

```
// FeliCaLibSample11.cpp : コンソール アプリケーションのエントリ ポイントを定義します。  
// サービスコードのリストを取得
```

```
#include "stdafx.h"  
#include <stdio>  
#include <stdlib>  
#include "felica.h"
```

```
void print_vector(char* title, unsigned char* vector, int length);
```

```
int main(void)
```

```
{  
    fprintf(stdout, "サービスコードのリストを取得\n\n");  
  
    /* ライブラリの初期化 */  
    if (!initialize_library()) {  
        fprintf(stderr, "ライブラリの初期化に失敗しました。 \n\n");  
        return EXIT_FAILURE;  
    }  
  
    /* リーダ・ライタの自動認識とオープン */  
    if (!open_reader_writer_auto()) {  
        fprintf(stderr, "リーダ・ライタのオープンに失敗しました。 \n\n");  
        return EXIT_FAILURE;  
    }  
  
    /* 変数の定義と初期化 */  
    structure_polling polling;  
    unsigned char system_code[2] = {0x00, 0x00};  
    polling.system_code = system_code;  
    polling.time_slot = 0x00;  
  
    input_structure_search_service_code input_search_service_code;  
    input_search_service_code.buffer_size_of_area_codes = 30;  
    input_search_service_code.buffer_size_of_service_codes = 30;  
    input_search_service_code.offset_of_area_service_index = 0;  
  
    structure_card_information card_information;  
    unsigned char card_idm[8];  
    unsigned char card_pmm[8];  
    card_information.card_idm = card_idm;  
    card_information.card_pmm = card_pmm;  
  
    output_structure_search_service_code output_search_service_code;  
    unsigned char service_code_list[2*30];  
    unsigned char area_code_list[2*30];  
    unsigned char end_service_code_list[2*30];  
    output_search_service_code.service_code_list = service_code_list;  
    output_search_service_code.area_code_list = area_code_list;  
    output_search_service_code.end_service_code_list = end_service_code_list;
```

```

/* ポーリングとサービスコードのリストを取得 */
if (!polling_and_search_service_code(&polling, &input_search_service_code,
                                     &card_information, &output_search_service_code)) {
    fprintf(stderr, "FeliCaカードが見つかりません。¥n¥n");
    return EXIT_FAILURE;
}

/* 取得サービスコードの表示 */
fprintf(stdout, "number of service codes: %ld¥n", output_search_service_code.number_of_service_codes);
print_vector("service code list:", service_code_list, 2*output_search_service_code.number_of_service_codes);

/* 取得エリアコードの表示 */
fprintf(stdout, "number of area codes: %ld¥n", output_search_service_code.number_of_area_codes);
print_vector("area code list:", area_code_list, 2*output_search_service_code.number_of_area_codes);

/* 取得サービスコードの表示 */
fprintf(stdout, "number of end service codes: %ld¥n", output_search_service_code.number_of_area_codes);
print_vector("end service code list:", end_service_code_list,
            2*output_search_service_code.number_of_area_codes);

/* リーダ・ライタのクローズ */
if (!close_reader_writer()) {
    fprintf(stderr, "リーダー・ライタのクローズに失敗しました。¥n¥n");
    return EXIT_FAILURE;
}

/* ライブラリの解放 */
if (!dispose_library()) {
    fprintf(stderr, "ライブラリの解放に失敗しました。¥n¥n");
    return EXIT_FAILURE;
}

fprintf(stdout, "¥nプログラムの実行を終了します。¥n¥n");

return EXIT_SUCCESS; // 正常終了を示す戻り値
}

void print_vector(char* title, unsigned char* vector, int length)
{
    if (title != NULL) {
        fprintf(stdout, "%s ", title);
    }

    int i;
    for (i = 0; i < length - 1; i++) {
        fprintf(stdout, "%02x ", vector[i]);
    }
    fprintf(stdout, "%02x", vector[length - 1]);
    fprintf(stdout, "¥n");
}

```