|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C2\_실습과제\_9주** | **이름 :** | **이갑성** | **학번 :** | **20135151** |

* **강의내용**

1. 구조체 포인터

struct lecture{

char name[20]; **//강좌명**

int type; **//강좌구분 0:교양, 1:일반선택, 2:전공필수, 3:전공선택**

int credit; **//학점**

int hours; **//시수** };

typedef struct lecture lecture; **//타입 재정의**

char \*lectype[] = {"교양", "일반선택", "전공필수", "전공선택"};

char \*head[] = {"강좌명", "강좌구분", "학점", "시수"};

int main(){

lecture os = {"운영체제", 2, 3, 3};

lecture c = {"C프로그래밍", 3, 3, 4};

**lecture \*p = &os;** **//구조체 포인터 초기화**

printf("%10s %12s %6s %6s\n", head[0], head[1], head[2], head[3]);

**//포인터를 이용한 멤버 접근: 구조체포인터변수 -> 멤버변수, (\*구조체 포인터 변수).멤버변수**

printf("%12s %10s %5d %5d\n", **p->name, lectype[p->type], p->credit, p->hours**);

**p = &c;**

printf("%12s %10s %5d %5d\n", **(\*p).name, lectype[(\*p).type], (\*p).credit, (\*p).hours**);

return 0;

}

1. 구조체 배열과 포인터

**//구조체 lecture 사용**

int main(void){

lecture course[] = {{"인간과사회", 0, 2, 2}, {"경제학개론", 1, 3, 3}, {"자료구조", 2, 3, 3}, {"모바일프로그래밍", 2, 3, 4}, {"고급C프로그래밍", 3, 3, 4}};

**lecture \*p = course;** **//구조체 배열을 가리키는 구조체 배열 포인터 선언 & 초기화**

int i;

printf("%12s %12s %6s %6s\n", head[0], head[1], head[2], head[3]);

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

**//구조체 배열 포인터를 사용한 멤버 참조, (\*(p+i)).name 로도 참조 가능**

printf("%16s %10s %5d %5d\n", **(p+i)->name, lectype[(p+i)->type],(p+i)->credit, (p+i)->hours);**

**//또는 printf("%16s %10s %5d %5d\n", p[i].name, lectype[p[i].type], p[i].credit, p[i].hours); 도 가능**

return 0;

}

* 프로그램 과제

1. 7주 실습과제 3번문제를 다음과 같이 수정하시오.

* struct\_array\_sort() – 구조체 배열을 구조체 포인터 매개변수로 받아 크기순으로 정렬하는 메소드, 반환값 없음, 구조체 배열원소 참조 시 ->을 사용할 것
* struct\_array\_write() – 구조체 배열을 구조체 포인터 매개변수 받아 배열 원소를 출력하는 메소드, 반환값 없음, 구조체 배열 원소 참조 시 \*을 사용할 것

|  |
| --- |
| * 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);  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);  }  void struct\_array\_sort(employee \*em, int size)  {  int i, j, max;  employee temp;  for (i = 0; i < size - 1; i++)  {  max = i;  for (j = i + 1; j < size; j++)  {  if ((em+max)->pay < (em + j)->pay)  {  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);  }  } |
| * 실행결과 |

1. 은행 계좌를 생성하고 입출금을 처리하는 프로그램을 작성하시오. 제시된 조건 외는 본인 임의로 처리할 것

* 은행 계좌 (계좌번호, 비밀번호, 이름, 잔액)를 표현하는 bankaccount 구조체 선언
* 계좌 생성 시 bankaccount 구조체를 사용하여 계좌 정보 입력(사용자로 부터)
* 비밀번호 검사, 입금, 출금 관련 함수 작성(입출금의 경우 잔액을 리턴 값으로 받음)
* 입금과 출금의 경우 해당 계좌번호를 사용자로부터 입력 받는다.
* 처리할 레코드 개수는 최대 100개
* struct bankaccount createaccount(); //생성된 계좌 반환
* int checkpasswd(char \*confirmpass, char \*savedpass); //비밀번호 검사
* int deposit(int balance, int amount); // 입금, 비밀번호 검사는 checkpasswd() 사용
* int withdraw(int balance, int amount); // 출금, 비밀번호 검사는 checkpasswd() 사용
* void display(struct bankaccount); //입출금 후 해당 계좌 출력

int main(void) {

do {

printf("\n 종료(0), 계좌생성(1), 입금(2), 출금(3) : ");

scanf("%d", &state);

if( !state ) break; //0이 입력되면 종료

switch(state){

case 1: //계좌생성

case 2: //입금

case 3: //출금

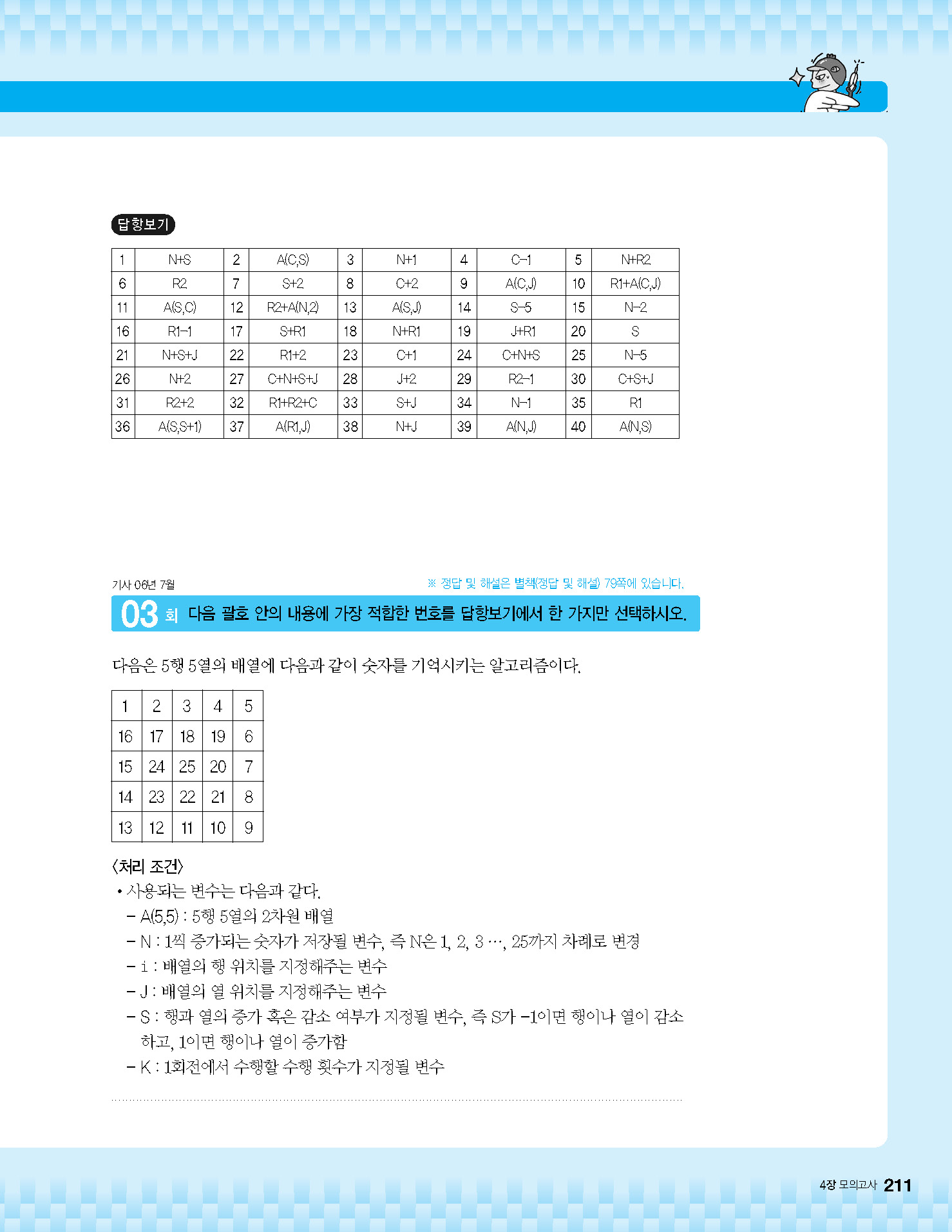
} while(1);

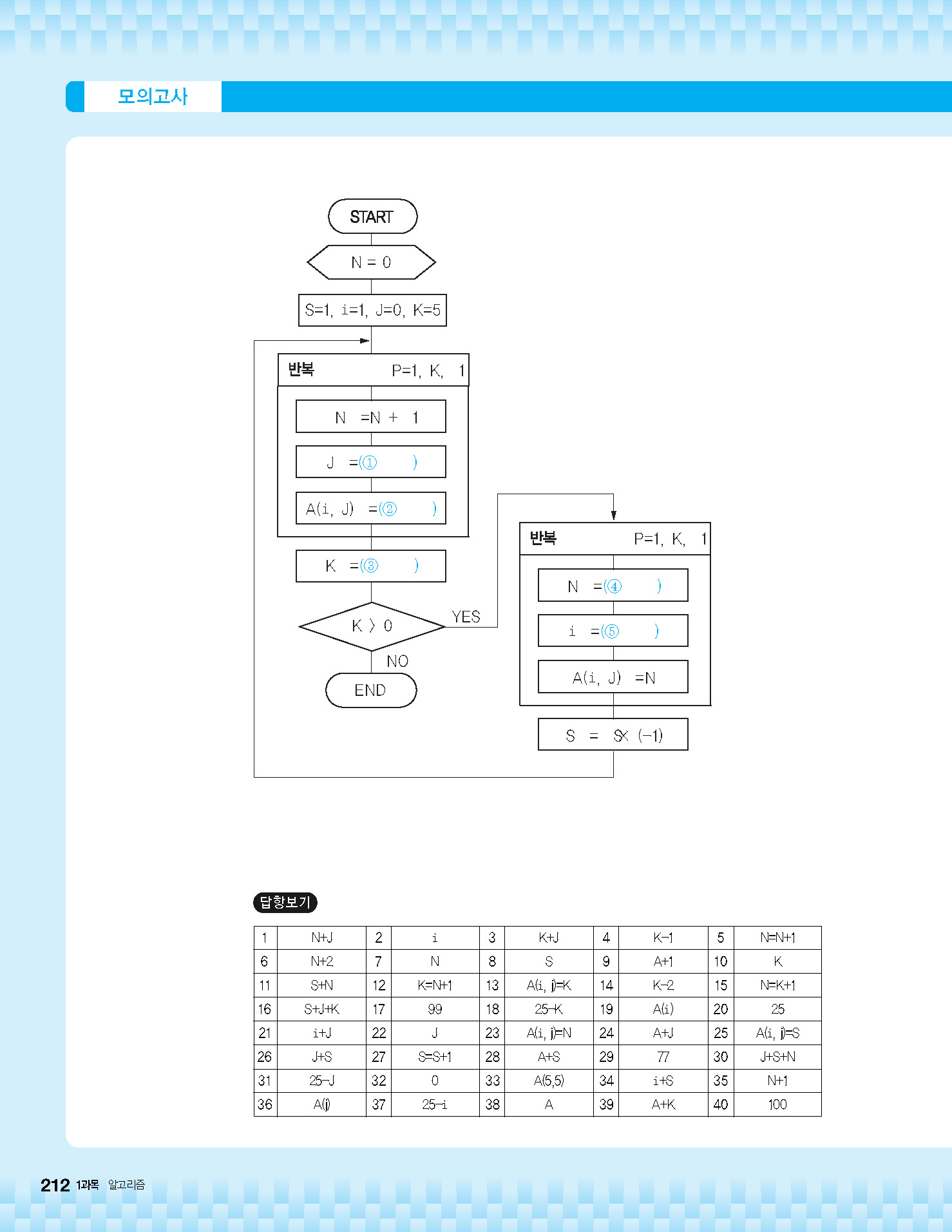
return 0;

}

|  |
| --- |
| * 프로그램 소스   #include <stdio.h>  typedef struct {  char account[20]; // 계좌번호  char keyword[20]; // 비밀번호  char name[20]; // 이름  int rest; // 잔돈  }bankaccount;  bankaccount createaccount(); //생성된 계좌 반환  int checkpasswd(int); //비밀번호 검사  void deposit(int); // 입금, 비밀번호 검사는 checkpasswd() 사용  void withdraw(int); // 출금, 비밀번호 검사는 checkpasswd() 사용  void display(int); //입출금 후 해당 계좌 출력  int checkAccount(); // 계좌번호 입력해서 체크하는 함수  bankaccount list[100];  int listSize = 0;  int check = 0; // 계좌번호가 있는지 알려주는 변수 0이면 없고 1이면 있는 계좌번호  int isRight = 0; // 비밀번호가 맞는지 알려주는 변수 0이면 틀리고 1이면 맞다  int main(void) {  int state;  int targetIndex; // 계좌번호로 통해 찾은 인덱스 값이 저장된다.  do {  check = 0; // 다시 실행 될때마다 초기화  isRight = 0; // 다시 실행 될때마다 초기화  printf("\n 종료(0), 계좌생성(1), 입금(2), 출금(3) : ");  scanf("%d", &state);  if (!state) break; //0이 입력되면 종료  switch (state)  {  case 1: //계좌생성  list[listSize] = createaccount();  list[listSize].rest = 0;  listSize = listSize + 1;  break;  case 2: //입금  targetIndex = checkAccount();  if (isRight == 0)  break;  deposit(targetIndex);  if (check == 0)  break;  display(targetIndex);  break;  case 3: //출금  targetIndex = checkAccount();  if (isRight == 0)  break;  withdraw(targetIndex);  if (check == 0)  break;  display(targetIndex);  break;  }  } while (1);    }  bankaccount createaccount()  {  bankaccount creat;  printf("============ 계좌 생성 ============ \n");  printf("계좌번호 : ");  scanf("%s", creat.account);  printf("비밀번호 : ");  scanf("%s", creat.keyword);  printf("이름 : ");  scanf("%s", creat.name);  printf("=== 생성된 계좌 정보 === \n");  printf("계좌번호: %s, 비밀번호: %s, 이름: %s \n", creat.account, creat.keyword, creat.name);  return creat;  }  int checkAccount()  {  char num[20];  int i;  int targetIndex;  printf("계좌번호를 입력해주세요: ");  scanf("%s", num);    for (i = 0; i <= listSize; i++)  {  if (strcmp(list[i].account, num) == 0)  {  targetIndex = i;  checkpasswd(targetIndex);  check = 1;  }  }  if (check == 0)  {  printf("해당 계좌번호는 없습니다. \n");  }  return targetIndex;  }  int checkpasswd(int targetIndex)  {  char key[20];  printf("비밀번호를 입력해주세요: ");  scanf("%s", key);  if (strcmp(list[targetIndex].keyword, key) == 0)  {  isRight = 1;  }  else  {  printf("비밀번호가 틀렸습니다. \n");  }  return isRight;  }  void deposit(int targetIndex)  {  int money;  printf("======== 입금 ========= \n");  printf("입금할 금액을 입력하세요: ");  scanf("%d", &money);    list[targetIndex].rest = list[targetIndex].rest + money;  }  void withdraw(int targetIndex)  {  int money;  printf("======== 출금 ========= \n");  printf("출금할 금액을 입력하세요: ");  scanf("%d", &money);  if (list[targetIndex].rest == 0 || list[targetIndex].rest < money)  printf("출금할 금액의 한도를 넘겼습니다.");  else  list[targetIndex].rest = list[targetIndex].rest - money;  }  void display(int targetIndex)  {  printf("인덱스 : %d \n", targetIndex);  printf("이름: %s \n", list[targetIndex].name);  printf("계좌번호: %s \n", list[targetIndex].account);  printf("남은 금액: %d \n", list[targetIndex].rest);   * } |
| * 실행결과 |

1. 다음과 같이 이차원 배열을 초기화 한 후 출력하는 프로그램을 작성하시오





J+S

N

K-1

N+1

i+S

|  |
| --- |
| * 프로그램 소스   #include <stdio.h>  int main()  {  int count = 1;  int x = 0, y = -1;  int i, j;  int k = 5, s = 1;  int arr[5][5];  while (k > 0)  {  for (i = 0; i < k; i++)  {  y = y + s;  arr[x][y] = count++;  }  k--;    for (j = 0; j < k; j++)  {  x = x + s;  arr[x][y] = count++;  }  s = s \* (-1);  }  for (i = 0; i < 5; i++)  {  for (j = 0; j < 5; j++)  {  printf("%d ", arr[i][j]);  }  printf("\n");  }  } |
| * 실행결과 |