전화번호부

◆ 소스 코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <ctype.h>  #include <string.h>  #define USER\_COUNT 255  //기본 필수 자료형[available: 사용 가능한 자료인지 표기, name: 이름, num: 전화번호]  typedef struct \_user { int available = 0; char name[20]; char num[20]; } user;  void user\_add(user \*array, const int index);  void user\_del(user \*array, const int index);  void print\_all(user \*array, const int index);  void print\_search(user \*array, const int index);  int user\_struct\_compare\_name(const void \*a, const void \*b);  int user\_struct\_compare\_num(const void \*a, const void \*b);  int user\_search\_num(user \*array, const int index, const char \*input\_num);  int user\_search\_name(user \*array, const int index, const char \*input\_name);  void sort\_array(user \*array, const int index);  void user\_add(user \*array, const int index) {  rewind(stdin);  char input\_name[20] = { 0 }, input\_num[20] = { 0 };  printf("저장할 이름을 입력해 주세요: "); gets\_s(input\_name);  printf("전화번호를 입력해 주세요 : "); gets\_s(input\_num);  if (user\_search\_name(array, index, input\_name) > -1) {  printf("이미 겹치는 이름이 있습니다.\n저장을 취소합니다.\n");  return;  }  if (user\_search\_num(array, index, input\_num) > -1) {  printf("이미 겹치는 전화번호가 있습니다.\n저장을 취소합니다.\n");  return;  }  for (int i = 0; i < index; i++) if (!((array + i)->available)) {  strcpy((array + i)->name, input\_name);  strcpy((array + i)->num, input\_num);  (array + i)->available = 1;  printf("전화번호부 %d번에 %s 님의 전화번호(%s)를 저장했습니다.", i, input\_name, input\_num);  return;  }  printf("전화번호부에 저장할 공간이 없습니다.\n저장하지 못했습니다.\n");  return;  }  void user\_del(user \*array, const int index) {  int mod = 0, result = -1;  char input\_str[20] = { 0 };  while (1) {  printf("1) 이름으로 찾기 2) 전화번호로 찾기\n"); scanf(" %d", &mod);  if (0 < mod && mod < 3) break;  else printf("잘못 선택하셨습니다.\n");  mod = 0;  }  rewind(stdin);  if (mod == 1) {  printf("삭제할 이름을 입력해 주세요: "); gets\_s(input\_str);  result = user\_search\_name(array, index, input\_str);  }  else if (mod == 2) {  printf("삭제할 전화번호를 입력해 주세요 : "); gets\_s(input\_str);  result = user\_search\_num(array, index, input\_str);  }  else return;  rewind(stdin);  if (result == -1) { printf("존재하지 않는 사용자입니다.\n"); return; }  else {  printf("%s 님의 정보를 삭제합니다. 계속하시겠습니까?(Y/N)\n", (array + result)->name);  int check\_del = getchar();  if (check\_del == 'Y' || check\_del == 'y') {  (array + result)->available = 0;  printf("삭제되었습니다.\n");  }  else printf("삭제 작업이 취소되었습니다.\n");  return;  }  }  int user\_search\_name(user \*array, const int index, const char \*input\_name) {  for (int i = 0; i < index; i++)  if ((array + i)->available)  if (!(strcmp((array + i)->name, input\_name)))  return i;  return -1;//못찾으면 -1 반환  }  int user\_search\_num(user \*array, const int index, const char \*input\_num) {  for (int i = 0; i < index; i++)  if ((array + i)->available)  if (!(strcmp((array + i)->num, input\_num)))  return i;  return -1;//못찾으면 -1 반환  }  void print\_search(user \*array, const int index) {  int mod = 0, result = -1;  char input\_str[20] = { 0 };  while (1) {  printf("1) 이름으로 찾기 2) 전화번호로 찾기\n"); scanf(" %d", &mod);  if (0 < mod && mod < 3) break;  else printf("잘못 선택하셨습니다.\n");  mod = 0;  }  rewind(stdin);  if (mod == 1) {  printf("검색할 이름을 입력해 주세요: "); gets\_s(input\_str);  result = user\_search\_name(array, index, input\_str);  }  else if (mod == 2) {  printf("검색할 전화번호를 입력해 주세요 : "); gets\_s(input\_str);  result = user\_search\_num(array, index, input\_str);  }  else return;  rewind(stdin);  if (result == -1) { printf("존재하지 않는 사용자입니다.\n"); return; }  else {  printf(" 이름 | 전화번호 \n");  printf(" %10s | %14s \n", (array + result)->name, (array + result)->num);  }  }  void print\_all(user \*array, const int index) {  printf(" 이름 | 전화번호 \n");  for (int i = 0; i < index; i++)  if ((array + i)->available)  printf(" %10s | %14s \n", (array + i)->name, (array + i)->num);  }  int user\_struct\_compare\_name(const void \*a, const void \*b) {  user \*ptrA = (user\*)a;  user \*ptrB = (user\*)b;  return strcmp((char \*)ptrA->name, (char \*)ptrB->name);  }  int user\_struct\_compare\_num(const void \*a, const void \*b) {  user \*ptrA = (user\*)a;  user \*ptrB = (user\*)b;  return strcmp((char \*)ptrA->num, (char \*)ptrB->num);  }  void sort\_array(user \*array, const int index) {  int mod = 0, result = -1;  char input\_str[20] = { 0 };  while (1) {  printf("1) 이름으로 정렬 2) 전화번호로 정렬\n"); scanf(" %d", &mod);  if (0 < mod && mod < 3) break;  else printf("잘못 선택하셨습니다.\n");  mod = 0;  }  rewind(stdin);  if (mod == 1) {  qsort(array, index, sizeof(user), user\_struct\_compare\_name);  }  else if (mod == 2) {  qsort(array, index, sizeof(user), user\_struct\_compare\_num);  }  else return;  rewind(stdin);  }  int main(void) {  int choice = 0;  user PhoneBook[USER\_COUNT];  while (1) {  system("cls");  printf("1) 입력\n");  printf("2) 삭제\n");  printf("3) 검색\n");  printf("4) 전체 출력\n");  printf("5) 정렬 후 출력\n");  printf("6) 종료\n");  printf("원하는 번호를 선택하세요 : ");  scanf("%d", &choice);  system("cls");  switch (choice) {  case 1:  printf("입력 선택\n");  user\_add(PhoneBook, USER\_COUNT);  break;  case 2:  printf("삭제 선택\n");  user\_del(PhoneBook, USER\_COUNT);  break;  case 3:  printf("검색 선택\n");  print\_search(PhoneBook, USER\_COUNT);  break;  case 4:  printf("출력 선택\n");  print\_all(PhoneBook, USER\_COUNT);  break;  case 5:  printf("정렬 선택\n");  sort\_array(PhoneBook, USER\_COUNT);  print\_all(PhoneBook, USER\_COUNT);  break;  case 6:  printf("프로그램을 종료합니다.\n");  exit(0);  break;  default:  printf("잘못 선택하셨습니다.\n");  break;  }  rewind(stdin); getchar();  }  return 0;  } |

◆ 설명

main에서는 수를 입력 받아 switch로 모드를 선택합니다. 아래는 포함된 함수의 간단한 설명입니다.

* user\_add 함수는 이름 그대로 유저를 추가합니다.

구조체 배열을 맨 처음부터 검사해서 available이 거짓인 곳이 있으면 해당 배열에 저장합니다.

* user\_del 함수는 유저를 지웁니다.

단, 어차피 available가 0이면 정보가 없는 것으로 인식하기 때문에 available을 0으로 설정만 해주고 끝냅니다.

* print\_all 함수는 모든 유저를 출력합니다.

구조체 배열 맨 처음부터 available이 참인 곳을 출력합니다.

* user\_search\_name / user\_search\_num 함수는 일치하는 이름 또는 전화번호가 배열의 몇 번째에 있는지를 반환합니다.

만약 일치하는 문자열이 없을 경우 -1을 반환합니다.

* print\_search 함수는 user\_search\_name / user\_search\_num 를 사용하여 이름 또는 전화번호를 검색하여 일치하는 결과를 출력합니다.
* user\_struct\_compare\_name / user\_struct\_compare\_num 함수는 strcmp 함수가 user 구조체에서 이름이나 전화번호를 비교할 수 있도록 연결해주는 함수입니다. 반환값은 strcmp와 같습니다.
* sort\_array 함수는 user\_struct\_compare\_name / user\_struct\_compare\_num , qsort 함수를 사용하여 구조체 배열을 정렬합니다.

스크린샷은 다음 페이지에 있습니다.

모든 코드는https://github.com/MU-Software/hs\_c/tree/master/1122에서 확인하실 수 있습니다.

해당 링크는 아래의 QR코드로 이동하실 수 있습니다.

