Приложение 1. Тесты.

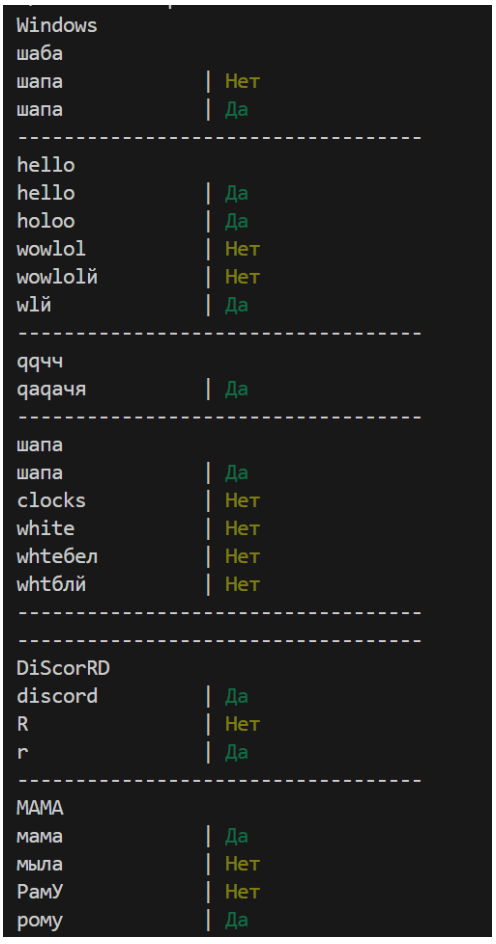
Входной файл “input.txt”. (Содержание):

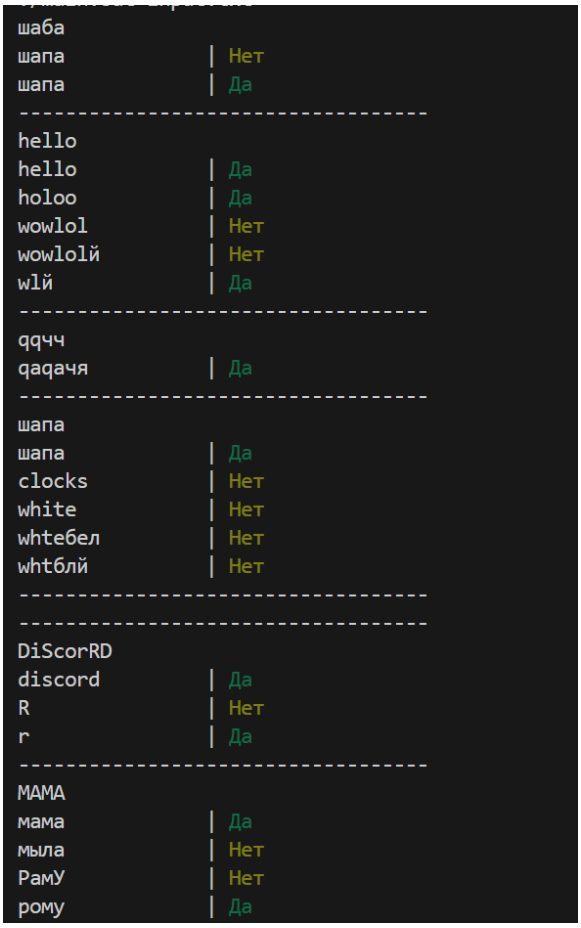
шаба шапа шапа
hello hello holoo wowlol wowlolй wlй
qqчч qaqачя
шапа шапа clocks white whтебел whтблй

DiScorRD discord R r
МАМА мама мыла РамУ рому

Тест.

Windows PowerShell	Linux Bash
>>./main.out input.txt <<	>>./main.out input.txt <<





Полный код программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <wchar.h>
#include <locale.h>

#if defined(_WIN64)
 /* Microsoft Windows (64-bit) */
 #include <windows.h>
#endif

int isRu(wchar_t ch) {
 return (ch >= L'А' && ch <= L'я');
}
int isEn(wchar_t ch) {
 return (ch >= L'A' && ch <= L'z');
}
int isUpper(wchar_t ch) {
 return ((ch >= L'А' && ch <= L'Я') || (ch >= L'A' && ch <= L'Z'));
}

void printBits(int set, int bitsN) {
 for(int i = 0; i < bitsN; ++i) {
  wprintf(L"%d", (set >> (bitsN - i - 1)) & 1);
 }
 wprintf(L"\n");
}

unsigned int indexOfRu(wchar_t c) {
 return c - L'а';
}

unsigned int indexOfEn(wchar_t c) {
 return c - L'a';
}

int isInSet(int set, unsigned long long index) {
 if(index > sizeof(int) * 8 - 1)
  return 0;

 return set & (1 << index);
}

int addToSet(int set, unsigned long long index) {
 if(index > sizeof(int) * 8 - 1)
  return set;

 return set | (1 << index);
}
```

```
int createSet(const wchar_t * s) {
  int result = 0;
  for(size_t i = 0; s[i] != '\0'; ++i)
    result = addToSet(result, s[i] - 'a');

  return result;
}

int createSetRu(const wchar_t * s) {
  int result = 0;
  for(size_t i = 0; s[i] != '\0'; ++i)
    result = addToSet(result, s[i] - L'a');

  return result;
}

int main(int argc, char * argv[]) {
  setlocale(LC_ALL, "");
  #if defined(_WIN64)
    setlocale(LC_ALL, "en_US.UTF-8");
    SetConsoleOutputCP(CP_UTF8);
    SetConsoleCP(CP_UTF8);
    printf("Windows\n");
  #endif

  assert(("Error 1: input file name not specified", argc > 1));

  const char * inputFileName = argv[1];
  FILE * file = fopen(inputFileName, "r");
  assert(("Error 2: can't open file", file != NULL));

  // wprintf(L"zyxwvutsrqponmlkjihgfedcba\n");
  const wchar_t* vowels = L"aeiouy";
  int vowelSet = createSet(vowels);
  int consonantsSet = ~vowelSet;

  // wprintf(L"яюэьыъщшчцхфутсрпонмлкйизжедгвба\n");
  const wchar_t* vowels_ru = L"аоуыэеёиюя";
  int vowelSetRu = createSetRu(vowels_ru);
  int consonantsSetRu = ~vowelSetRu;

  wchar_t c;
  wchar_t c_lower;
  int wordSetRu = 0;
  int wordSetEn = 0;
  int oldWordSetRu = 0;
  int oldWordSetEn = 0;
  size_t CURRERCT_ITER;
  size_t MAX_ITER = 10000;
```

```
  while (CURRERCT_ITER < MAX_ITER) {
    c = fgetwc(file);

    if (isRu(c)) {
      if (isUpper(c)) {
        c_lower = c + 32;
      } else {
        c_lower = c;
      }
      wordSetRu = addToSet(wordSetRu, indexOfRu(c_lower));
      wprintf(L"%lc", c);

    } else if (isEn(c)) {
      if (isUpper(c)) {
        c_lower = c + 32;
      } else {
        c_lower = c;
      }
      wordSetEn = addToSet(wordSetEn, indexOfEn(c_lower));
      wprintf(L"%lc", c);

    } else if ((wordSetRu || wordSetEn) || (c == WEOF)) {
      // wprintf(L"\n"); printBits(oldWordSetEn, 26); printBits(oldWordSetRu, 32);
      if ((oldWordSetRu || oldWordSetEn) && ((wordSetRu & consonantsSetRu) == oldWordSetRu) &&
((wordSetEn & consonantsSet) == oldWordSetEn)){
        wprintf(L"\t\t| \e[2;32mДа\e[0;0m");
      } else if ((oldWordSetRu || oldWordSetEn) && (wordSetEn || wordSetRu)) {
        wprintf(L"\t\t| \e[2;33mНет\e[0;0m");
      }
      oldWordSetRu = wordSetRu & consonantsSetRu;
      oldWordSetEn = wordSetEn & consonantsSet;
      wordSetRu = 0;
      wordSetEn = 0;
      wprintf(L"\n");
    }

    if (c == L'\n'){
      oldWordSetEn = 0;
      oldWordSetRu = 0;
      wprintf(L"----------------------------------\n");
    }

    if (c == WEOF){
      break;
    }
    ++CURRERCT_ITER;
  }
  return 0;
}
```