|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C2\_실습과제\_7주** | **이름 :** | **이갑성** | **학번 :** | **20135151** |

* **강의내용**

1. **변수의 종류**

* 지역 변수 : auto, register, static
* 전역 변수 : static, extern, 자동 초기화
* extern : 다른 파일에서 선언된 전역변수임을 선언
* static : 정적 변수의 초기화는 단 한번만 수행
* static 전역 변수 : extern에 의해 다른 파일에서 참조가 불가능

===> 전역변수를 사용한 stack 구현

#include <stdio.h>

#include <stdlib.h>

#define STACK\_SIZE 100

void push(int item); //insert

int pop(); //pop

void del(); //delete

int peek(); //스택의 top 원소 반환

void printStack(); //스택에 저장된 원소 출력

int stack[STACK\_SIZE];

int top = -1; // 스택의 top의 초기값을-1로 설정, 전역변수

int main(void) {

int item;

printStack();

push(1);printStack(); push(2);printStack(); push(3);printStack();

item = peek();printStack(); printf("peek top => %d", item);

del();printStack();

item = pop();printStack(); printf("\t pop top => %d", item);

item = pop();printStack(); printf("\t pop top => %d", item);

pop();

return 0;}

void push(int item){ // 스택의 삽입 연산

if (top >= STACK\_SIZE - 1) { // 스택이 이미 Full인 경우

printf("\n\n Stack is FULL ! \n");

return;

}

else stack[++top] = item;

}

int pop(){ // 스택의 삭제 후 반환 연산

if (top == -1) { // 현재 스택이 공백인 경우

printf("\n\n Stack is Empty!!\n");

return 0;

}

else return stack[top--];

}

void del(){ // 스택의 삭제 연산

if (top == -1) { // 현재 스택이 공백인 경우

printf("\n\n Stack is Empty !\n");

exit(1);

}

else top--;

}

int peek() { // 스택의 top 원소 검색 연산

if (top == -1){ // 현재 스택이 공백인 경우

printf("\n\n Stack is Empty !\n");

exit(1);

}

else return stack[top];

}

void printStack(){ // 스택 내용 출력 연산

int i;

printf("\n STACK [ ");

for (i = 0; i <= top; i++)

printf("%d ", stack[i]);

printf("] ");

}

1. 문자열 처리

* 한 개 문자열 – 문자열 마지막엔 ‘\0’문자를 자동으로 추가

char str[]=”string”;

char \*str=”string”;

* 두 개 이상의 문자열 처리

char str[][10]={“string1”, “string2”};

char \*str[]={“string1”, “string2”};

1. 문자열 입력과 출력

char str1[20], str2[3][10];

* gets(str1), gets\_s(str2[0],10)
* scanf(“%s”,str1), scanf\_s(“%s”, str2[1], 10)
* puts(str1), puts(str2[2]);

=======> 영문자 자판 연습

void keyboard\_practice(){

int i, total=0;

char output[15], check[15], input[15];

srand(time(NULL));

system("cls");

for(i=0;i<15;i++){

output[i]=rand()%26+97; //임의의 문자 생성

gotoxy(4,9);

printf("%2d번 문자: %c", i+1, output[i]);

input[i]=\_getch(); //문자 입력

if (output[i]==input[i]){ //생성된 문자와 입력된 문자가 같으면 1

total++;

check[i]=1;

}

else //다르면 0

check[i]=0;

}

gotoxy(4,10); printf("자판 연습이 끝났습니다. \n");

gotoxy(4,11); printf("결과를 보려면 아무키나 누르시오. ");

\_getch();

practice\_result(output, input, check, total); //연습결과 출력

}

void practice\_result(char output[], char input[],char check[], int total){

int i;

system("cls");

printf("영문 소문자 자판 연습결과\n\n");

printf(" 출력 문자 입력문자 OX\n");

printf("--------------------------\n");

for(i=0;i<15;i++){

printf("%2d: %c %c ", i+1, output[i], input[i]);

if (check[i]==0)

printf(" X\n");

else

printf(" O\n");}

printf("\n");

printf("맞은 개수:%d(총15)\n", total);

}

1. 구조체

#define size 3

#include <stdio.h>

#include <stdlib.h>

//구조체 정의 & 타입 재정의

struct person{

char name[10]; //이름

int age; //나이

};

typedef struct person PER;

typedef struct {

PER p;

double grade; //평균 학점

} SCORE;

void input(SCORE[]);

void output(SCORE[]);

SCORE max\_score(SCORE []);

int main(void){

SCORE str[size], max;

input(str); //구조체 배열을 매개변수로 전달

output(str);

max = max\_score(str); //평점이 가장 높은 구조체 반환

printf("=== 최고 학점 ======= \n");

printf("%s %d %.1f\n", max.p.name, max.p.age, max.grade);

return 0;

}

void input(SCORE str[]){

int i;

for (i = 0; i<size; i++) {

//공백이 포함되지 않은 문자열 입력

printf("이름 : ");

gets\_s(str[i].p.name, 10);

printf("나이 : ");

scanf\_s("%d", &str[i].p.age);

printf("평균학점 : ");

scanf\_s("%lf", &str[i].grade);

fflush(stdin);

}

}

void output(SCORE str[]){

int i;

printf(" 이름 나이 평균학점\n");

for (i = 0; i<size; i++) {

printf("%s %d %.1f\n", str[i].p.name, str[i].p.age, str[i].grade);

}

printf("========================\n");

}

SCORE max\_score(SCORE str[]){

SCORE max;

int i;

max = str[0]; //구조체 배열 첫번째 원소를 max에 저장

for (i = 1; i<size; i++) {

if (str[i].grade > max.grade) //구조체 멤버 비교

max = str[i];

}

return max; //평점이 가장 높은 구조체 반환

}

* 프로그램 과제

1. ‘영문 자판 연습’ 부분소스를 완성하여 결과를 확인한 후 제시된 조건대로 수정하시오

* 조건1: 20개의 문자열을 배열로 초기화 한다
* 조건2: 조건1의 배열에 있는 문자열을 무작위로 출력한다-난수 사용
* 조건3: 출력된 문자열을 사용자로부터 입력 받는다.
* 조건4: 결과를 출력한다. 즉, 맞춘 수, 입력 받은 문자열, 조건1의 배열에 저장된 문자열

|  |
| --- |
| * 프로그램 소스   #include <stdio.h>  #include <stdlib.h>  #include <time.h>  #include <windows.h>  void keyboard\_practice();  void practice\_result(char randChar[][20], char input[][20], char check[20], int total);  void gotoxy(int x, int y);  int main()  {  srand((long)time(NULL));    keyboard\_practice();  }  void keyboard\_practice() {  int i, total = 0;  int index;  int randIndex;  char output[20][20] = { "Lion", "Bird", "Dog", "Junho", "Jinsu", "Cat", "Snake", "Kabsung", "Mouse", "Red"  , "Sprite", "Coke", "Saida", "Soju", "dump", "love", "abc", "asd", "ewq", "qwer"};  char check[20];  char input[20][20];  char randChar[20][20];    for (i = 0; i < 20; i++) {  index = rand() % 20;  strcpy(randChar[i], output[index]);  gotoxy(4, 9);  printf("%2d번 문자: %s", i + 1, randChar[i]);    gotoxy(4, 11);  printf("정답: ");  gets(input[i]); //문자열 입력  printf("\n");  if (strcmp(randChar[i], input[i]) == 0) { //생성된 문자열와 입력된 문자열가 같으면 0  total++;  check[i] = 1;  }  else //다르면  check[i] = 0;  system("cls");  }  gotoxy(4, 10); printf("자판 연습이 끝났습니다. \n");  gotoxy(4, 11); printf("결과를 보려면 아무키나 누르시오. ");  \_getch();  practice\_result(randChar, input, check, total); //연습결과 출력  }  void practice\_result(char randChar[][20], char input[][20], char check[20], int total) {  int i;  system("cls");  printf("영문 소문자 자판 연습결과\n\n");  printf(" 출력 문자 입력문자 OX\n");  printf("--------------------------\n");  for (i = 0; i<20; i++) {  printf("%2d: %s %s ", i + 1, randChar[i], input[i]);  if (check[i] == 0)  printf(" X\n");  else  printf(" O\n");  }  printf("\n");  printf("맞은 개수:%d(총20)\n", total);  }  void gotoxy(int x, int y) {  COORD Pos = { x - 1, y - 1 };  SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), Pos);  } |
| * 실행결과 |

1. 다음 내용을 참고로 구조체 person을 정의하고 5명의 구조체 배열을 선언하면서 초기화 한 후 제시된 조건대로 처리하시오.

* 구조체 person 멤버 구성: 이름(char), 주민등록번호(char)
* 조건1 : 사용자로부터 입력된 이름과 일치하는 이름이 존재하면 주민등록 번호를 출력 한다. – strcmp()
* 조건2 : 수행하면서 남자인지 여자인지를 출력하시오 – 주민등록번호를 사용하여 남/여 구분
* search\_num() : 구조체 배열과 이름을 매개변수로 받아 주민등록번호를 반환한다
* search\_seong() : 구조체 배열을 매개변수로 받아 남/여를 구분하여 결과를 반환한다.

|  |
| --- |
| * 프로그램 소스   #include <stdio.h>  typedef struct {  char name[10];  char jumin[20];  } person;  int search\_num(person[], char \*, int);  char\* search\_seong(person[], int, int);  int main()  {  person per[5] = { {"이갑성", "940602-1600000"}, {"이준호", "940425-1365873"}, {"심청이", "940827-2367891"},  {"박진수", "941225-1123654"}, {"춘향이", "940222-2684315"} };  char findName[10]; //이름입력 저장변수  int resultIndex; //search\_num메소드에서 리턴한 인덱스 저장변수  int size = sizeof(per) / sizeof(person);  char \*sex; //찾은 사람의 성을 받아 저장하는 포인터 변수  printf("찾고자하는 이름을 입력하세요: ");  gets(findName);  resultIndex = search\_num(per, findName, size);  if (resultIndex == -1)  printf("찾고자하는 사람이 없습니다. \n");  else  printf("이름: %s, 주민등록번호: %s \n", findName, per[resultIndex].jumin);  sex = search\_seong(per, resultIndex, size);  printf("성: %s \n", sex);  }  int search\_num(person per[], char \*name, int size)  {  int i, resultIndex = - 1;    for (i = 0; i < size; i++)  {  if (strcmp(name, per[i].name) == 0)  {  resultIndex = i;  }  }  return resultIndex;  }  char\* search\_seong(person per[], int resultIndex, int size)  {  char sexNum; //성의 넘버 1이면 남자 2이면 여자  sexNum = per[resultIndex].jumin[7];  if (sexNum == '1')  {  return "남자입니다.";  }  else if(sexNum == '2')  {  return "여자입니다.";  }   * } |
| * 실행결과 |

1. 다음 내용을 참고로 구조체 employee를 정의하고, 입력값으로 초기화 한 후 출력하는 프로그램을 작성하시오.

* 조건1 : 크기 5인 구조체 배열을 선언한 후 사용자로부터 입력 받은 값으로 초기화 한다.
* 조건2 : 구조체 employee 멤버의 구성: 개인정보(person), 월급, 연봉
  + 연봉은 월급 \* 12 로 계산
* 조건3 : 개인정보는 1번에서 만든 구조체 person을 이용하고 연봉을 내림차순 또는 오름차순으로 정렬하여 출력
* struct\_array\_sort() – 구조체 배열을 매개변수로 받아 크기순으로 정렬하는 메소드, 반환값 없음
* struct\_array\_write() – 구조체 배열을 매개변수로 받아 배열 원소를 출력하는 메소드, 반환값 없음
* struct\_input() – 사용자로부터 입력 받은 값으로 employee구조체 멤버를 초기화 한 후 반환한다. 매개변수 없음

|  |
| --- |
| * 프로그램 소스   #include <stdio.h>  typedef struct  {  char name[10];  char jumin[20];  }person;  typedef struct  {  person per;  int pay;  int yearPay;  }employee;  void struct\_array\_sort(employee[], int);  void struct\_array\_write(employee[], int);  employee struct\_input();  int main()  {  employee em[5] = {  {{ "이갑성", "940602-1600000" }, 500, 6000},{{ "이준호", "940425-1365873" }, 400, 4800},{{ "심청이", "940827-2367891" }, 300, 3600},  {{ "박진수", "941225-1123654" }, 200, 2400},{{ "춘향이", "940222-2684315" }, 100, 1200}  };  employee userEm[5] = {" "}; //사용자 등록하는 employee구조체 멤버  int size = sizeof(em) / sizeof(employee);  struct\_array\_sort(em, size);  struct\_array\_write(em, size);  struct\_input();  }  void struct\_array\_sort(employee em[], int size)  {  int i, j;  employee temp;  int max = 0;  for (i = 0; i < size-1; i++)  {  max = i;  for (j = i+1; j < size; j++)  {  if (em[j].yearPay > em[max].yearPay)  {  max = j;  }  }  temp = em[i];  em[i] = em[max];  em[max] = temp;  }  }  void struct\_array\_write(employee em[], int size)  {  int i;  for (i = 0; i < size; i++)  {  printf("=========%d번========= \n", i + 1);  printf("이름: %s \n", em[i].per.name);  printf("주민등록번호: %s \n", em[i].per.jumin);  printf("월급: %d \n", em[i].pay);  printf("연봉: %d \n", em[i].yearPay);  }  }  employee struct\_input()  {  employee userEm;  printf("사원의 이름을 입력하세요: ");  gets(userEm.per.name);  printf("\n사원의 주민등록번호를 입력하세요: ");  gets(userEm.per.jumin);  printf("\n사원의 월급을 입력하세요: ");  scanf("%d", &userEm.pay);  printf("\n사원의 연봉을 입력하세요: ");  scanf("%d", &userEm.yearPay);   * } |
| * 실행결과 |

1. 도서 카드 목록을 처리하는 프로그램을 작성하시오.

* 조건1: 처리해야 할 최대 카드 갯수는 100개
* 조건2: 구조체 배열과 이전에 참조된 배열 위치를 저장하기 위한 변수는 전역변수로 선언하여 사용할 것
* 조건 3: 메뉴구성은 다음과 같으며 4가 입력될 때 까지 반복

1 : 레코드 입력 2. 저장이름에 의한 검색 3. 제목에 의한 검색 4. 종료

* 다음과 같은 함수로 구성
  + int menu(void); //선택된 메뉴 반환
  + void display(int i); //해당 레코드 출력
  + void author\_search(void); //사용자로부터 입력 받은 지은이이름과 일치하는 레코드 출력
  + void title\_search(void); //사용자로부터 입력 받은 책 제목과 일치하는 레코드 출력
  + void enter(void); //사용자로부터 입력 받은 레코드를 구조체 배열에 저장
* 구조체는 다음과 같이 구성

struct book\_type {

unsigned date; /\* 발행일 \*/

unsigned pages; /\* 책의 쪽수 \*/

};

struct catalog {

char name[40]; /\* 지은이 이름 \*/

char title[40]; /\* 제목 \*/

char pub[40]; /\* 출판사 \*/

struct book\_type book; /\* 기술적인 정보 \*/

}

|  |
| --- |
| <소스>  #include <stdio.h>  #include <stdlib.h>  typedef struct {  unsigned int date; /\* 발행일 \*/  unsigned int pages; /\* 책의 쪽수 \*/  }bookInfo;  typedef struct {  char name[40]; /\* 지은이 이름 \*/  char title[40]; /\* 제목 \*/  char pub[40]; /\* 출판사 \*/  bookInfo info; /\* 기술적인 정보 \*/  }catalog;  int menu(void); //선택된 메뉴 반환  void display(int i); //해당 레코드 출력  void author\_search(void); //사용자로부터 입력 받은 지은이이름과 일치하는 레코드 출력  void title\_search(void); //사용자로부터 입력 받은 책 제목과 일치하는 레코드 출력  void enter(void); //사용자로부터 입력 받은 레코드를 구조체 배열에 저장  //전역변수 선언  catalog ca[100];  int index = 1;  int main()  {  int selectMenu; // 선택된 메뉴 번호 저장 변수  int num;  do {    selectMenu = menu();  if (selectMenu == 1)  {  enter();  }  else if (selectMenu == 2)  {  author\_search();  }  else if (selectMenu == 3)  {  title\_search();  }  } while (selectMenu != 4);  }  int menu(void)  {    int select;  printf("===========메뉴===========\n");  printf("1. 레코드 입력 \n");  printf("2. 저장이름에 의한 검색 \n");  printf("3. 제목에 의한 검색 \n");  printf("4. 종료 \n");    printf("메뉴 입력 >>");  scanf("%d", &select);    return select;  }  void display(int i)  {  if (i > index || i < 0)  {  printf("%d번째 레코드 값은 없습니다. \n");  }  else  {  printf("책의 발행일 : %d \n", ca[i].info.date);  printf("책의 쪽수: %d \n", ca[i].info.pages);  printf("지은이 이름: %s \n", ca[i].name);  printf("제목: %s \n", ca[i].title);  printf("출판사: %s \n", ca[i].pub);  }    }  void author\_search(void)  {  char findAuthor[20];  int i;  int find = -1;  system("cls");  printf("찾고자하는 저자의 이름을 입력하세요: ");  scanf("%s", findAuthor);  for (i = 1; i <= index; i++)  {  if (strcmp(findAuthor, ca[i].name) == 0)  {  printf("책의 발행일 : %d \n", ca[i].info.date);  printf("책의 쪽수: %d \n", ca[i].info.pages);  printf("지은이 이름: %s \n", ca[i].name);  printf("제목: %s \n", ca[i].title);  printf("출판사: %s \n", ca[i].pub);    find = 0;  }  }  if (find == -1)  {  printf("찾고자하는 제목에 해당되는 책은 없습니다. \n");  }  }  void title\_search(void)  {  char findBook[20];  int i;  int find = -1;  system("cls");  printf("찾고자하는 책 제목을 입력하세요: ");  scanf("%s", findBook);  for (i = 1; i <= index; i++)  {  if (strcmp(ca[i].title , findBook) == 0)  {  printf("책의 발행일 : %d \n", ca[i].info.date);  printf("책의 쪽수: %d \n", ca[i].info.pages);  printf("지은이 이름: %s \n", ca[i].name);  printf("제목: %s \n", ca[i].title);  printf("출판사: %s \n", ca[i].pub);  find = 0;  }  }  if (find == -1)  {  printf("찾고자하는 제목에 해당되는 책은 없습니다. \n");  }  }  void enter(void)  {  system("cls");  int go = 0;  int i;  printf("책의 발행일 : ");  scanf("%d", &ca[index].info.pages);  printf("책의 쪽수: ");  scanf("%d", &ca[index].info.date);  printf("지은이 이름: ");  scanf("%s", ca[index].name);  printf("제목: ");  scanf("%s", ca[index].title);  printf("출판사: ");  scanf("%s", ca[index].pub);  index++;  } |
| <결과> |