
  
    char* str3 = new char[len];  
    WideCharToMultiByte(CP_ACP, 0, str2, -1, str3, len, 0, 0);  
    delete[] str3;  
}
```

**Вопросы?**