

int main (argc, argv)

- 예외처리
- 동작할당
- 정수변환
- 계산
- 배열 출력
- 할당해제

main.c
모든 함수 호출
malloc으로
동작할당

int ft_handle_exception (char * str, int argc)

예외처리 하는 함수

입력 : 인자로 들어오는 문자열, 인자 개수

return : 예외처리 결과

0 : Error

1 : Success

- (예외)
- 배열길이가 != 16일 때
 - 정수가 1 ≤ ≤ 4 범위인지
 - argc != 2
 - " " 공백이 아닌 경우
- } 출력

void ft_free (udlr, ans)

메모리 할당 해제하는 함수

Void ft_atoi (char * str)

문자열 자르고 정수 배열에 저장하는 함수

입력 : 인자로 들어오는 문자열
"col1up col2up ..."

return : 배열의 주소값

Void ft_print_box (int * ans)

답을 출력하는 함수

```
| v | v | v | /n
| v | v | v | /n
| v | v | v | /n
| v | v | v | /n
```

int ft_stack_box (int index, int ** udlr, int * ans)

상자 쌓는 함수 (재귀로 구현)

입력 : 답배열의 인덱스, 입력으로 들어온 행·열 저장 배열, 답배열 주소값

return : 1 상자 개수 가능
0 상자 개수 불가능

→ 옳은 자리인지 확인 : ft_check (index, udlr, ans)

int ft_check (int index, int ** udlr, int * ans)

상자 개수가 맞는지 확인하는 함수

- 상자 개수를 udlr과 비교
- 중복이 있는지 확인



1: True 0: False

int ft_1234check (int index, int * ans)

중복 없는지 체크하는 함수
(1, 2, 3, 4)가 되어야함

1: True
0: False

int ft_udlr_check (int index, int *udlr[], int *ans)

상자 개수를 확인하는 함수

int r-check (index, udlr, ans)

오른쪽 상자 개수를 확인하는 함수

index가 3, 7, 11, 15일 때 확인한다

int l-check (index, udlr, ans)

왼쪽 상자 개수를 확인하는 함수

index가 3, 7, 11, 15 일 때 확인한다

int u-check (index, udlr, ans)

위쪽 상자 개수를 확인하는 함수

index가 12, 13, 14, 15 일 때 확인한다

int d-check (index, udlr, ans)

아래쪽 상자 개수를 확인하는 함수

index가 12, 13, 14, 15 일 때 확인한다

ft_udlr_check.c
상자 개수 확인하는
함수들이 들어있다

```
int main ( int argc, char* argv[] )
```

< main 흐름도 >

메인함수

```
int result  
int ** udlr  
int * ans
```

입력 예외처리

```
( if ( ft_handle_exception (argc, argv[1]) == 0 )  
  return 0 )
```

입력 정수로 변경해서 저장

```
ans = (int *) malloc ( sizeof(int) * 16 )
```

```
udlr = (int **) malloc ( sizeof(int *) * 4 )
```

```
while ( i < 4 )
```

```
  udlr[i] = (int *) malloc ( sizeof ... )
```

계산

```
ft_atoi( argv[i], udlr )
```

배열 출력

```
( result = ft_stack_box ( 0, udlr, ans ) )
```

할당 해제

```
( if ( result )  
  ft_print_box ( ans )  
  else  
    write ( "Error" ) )
```

```
( ft_free ( ans, udlr ) )
```

```
return 0
```

성민

상민

재현



