인터페이스 정의서

목차

1. 함수 정의
2. 구조체 정의
3. 매크로 상수 정의

각각의 섹션에서, 파일 단위로 정리되어 있다. 예를 들어 함수 정의 > sign.h의 섹션에 sign.h에 있는 함수들이 정리되어 있다.

함수 정의

<SIGN IN/OUT>

## sign.h ##

|  |  |  |
| --- | --- | --- |
| 함수 | char \*signinup(); | |
| 설명 | main.c 에서 성공적으로 로그인 할 때까지 계속 호출됨. sign in 성공 시 id를 리턴하고, 실패(예외 상황 발생) or sign up 시 NULL을 반환함. | |
| 리턴 | id 문자열  (char arr 첫 주소) | sign in 성공 시 |
| NULL | sign up, 예외 상황 발생 시 |

|  |  |
| --- | --- |
| 함수 | char \*get\_str(); |
| 설명 | 최대 길이 30(함수 안에서 수정 가능함, 바꿔 쓸거면 바꿔 쓰기)의 문자열을 받는다. 동적으로 받아 메모리의 낭비를 줄인다. |
| 리턴 | 동적할당한 문자열 |

|  |  |  |
| --- | --- | --- |
| 함수 | char \*signin(); | |
| 설명 | id, password를 입력받아 저장된 id파일이 있다면 저장된 정보와 비교해 일치 시 id를 리턴한다. 입력 실패, id 존재X, password 불일치 등의 나머지 상황에 NULL을 리턴한다. | |
| 리턴 | id 문자열 | sign in 성공(id, password 일치) 시 |
| NULL | sign in 실패 시 |

|  |  |
| --- | --- |
| 함수 | void signup(); |
| 설명 | id, password를 입력받아 이미 존재하는 id인지 중복 체크를 하고, 존재하지 않는다면 data/personal\_info/{id}.txt를 만들어 파일에 한 줄씩 id, password를 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | char \*\*get\_filenames(const char \*directory, int \*count); |
| 설명 | 디렉토리 내 파일 이름들 배열과 배열의 길이를 리턴한다. (배열은 명시적 리턴, 배열 길이는 주소를 넣어서 암시적으로 리턴) 디렉토리 path와 배열 길이를 저장할 변수를 넣으면 그 디렉토리의 파일 이름들의 배열을 리턴한다. |
| 리턴 | 디렉토리의 파일 이름들 배열 |

|  |  |  |
| --- | --- | --- |
| 함수 | int check\_id(char \*id); | |
| 설명 | personal\_info 에 특정 id에 대해 id.txt가 있는지 확인하고 있다면 1, 없다면 0을 리턴한다. | |
| 리턴 | 0 | id.txt 가 personal\_info에 존재하지 X |
| 1 | id.txt 가 personal\_info에 존재 |

|  |  |  |
| --- | --- | --- |
| 함수 | int check\_password(char \*id, char \*password); | |
| 설명 | id.txt 파일에서 password 정보가 파라미터로 들어온 password와 일치하면 1 리턴, 일치하지 않으면 0 리턴 | |
| 리턴 | 0 | id.txt에서 password 불일치 |
| 1 | id.txt에서 password 일치 |

<HOME>

## home.h

|  |  |
| --- | --- |
| 함수 | int home() |
| 설명 | home 창에서 1,2,3,4를 입력하여 FEED, SEARCH, MY PAGE, RECOMMEND 함수를 실행하여 그 메뉴로 이동하게 하는 함수이다. |
| 리턴 | 입력받은 값을 그대로 리턴한다. |

<FEED>

## basic\_function.h ##

|  |  |
| --- | --- |
| 함수 | void to\_lower\_case(char \*str); |
| 설명 | 소문자로 변환하는 함수 |
| 리턴 | 주어진 코드가 public.txt에 있으면 1, 없으면 0 |

|  |  |
| --- | --- |
| 함수 | char \*trim\_whitespace(char \*str); |
| 설명 | 공백 문자 제거 함수 |
| 리턴 | 공백 문자가 제거된 str을 반환 |

|  |  |  |
| --- | --- | --- |
| 함수 | void refresh\_print(const char \*format, ...); | |
| 설명 | printf와 비슷, shell에서 이전까지 입력/출력된 것을 모두 지우고 printf한다. 매개변수는 printf와 동일하게 넣으면 된다. | |
| 리턴 | 0 | id.txt에서 password 불일치 |
| 1 | id.txt에서 password 일치 |

## structs.h ##

|  |  |
| --- | --- |
| 함수 | void init\_darray(darray \*array); |
| 설명 | 동적 배열 초기화 함수 (다른 종류의 동적 배열에도 존재) |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | void addto\_darray(darray \*array, char \*item); |
| 설명 | 동적 배열에 항목 추가 함수 (다른 종류의 동적 배열에도 존재) |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | void free\_darray(darray \*array); |
| 설명 | 동적 배열 해제 함수 (다른 종류의 동적 배열에도 존재) |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | f\_darray return\_nutrient(char\* name); |
| 설명 | 재료 이름으로부터 영양소 f\_darray로 반환하는 함수 |
| 리턴 | 영양소 양이 담긴 f\_darray |

## recipe\_utils.h ##

|  |  |
| --- | --- |
| 함수 | void save\_memo(const char \*code, const char \*memo); |
| 설명 | 고유코드 및 메모와 id를 받아서 그 id를 가지는 파일의 마지막에 레시피의 고유코드와 메모를 저장하는 코드 |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | void read\_recipe\_from\_file(const char \*code, RECIPE \*recipe); |
| 설명 | 텍스트 파일에서 Recipe 구조체로 데이터를 읽어오는 함수 |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | void store\_lines\_public(char \*\*lines, int \*line\_count, RECIPE \*recipe); |
| 설명 | recipe에서 출력할 줄들을 저장하는 배열 만드는 함수, public에서 사용 |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | void store\_lines\_private(char \*\*lines, int \*line\_count, RECIPE \*recipe); |
| 설명 | recipe에서 출력할 줄들을 저장하는 배열 만드는 함수, private에서 사용 |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | void display\_recipe\_public(RECIPE \*recipe); |
| 설명 | store\_lines에서 만든 배열로부터 각 줄을 꺼내서 30줄씩 출력하는 함수,  g를 누르면 좋아요 추가, c를 누르면 댓글 추가, u & d를 누르면 페이지 전환 , q를 누르면 셀이 모두 지워지면서 함수가 종료됨 |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | void display\_recipe\_private(RECIPE \*recipe); |
| 설명 | 재료 이름으로부터 영양소 f\_darray로 반환하는 함수, u & d를 누르면 페이지 전환 , q를 누르면 셀이 모두 지워지면서 함수가 종료됨 (댓글, 좋아요 기능 없음) |
| 리턴 | 영양소 양이 담긴 f\_darray |

|  |  |
| --- | --- |
| 함수 | f\_darray return\_nutrient(char\* name); |
| 설명 | 재료 이름으로부터 영양소 f\_darray로 반환하는 함수 |
| 리턴 | 영양소 양이 담긴 f\_darray |

|  |  |
| --- | --- |
| 함수 | void create\_recipe\_file(); |
| 설명 | 레시피를 입력받아 파일에 저장하는 함수 |
| 리턴 | 없음 |

## feed.h ##

|  |  |
| --- | --- |
| 함수 | int is\_code\_in\_public\_txt(const char \*code); |
| 설명 | 주어진 코드가 public.txt에 있는지 확인 |
| 리턴 | 주어진 코드가 public.txt에 있으면 1, 없으면 0 |

|  |  |
| --- | --- |
| 함수 | char \*get\_code(const char \*input); |
| 설명 | 코드 처리 함수 |
| 리턴 | Input에 따라 random 혹은 입력값으로 코드를 반 |

|  |  |
| --- | --- |
| 함수 | void store\_lines\_feed(char \*\*lines, int \*lineCount, RECIPE \*recipe); |
| 설명 | recipe에서 출력할 줄들을 저장하는 배열 만드는 함수, feed에서 사용 |
| 리턴 | 없음 |

|  |  |
| --- | --- |
| 함수 | void display\_recipe\_feed(); |
| 설명 | store\_lines에서 만든 배열로부터 각 줄을 꺼내서 30줄씩 출력하는 함수  g를 누르면 좋아요 추가, c를 누르면 댓글 추가, u & d를 누르면 페이지 전환 , q를 누르면 셀이 모두 지워지면서 함수가 종료됨, A를 누르면 다음 피드로 |
| 리턴 | 없음 |

<SEARCH>

## search.h ##

|  |  |
| --- | --- |
| 함수 | void get\_ingre\_name(ingre\_darray ingredients, darray \*ingre\_names) |
| 설명 | ingre\_darray ingredients의 ingre들의 이름을 darray \*ingre\_names에 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | int count\_matches(darray \*arr, darray \*subarr) |
| 설명 | 두 darray의 공통 원소의 개수를 반환한다. |
| 리턴 | arr의 원소와 subarr의 원소 중 일치하는 쌍의 개수 |

|  |  |
| --- | --- |
| 함수 | void sort\_codes(darray \*codes, int \*keys) |
| 설명 | 고유코드를 keys의 원소에 따라 내림차순으로 정렬한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void split\_string(darray \*array, const char \*string, char key) |
| 설명 | string을 입력받아 key를 기준으로 분할하여 darray \*array에 저장한다. 파이썬의 .split(key)와 유사하다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void get\_full\_code(darray \*codes) |
| 설명 | public의 모든 레시피의 고유코드를 darray codes에 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | int \*get\_likes(darray \*codes) |
| 설명 | codes에 저장된 레시피들의 좋아요 수를 int형 배열로 반환한다. |
| 리턴 | 레시피들의 좋아요 수를 담은 int형 배열의 주소 |

|  |  |
| --- | --- |
| 함수 | int \*get\_likes(darray \*codes) |
| 설명 | codes에 저장된 레시피들의 좋아요 수를 int형 배열로 반환한다. |
| 리턴 | 레시피들의 좋아요 수를 담은 int형 배열의 주소 |

|  |  |
| --- | --- |
| 함수 | void split\_args(QUERY \*query, const char \*args) |
| 설명 | 입력된 텍스트 args를 받아 변환한 쿼리를 QUERY query에 저장한다.  args는 다음의 형식으로 입력된다.  text=[]/tag=[]&[]/code=[]/ingredient=[]&[]/option=[like/tag/ingredient] |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void search\_with\_text(darray \*codes, darray \*full\_codes, const char \*text) |
| 설명 | full\_codes에 저장된 레시피 중 제목에 text를 포함하는 레시피의 고유코드를 darray codes에 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | int \*search\_with\_tag(darray \*codes, darray \*full\_codes, darray \*tags) |
| 설명 | full\_codes에 저장된 레시피 중 tags의 원소 중 하나 이상을 포함하는 레시피의 고유코드를 darray codes에 저장한다. |
| 리턴 | 각 레시피에 포함된 tags의 원소의 개수를 담은 int형 배열의 주소 |

|  |  |
| --- | --- |
| 함수 | int \*search\_with\_ingredient(darray \*codes, darray \*full\_codes, darray \*ingres) |
| 설명 | full\_codes에 저장된 레시피 중 ingres의 원소 중 하나 이상을 포함하는 레시피의 고유코드를 darray codes에 저장한다. |
| 리턴 | 각 레시피에 포함된 ingres의 원소의 개수를 담은 int형 배열의 주소 |

|  |  |
| --- | --- |
| 함수 | void select\_recipe(darray \*codes) |
| 설명 | codes에 저장된 레시피를의 고유코드와 제목을 30개씩 출력한다. u, d키를 이용해 커서를 이동하고 s키를 이용해 레시피를 선택할 수 있다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void select\_recipe(darray \*codes) |
| 설명 | codes에 저장된 레시피를의 고유코드와 제목을 30개씩 출력한다. u, d키를 이용해 커서를 이동하고 s키를 이용해 레시피를 선택할 수 있다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void copy\_darray(darray \*target, darray \*original) |
| 설명 | darray target으로 darray original의 값을 복사한다. (얕은 복사) |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void search() |
| 설명 | 텍스트 입력을 받아 검색 결과를 출력한다. q를 입력하면 홈 화면으로 복귀한다. |
| 리턴 | 리턴값 없음 |

<MY PAGE>

## my\_page.h ##

|  |  |
| --- | --- |
| 함수 | void memo() |
| 설명 | 레시피 고유 코드를 받아 대응되는 레시피에 메모를 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | int varify\_id() |
| 설명 | Id와 password를 입력받아 일치하는지 확인한다. |
| 리턴 | 중간에 문제가 있었을 경우 0, 아닐 경우 1을 리턴한다 |

|  |  |
| --- | --- |
| 함수 | void replace\_second\_line(const cahr \*filename, const char \*new\_line) |
| 설명 | 파일의 두번째 줄을 찾아 바꾼다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | int modify(int re) |
| 설명 | 새로운 비밀번호를 받아 replace\_second\_line함수를 이용하여 id.txt파일의 비밀번호를 수정한다. |
| 리턴 | 제대로 수행되었으면 0, 아니라면 1 |

<RECOMMEND>

## recommend.h ##

|  |  |
| --- | --- |
| 함수 | void recommend() |
| 설명 | 명령어를 입력받는 recommend 함수로, 함수 내에서 명령어를 입력하면 그에 맞는 옵션을 추천해준다. 가능한 옵션으로는 :q, help, cal, protein, fat, lcar\_hfat, today\_menu가 있으며, 각 옵션의 기능은 후술한다. |
| 리턴 | 리턴값 없음 |
| 함수 | void help() |
| 설명 | help 함수를 실행하면 사용 가능한 모든 명령어의 종류를 설명하고, 각 명령어에 대한 키워드를 입력했을 때 그 명령어의 자세한 설명을 출력한다. 가령 :q와 cal의 키워드는 각각 q와 c이다. 잘못된 키워드를 입력했을 경우 함수가 처음부터 다시 실행된다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void sort\_calories() |
| 설명 | 칼로리 순으로 공용 아카이브에 있는 레시피들을 정렬한다. 사용자는 칼로리 순을 오름차순으로 할 지 내림차순으로 할 지 설정 가능하며, 이 중 하나를 선택하여 정렬하고 정렬된 배열의 몇 번째 인덱스의 레시피를 볼지 설정 가능하다. 선택되면 칼로리 값을 출력하며, 레시피를 확인할 수 있다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void sort\_protein() |
| 설명 | 단백질 순으로 공용 아카이브에 있는 레시피들을 정렬한다. 사용자는 단백질 순을 오름차순으로 할 지 내림차순으로 할 지 설정 가능하며, 이 중 하나를 선택하여 정렬하고 정렬된 배열의 몇 번째 인덱스의 레시피를 볼지 설정 가능하다. 선택되면 단백질 값을 출력하며, 레시피를 확인할 수 있다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void sort\_fat() |
| 설명 | 지방 순으로 공용 아카이브에 있는 레시피들을 정렬한다. 사용자는 지방 순을 오름차순으로 할 지 내림차순으로 할 지 설정 가능하며, 이 중 하나를 선택하여 정렬하고 정렬된 배열의 몇 번째 인덱스의 레시피를 볼지 설정 가능하다. 선택되면 지방 값을 출력하며, 레시피를 확인할 수 있다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void calc\_calories(RECIPE\* recipe, float\* calorie) |
| 설명 | 특정한 레시피 포인터가 입력되었을 때. 그 레시피의 ingre\_darray에 접근하여 각 재료의 영양 값들을 이용해서 그 레시피의 칼로리를 계산하여 float 포인터 변수 값에 저장한다. |
| 리턴 | 리턴값 없음 |
| 함수 | void calc\_protein(RECIPE\* recipe, float\* protein) |
| 설명 | 특정한 레시피 포인터가 입력되었을 때. 그 레시피의 ingre\_darray에 접근하여 각 재료의 영양 값들을 이용해서 그 레시피의 단백질을 계산하여 float 포인터 변수 값에 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void calc\_fat(RECIPE\* recipe, float\* fat) |
| 설명 | 특정한 레시피 포인터가 입력되었을 때. 그 레시피의 ingre\_darray에 접근하여 각 재료의 영양 값들을 이용해서 그 레시피의 지방을 계산하여 float 포인터 변수 값에 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void calc\_car(RECIPE\* recipe, float\* car) |
| 설명 | 특정한 레시피 포인터가 입력되었을 때. 그 레시피의 ingre\_darray에 접근하여 각 재료의 영양 값들을 이용해서 그 레시피의 탄수화물 양을 계산하여 float 포인터 변수 값에 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void lcar\_hfat() |
| 설명 | low carbohydrate, high fat의 약자로, 저탄고지 옵션을 의미한다. 저탄고지를 기준으로 공용 아카이브에 있는 레시피들을 오름차순과 내림차순으로 정렬하며, 이 중 인덱스를 선택하여 저탄고지 지수와 레시피를 확인할 수 있다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void today\_menu() |
| 설명 | 오늘의 메뉴를 추천해주는 함수로, 유저가 원하는 만큼 태그와 재료를 입력 가능하다. 그런 태그와 재료들이 모두 포함되어 있는 레시피를 공용 아카이브로 찾아, 좋아요와 댓글 수를 기준으로 레시피를 추천해준다. 레시피 선정 기준은 exp(2\*likes + 1\*comments)에 비례하는 확률 분포로 결정된다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | int weighted\_random\_selection(int\* values,float\* probabilities, int size) |
| 설명 | 함수 today\_menu에서 확률 분포를 입력했을 경우에 확률 분포에 따라 값을 뽑는 함수이다. values에 값들이 저장되어 있고, probabilities는 합이 1인 확률 분포이다. size는 values 배열의 값 개수이며, values 안의 값을 하나 선택하여 리턴한다. |
| 리턴 | 뽑힌 int 변수 하나 |

|  |  |
| --- | --- |
| 함수 | void return\_all\_code(darray\* filenames) |
| 설명 | data/archive에 있는 모든 레시피 파일들의 고유 코드를 filenames에 저장한다. 구조체 darray는 char\*\*arr 과 원소 개수 count로 이루어져 있으며, arr의 각 위치 원소에 고유 코드를 저장하고 그 개수를 count에 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void return\_public\_code(darray\* filenames) |
| 설명 | data/archive에 있는 모든 레시피 파일들 중 public.txt에 있는 공용 아카이브의 고유 코드만을 filenames에 저장한다. 구조체 darray는 char\*\*arr 과 원소 개수 count로 이루어져 있으며, arr의 각 위치 원소에 고유 코드를 저장하고 그 개수를 count에 저장한다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void get\_file\_names(const char\* directoryPath, darray\* filenames) |
| 설명 | return\_all\_code와 return\_public\_code 함수에 사용되는 함수로, 디렉토리 경로를 입력하면 그 경로에 있는 모든 파일의 이름을 drray\* filenames의 성분에 저장한다. 앞 두 함수와 차이는 .txt 확장자가 있는 상태로 저장한다는 것이다. |
| 리턴 | 리턴값 없음 |

|  |  |
| --- | --- |
| 함수 | void remove\_txt\_extension(char\* filename) |
| 설명 | get\_file\_names 함수에서 구해진 확장자가 있는 파일 명에서 .txt를 제거하는 역할을 하는 함수이다. char 포인터를 사용하며, 예를 들어 111111.txt를 111111로 바꾼다. |
| 리턴 | 리턴값 없음 |

구조체 정의

## structs.h ##

|  |  |
| --- | --- |
| 구조체 | darray  typedef struct {      char \*\*arr;      int count;  } darray; |
| 사용 | 길이를 모르는 문자열의 배열에 사용(레시피의 본문, 태그 등) |
| 같이 쓰는 함수 | init\_darray, addto\_darray, free\_darray |

|  |  |
| --- | --- |
| 구조체 | f\_darray  typedef struct {      char \*\*arr;      int count;  } f\_darray; |
| 사용 | 길이를 모르는 실수의 배열에 사용(영양소 정보 배열 등) |
| 같이 쓰는 함수 | init\_f\_darray, addto\_f\_darray, free\_f\_darray, return\_nutrient |

|  |  |
| --- | --- |
| 구조체 | ingre  struct ingre { // 재료 구조체의 경우 모든 속성이 동적할당      char \*name;      f\_darray nutrient;      float \*quantity;      char \*unit;  }; |
| 사용 | 레시피의 재료 객체 다룰 때 사용 |
| 같이 쓰는 함수 | free\_ingre |

|  |  |
| --- | --- |
| 구조체 | ingre\_darray  typedef struct {      struct ingre \*\*arr;      int count;  } ingre\_darray; |
| 사용 | 레시피에서 재료들의 배열에 사용 |
| 같이 쓰는 함수 | Init\_ingre\_darray, addto\_ingre\_darray, free\_ingre\_darray |

|  |  |
| --- | --- |
| 구조체 | RECIPE  typedef struct { // 레시피 구조체의 경우 darray 꼴 속성만 동적할당      char \*title;      char \*username;      darray tag;      ingre\_darray ingre\_arr;      darray text;      int likes;      darray comments;      darray refer;      darray refered;      char \*code;  } RECIPE; |
| 사용 | 레시피의 구조체 |
| 같이 쓰는 함수 | init\_recipe, free\_recipe |

|  |  |
| --- | --- |
| 구조체 | QUERY  typedef struct {  char \*text;  darray \*tags;  darray \*ingres;  char \*code;  int orderby;  } QUERY; |
| 사용 | 검색 쿼리의 구조체 |
| 같이 쓰는 함수 | split\_args, get\_codes |

매크로 상수 정의

## recipe\_utils.h ##

|  |  |
| --- | --- |
| 매크로 상수 | MAX\_LENGTH(=2048) |
| 사용 | 레시피 객체 등 텍스트 화면에 출력 시 한 줄 최대 길이 |

|  |  |
| --- | --- |
| 매크로 상수 | MAX\_LINES(=30) |
| 사용 | 레시피 객체 등 텍스트 화면에 출력 시 열 개수 최대 길이 |