

Operating Systems(24802)

Project #2

Division of Computer Science

2017011912

Nam jung hoon

Contents

1. Description of the function

2. Program source file

3. Compilation Process

4. Outputs and Descriptions

<Description of the function>

Function #1 : Check_rows

Valid 한 rows 의 결정 조건은 각 행에 1,2,3,4,5,6,7,8,9 가 각각 하나씩 들어있는 것.

9 번 반복하는 반복문을 사용해 각 행이 Valid 한가? 를 판단.

정수형 배열 check_data[10]={0,1,2,3,4,5,6,7,8,9} 를 선언해두고,

9 번 반복하는 반복문을 사용해 각 행의 원소(sudoku[i][j])들을 탐색.

각 원소들을 check_now 에 저장 후 check_data[check_now]의 값을 0 으로 초기화.

-> 모든 숫자가 다 들어있다면, check_data 배열은 {0,0,0,0,0,0,0,0,0,0}이 됨.

-> check_data 의 모든 원소를 더한 값(check_sum)이 0 이어야함.

Check_sum 이 0 이 아니라면 그 행이 Valid 하지 않다는 뜻이므로 valid[o][i]의 값을 0,

그렇지 않다면 그 행이 Valid 하다는 뜻이므로 valid[o][i]의 값을 1로 바꿈.

마지막으로 pthread_exit(NULL)을 이용해 스레드를 종료.

Function #2 : Check_columns

Valid 한 cloumns 의 결정 조건은 각 열에 1,2,3,4,5,6,7,8,9 가 각각 하나씩 들어있는 것.

9 번 반복하는 반복문을 사용해 각 열이 Valid 한가? 를 판단.

정수형 배열 check_data[10]={0,1,2,3,4,5,6,7,8,9} 를 선언해두고,

9 번 반복하는 반복문을 사용해 각 열의 원소(sudoku[j][i])들을 탐색.

각 원소들을 check_now 에 저장 후 check_data[check_now]의 값을 0 으로 초기화.

-> 모든 숫자가 다 들어있다면, check_data 배열은 {0,0,0,0,0,0,0,0,0,0}이 됨.

-> check_data 의 모든 원소를 더한 값(check_sum)이 0 이어야함.

Check_sum 이 0 이 아니라면 그 열이 Valid 하지 않다는 뜻이므로 valid[1][i]의 값을 0,

그렇지 않다면 그 열이 Valid 하다는 뜻이므로 valid[1][i]의 값을 1로 바꿈.

마지막으로 pthread_exit(NULL)을 이용해 스레드를 종료.

Function #3 : Check_subgrid

Valid 한 subgrid 의 결정조건도 Valid 한 행과 열의 결정조건과 동일.

변수 k 는 인자로 받은 arg (서브그리드 번호)의 정수형.

K 번째 서브그리드는 $(k/3)*3 \sim (k/3)*3+2$ by $(k\%3)*3 \sim (k\%3)*3+2$.

이중 반복문을 사용해 서브그리드의 모든 원소(sudoku[i][j])를 탐색 후

행과 열의 판단 과정과 같이 check_data(0,1,2,3,4,5,6,7,8,9)의 합인 check_sum 이 0 인지 확인.

Check_sum 이 0 이 아니라면 Valid 하지 않다는 뜻이므로 valid[2][k]의 값을 0,

그렇지 않다면 그 subgrid 가 Valid 하다는 뜻이므로 valid[2][k]의 값을 1로 바꿈.

마지막으로 pthread_exit(NULL)을 이용해 스레드를 종료.

Function #4 : check_sudoku

스도쿠의 현재 상태를 이중 반복문을 사용해 출력.

스도쿠가 valid 한지 판단할 각 n 개의 스레드의 ID 를 저장할 배열 th_id[n]선언.(0~8 : subgrid, 9:rows, 10:columns)

각 행들이 Valid 한지 판단하기 위해 th_id[9]에 해당하는 스레드를 pthread_create 로 생성 후, 에러 판별 및 처리.

각 열들이 Valid 한지 판단하기 위해 th_id[10]에 해당하는 스레드를 생성 후, 에러 판별 및 처리.

서브그리드 판별 스레드를 생성할때, 스레드의 번호를 check_subgrid 에 넘겨주기 위한 배열 subgrid_num 을 선언 후 인자로 사용.

각 서브그리드가 Valid 한지 판단하기 위해 th_id[0]~th_id[8]에 해당하는 9 개의 스레드를 생성 후, 에러 판별 및 처리.

위에서 생성한 n 개의 스레드의 실행이 종료될 때 까지 pthread_join(th_id[i],NULL)을 사용해 기다림.

Valid 배열에 저장된 현재 스도쿠의 판별 상태를 행, 열, 서브그리드 순으로 출력.

<Program source file>

```
/*
 * Copyright 2021. Heekuck Oh, all rights reserved
 * 이 프로그램은 한양대학교 ERICA 소프트웨어학부 재학생을 위한 교육용으로 제작되었습니다.
 */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <pthread.h>

/*
 * 기본 스도쿠 퍼즐
 */
int sudoku[9][9] =
{{6,3,9,8,4,1,2,7,5},{7,2,4,9,5,3,1,6,8},{1,8,5,7,2,6,3,9,4},{2,5,6,1,3,7,4,8,9},{4
,9,1,5,8,2,6,3,7},{8,7,3,4,6,9,5,2,1},{5,4,2,3,9,8,7,1,6},{3,1,8,6,7,5,9,4,2},{9,6,
7,2,1,4,8,5,3}};

/*
 * valid[0][0], valid[0][1], ..., valid[0][8]: 각 행이 올바른지 1, 아니면 0
 * valid[1][0], valid[1][1], ..., valid[1][8]: 각 열이 올바른지 1, 아니면 0
 * valid[2][0], valid[2][1], ..., valid[2][8]: 각 3x3 그리드가 올바른지 1, 아니면 0
 */
int valid[3][9];

/*
 * 스도쿠 퍼즐의 각 행이 올바른지 검사한다.
 * 행 번호는 0 부터 시작하며, i 번 행이 올바른지 valid[0][i]에 1 을 기록한다.
 */
void *check_rows(void *arg)
{
    for(int i=0;i<9;i++){
        int check_data[10] = {0,1,2,3,4,5,6,7,8,9};
        int check_now ,check_sum=0;
        for(int j=0;j<9;j++){
            check_now = sudoku[i][j];
            check_data[check_now]=0;
        }
        for(int j=1;j<=9;j++){
            check_sum += check_data[j];
        }
        if(check_sum != 0){
            valid[0][i] = 0;
        }
        else{
            valid[0][i] = 1;
        }
    }
    pthread_exit(NULL);
}
```

```

}

/*
 * 스도쿠 퍼즐의 각 열이 올바른지 검사한다.
 * 열 번호는 0 부터 시작하며, j 번 열이 올바르면 valid[1][j]에 1을 기록한다.
 */
void *check_columns(void *arg)
{
    for(int i=0;i<9;i++){
        int check_data[10]={0,1,2,3,4,5,6,7,8,9};
        int check_now ,check_sum = 0;
        for(int j=0;j<9;j++){
            check_now= sudoku[j][i];
            check_data[check_now]=0;
        }
        for(int j=1;j<=9;j++){
            check_sum += check_data[j];
        }
        if(check_sum != 0){
            valid[1][i] = 0;
        }
        else{
            valid[1][i] = 1;
        }
    }
    pthread_exit(NULL);
}

/*
 * 스도쿠 퍼즐의 각 3x3 서브그리드가 올바른지 검사한다.
 * 3x3 서브그리드 번호는 0 부터 시작하며, 왼쪽에서 오른쪽으로, 위에서 아래로 증가한다.
 * k 번 서브그리드가 올바르면 valid[2][k]에 1을 기록한다.
 */
void *check_subgrid(void *arg)
{
    int k = *(int *)arg;
    int check_data[10]={0,1,2,3,4,5,6,7,8,9};
    int check_now ,check_sum = 0;
    for(int i=(k/3)*3;i<(k/3)*3+3;i++){
        for(int j=(k%3)*3;j<(k%3)*3+3;j++){
            check_now = sudoku[i][j];
            check_data[check_now] = 0;
        }
    }
    for(int i=1;i<=9;i++){
        check_sum += check_data[i];
    }
    if(check_sum != 0){
        valid[2][k] = 0;
    }
}

```

```

    else{
        valid[2][k] = 1;
    }
    pthread_exit(NULL);
}

/*
 * 스도쿠 퍼즐이 올바르게 구성되어 있는지 11 개의 스레드를 생성하여 검증한다.
 * 한 스레드는 각 행이 올바른지 검사하고, 다른 한 스레드는 각 열이 올바른지 검사한다.
 * 9 개의 3x3 서브그리드에 대한 검증은 9 개의 스레드를 생성하여 동시에 검사한다.
 */
void check_sudoku(void)
{
    int i, j;
    /*
     * 검증하기 전에 먼저 스도쿠 퍼즐의 값을 출력한다.
     */
    for (i = 0; i < 9; ++i) {
        for (j = 0; j < 9; ++j)
            printf("%d", sudoku[i][j]);
        printf("\n");
    }
    printf("---\n");
    /*
     * 스레드를 생성하여 각 행을 검사하는 check_rows() 함수를 실행한다.
     */
    pthread_t th_id[11];
    if (pthread_create(&th_id[9], NULL, check_rows, NULL) != 0) {
        fprintf(stderr, "pthread_create error: check_rows\n");
        exit(-1);
    }
    /*
     * 스레드를 생성하여 각 열을 검사하는 check_columns() 함수를 실행한다.
     */
    if (pthread_create(&th_id[10], NULL, check_columns, NULL) != 0) {
        fprintf(stderr, "pthread_create error: check_columns\n");
        exit(-1);
    }
    /*
     * 9 개의 스레드를 생성하여 각 3x3 서브그리드를 검사하는 check_subgrid() 함수를 실행한다.
     * 3x3 서브그리드의 위치를 식별할 수 있는 값을 함수의 인자로 넘긴다.
     */
    int subgrid_num[9]={0,1,2,3,4,5,6,7,8};
    for(i=0;i<9;i++){
        int *num = &subgrid_num[i];
        if(pthread_create(&th_id[i],NULL,check_subgrid,(void *)num) != 0){
            fprintf(stderr, "pthread_create error: check_subgrid\n");
            exit(-1);
        }
    }
}

```

```

/*
 * 11 개의 스레드가 종료할 때까지 기다린다.
 */
for(i=0;i<11;i++){
    pthread_join(th_id[i],NULL);
}
/*
 * 각 행에 대한 검증 결과를 출력한다.
 */
printf("ROWS: ");
for (i = 0; i < 9; ++i)
    printf(valid[0][i] == 1 ? "(%d,YES)" : "(%d,NO)", i);
printf("\n");
/*
 * 각 열에 대한 검증 결과를 출력한다.
 */
printf("COLS: ");
for (i = 0; i < 9; ++i)
    printf(valid[1][i] == 1 ? "(%d,YES)" : "(%d,NO)", i);
printf("\n");
/*
 * 각 3x3 서브그리드에 대한 검증 결과를 출력한다.
 */
printf("GRID: ");
for (i = 0; i < 9; ++i)
    printf(valid[2][i] == 1 ? "(%d,YES)" : "(%d,NO)", i);
printf("\n---\n");
}

/*
 * 스도쿠 퍼즐의 값을 3x3 서브그리드 내에서 무작위로 섞는 함수이다.
 */
void *shuffle_sudoku(void *arg)
{
    int i, tmp;
    int grid;
    int row1, row2;
    int col1, col2;

    srand(time(NULL));
    for (i = 0; i < 100; ++i) {
        /*
         * 0 부터 8 번 사이의 서브그리드 하나를 무작위로 선택한다.
         */
        grid = rand() % 9;
        /*
         * 해당 서브그리드의 좌측 상단 행열 좌표를 계산한다.
         */
        row1 = row2 = (grid/3)*3;
        col1 = col2 = (grid%3)*3;
    }
}

```

```

/*
 * 해당 서브그리드 내에 있는 임의의 두 위치를 무작위로 선택한다.
 */
row1 += rand() % 3; col1 += rand() % 3;
row2 += rand() % 3; col2 += rand() % 3;
/*
 * 홀수 서브그리드이면 두 위치에 무작위 수로 채우고,
 */
if (grid & 1) {
    sudoku[row1][col1] = rand() % 8 + 1;
    sudoku[row2][col2] = rand() % 8 + 1;
}
/*
 * 짝수 서브그리드이면 두 위치에 있는 값을 맞바꾼다.
 */
else {
    tmp = sudoku[row1][col1];
    sudoku[row1][col1] = sudoku[row2][col2];
    sudoku[row2][col2] = tmp;
}
}
pthread_exit(NULL);
}

/*
 * 메인 함수는 위에서 작성한 함수가 올바르게 동작하는지 검사하기 위한 것으로 수정하면 안 된다.
 */
int main(void)
{
    int tmp;
    pthread_t tid;

    /*
     * 기본 스도쿠 퍼즐을 출력하고 검증한다.
     */
    check_sudoku();
    /*
     * 기본 퍼즐에서 값 두개를 맞바꾸고 검증해본다.
     */
    tmp = sudoku[5][3]; sudoku[5][3] = sudoku[6][2]; sudoku[6][2] = tmp;
    check_sudoku();
    /*
     * 기본 스도쿠 퍼즐로 다시 바꾼 다음, shuffle_sudoku 스레드를 생성하여 퍼즐을 섞는다.
     */
    tmp = sudoku[5][3]; sudoku[5][3] = sudoku[6][2]; sudoku[6][2] = tmp;
    if (pthread_create(&tid, NULL, shuffle_sudoku, NULL) != 0) {
        fprintf(stderr, "pthread_create error: shuffle_sudoku\n");
        exit(-1);
    }
    /*

```



```
    * 무작위로 섞는 중인 스도쿠 퍼즐을 검증해본다.
    */
    check_sudoku();
    /*
    * shuffle_sudoku 스레드가 종료될 때까지 기다린다.
    */
    pthread_join(tid, NULL);
    /*
    * shuffle_sudoku 스레드 종료 후 다시 한 번 스도쿠 퍼즐을 검증해본다.
    */
    check_sudoku();
    exit(0);
}
```

<Compilation Process>

```
proj2 — -zsh — 97x41
(base) namjeonghun@songhae proj2 % gcc -v os_project_2.c
Apple clang version 11.0.3 (clang-1103.0.32.62)
Target: x86_64-apple-darwin19.4.0
Thread model: posix
InstalledDir: /Library/Developer/CommandLineTools/usr/bin
"/Library/Developer/CommandLineTools/usr/bin/clang" -cc1 -triple x86_64-apple-macosx10.15.0 -Wde
precated-objc-isa-usage -Werror=deprecated-objc-isa-usage -emit-obj -mrelax-all -disable-free -di
sable-llvm-verifier -discard-value-names -main-file-name os_project_2.c -mrelocation-model pic -p
ic-level 2 -mthread-model posix -mframe-pointer=all -fno-strict-return -masm-verbose -munwind-tab
les -target-sdk-version=10.15.4 -target-cpu penryn -dwarf-column-info -debugger-tuning=lldb -targ
et-linker-version 556.6 -v -resource-dir /Library/Developer/CommandLineTools/usr/lib/clang/11.0.3
-isysroot /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk -I/usr/local/include -internal-isys
tem /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/local/include -internal-isystem /Lib
rary/Developer/CommandLineTools/usr/lib/clang/11.0.3/include -internal-externc-isystem /Library/D
eveloper/CommandLineTools/SDKs/MacOSX.sdk/usr/include -internal-externc-isystem /Library/Develo
per/CommandLineTools/usr/include -Wno-objc-signed-char-bool-implicit-int-conversion -Wno-extra-semi
-stmt -Wno-quoted-include-in-framework-header -fdebug-compilation-dir /Users/namjeonghun/Desktop/
운영체제/proj2 -ferror-limit 19 -fmessage-length 73 -stack-protector 1 -fstack-check -mdarwin-stk
chk-strong-link -fblocks -fencode-extended-block-signature -fregister-global-dtors-with-atexit -f
objc-runtime=macosx-10.15.0 -fmax-type-align=16 -fdiagnostics-show-option -fcolor-diagnostics -o
/var/folders/53/gwrl58gs1gz6yf235vtypd5r0000gn/T/os_project_2-173cea.o -x c os_project_2.c
clang -cc1 version 11.0.3 (clang-1103.0.32.62) default target x86_64-apple-darwin19.4.0
ignoring nonexistent directory "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/local/inc
lude"
ignoring nonexistent directory "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/Library/Frame
works"
#include "... search starts here:
#include <...> search starts here:
/usr/local/include
/Library/Developer/CommandLineTools/usr/lib/clang/11.0.3/include
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include
/Library/Developer/CommandLineTools/usr/include
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/System/Library/Frameworks (framework directo
ry)
End of search list.
"/Library/Developer/CommandLineTools/usr/bin/ld" -demangle -lto_library /Library/Developer/Comma
ndLineTools/usr/lib/libLT0.dylib -no_deduplicate -dynamic -arch x86_64 -platform_version macos 10
.15.0 10.15.4 -syslibroot /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk -o a.out /var/folde
rs/53/gwrl58gs1gz6yf235vtypd5r0000gn/T/os_project_2-173cea.o -L/usr/local/lib -lSystem /Library/D
eveloper/CommandLineTools/usr/lib/clang/11.0.3/lib/darwin/libclang_rt.osx.a
(base) namjeonghun@songhae proj2 %
```

```

(base) namjeonghun@songhae proj2 % gcc -v os_project_2.c
Apple clang version 11.0.3 (clang-1103.0.32.62)
Target: x86_64-apple-darwin19.4.0
Thread model: posix
InstalledDir: /Library/Developer/CommandLineTools/usr/bin
"/Library/Developer/CommandLineTools/usr/bin/clang" -cc1 -triple x86_64-apple-macosx10.15.0 -
Wdeprecated-objc-isa-usage -Werror=deprecated-objc-isa-usage -emit-obj -mrelax-all -disable-free -
disable-llvm-verifier -discard-value-names -main-file-name os_project_2.c -mrelocation-model pic -
pic-level 2 -mthread-model posix -mframe-pointer=all -fno-strict-return -masm-verbose -munwind-
tables -target-sdk-version=10.15.4 -target-cpu penryn -dwarf-column-info -debugger-tuning=lldb -
target-linker-version 556.6 -v -resource-dir
/Library/Developer/CommandLineTools/usr/lib/clang/11.0.3 -isysroot
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk -I/usr/local/include -internal-isystem
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/local/include -internal-isystem
/Library/Developer/CommandLineTools/usr/lib/clang/11.0.3/include -internal-externc-isystem
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include -internal-externc-isystem
/Library/Developer/CommandLineTools/usr/include -Wno-objc-signed-char-bool-implicit-int-conversion -
Wno-extra-semi-stmt -Wno-quoted-include-in-framework-header -fdebug-compilation-dir
/Users/namjeonghun/Desktop/운영체제/proj2 -ferror-limit 19 -fmessage-length 73 -stack-protector 1 -
fstack-check -mdarwin-stkchk-strong-link -fblocks -fencode-extended-block-signature -fregister-
global-dtors-with-atexit -fobjc-runtime=macosx-10.15.0 -fmax-type-align=16 -fdiagnostics-show-option
-fcolor-diagnostics -o /var/folders/53/gwr158gs1gz6ydf235vtypd5r0000gn/T/os_project_2-173cea.o -x c
os_project_2.c
clang -cc1 version 11.0.3 (clang-1103.0.32.62) default target x86_64-apple-darwin19.4.0
ignoring nonexistent directory
"/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/local/include"
ignoring nonexistent directory
"/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/Library/Frameworks"
#include "...": search starts here:
#include <...>: search starts here:
  /usr/local/include
  /Library/Developer/CommandLineTools/usr/lib/clang/11.0.3/include
  /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include
  /Library/Developer/CommandLineTools/usr/include
  /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/System/Library/Frameworks (framework directory)
End of search list.
"/Library/Developer/CommandLineTools/usr/bin/ld" -demangle -lto_library
/Library/Developer/CommandLineTools/usr/lib/libLTO.dylib -no_deduplicate -dynamic -arch x86_64 -
platform_version macos 10.15.0 10.15.4 -syslibroot
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk -o a.out
/var/folders/53/gwr158gs1gz6ydf235vtypd5r0000gn/T/os_project_2-173cea.o -L/usr/local/lib -lSystem
/Library/Developer/CommandLineTools/usr/lib/clang/11.0.3/lib/darwin/libclang_rt.osx.a

```

<Outputs and Descriptions>

```
proj2 — -zsh — 73x59
[(base) namjeonghun@songhae proj2 % gcc os_project_2.c -lpthread
[(base) namjeonghun@songhae proj2 % ./a.out
639841275
724953168
185726394
256137489
491582637
873469521
542398716
318675942
967214853
---
ROWS: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
COLS: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
---
639841275
724953168
185726394
256137489
491582637
873269521
544398716
318675942
967214853
---
ROWS: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,NO)(6,NO)(7,YES)(8,YES)
COLS: (0,YES)(1,YES)(2,NO)(3,NO)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,NO)(5,YES)(6,NO)(7,YES)(8,YES)
---
639841275
724953168
185726394
256137489
491582637
873469521
542398716
318675942
967214853
---
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
GRID: (0,YES)(1,NO)(2,YES)(3,NO)(4,YES)(5,NO)(6,YES)(7,NO)(8,YES)
---
582844279
134684468
769216351
862832486
441579474
828461722
294317194
138635523
675387786
---
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
GRID: (0,YES)(1,NO)(2,YES)(3,NO)(4,YES)(5,NO)(6,YES)(7,NO)(8,YES)
---
(base) namjeonghun@songhae proj2 %
```

Output

```
(base) namjeonghun@songhae proj2 % gcc os_project_2.c -lpthread
```

```
(base) namjeonghun@songhae proj2 % ./a.out
```

```
639841275
```

```
724953168
```

```
185726394
```

```
256137489
```

```
491582637
```

```
873469521
```

```
542398716
```

```
318675942
```

```
967214853
```

```
---
```

```
ROWS: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
```

```
COLS: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
```

```
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
```

```
---
```

```
639841275
```

```
724953168
```

```
185726394
```

```
256137489
```

```
491582637
```

```
873269521
```

```
544398716
```

```
318675942
```

```
967214853
```

```
---
```

```
ROWS: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,NO)(6,NO)(7,YES)(8,YES)
```

```
COLS: (0,YES)(1,YES)(2,NO)(3,NO)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
```

```
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,NO)(5,YES)(6,NO)(7,YES)(8,YES)
```

```
---
```

```
639841275
```

```
724953168
```

```
185726394
```

```
256137489
```

```
491582637
```

```
873469521
```

```
542398716
```

```
318675942
```

```
967214853
```

```
---
```

```
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
```

```
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
```

```
GRID: (0,YES)(1,NO)(2,YES)(3,NO)(4,YES)(5,NO)(6,YES)(7,NO)(8,YES)
```

```
---
```

```
582844279
```

```
134684468
```

```
769216351
```

```
862832486
```

```
441579474
```

```
828461722
```

```
294317194
```

```
138635523
```

```
675387786
```

```
---
```

```
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
```

```
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
```

```
GRID: (0,YES)(1,NO)(2,YES)(3,NO)(4,YES)(5,NO)(6,YES)(7,NO)(8,YES)
```

```
---
```

Descriptions

첫번째 `check_sudoku()`는 기본 스도쿠를 검증하므로 모두 YES 출력.

두번째 `check_sudoku()`는 값 두개를 맞바꾸었으므로 바뀐 열과 행만 NO 출력.

세번째 `check_sudoku()`는 무작위로 섞는 중인 스도쿠 퍼즐을 검증 후 출력.

네번째 `check_sudoku()`는 셔플이 완료된 스도쿠를 검증 후 출력.