정리노트 #8

예제 10-9

《핵심》

포인터를 이용해 문자열 선언

-char \*포인터명 = "문자열"

포인터명 -> pC

문자열 -> C programming

- 문자열을 변환기호를 이용해 출력

char \*pC = "C programming";

printf("%s₩n", pC);

- 반복문을 이용해 출력

while(\*pC)

      printf("%c", \*pC++);

printf("₩n");

예제 10-10

《핵심》

문자열을 동시에 여러 개 만들어야 하는 경우

char \*pStr[3] = {"문자열1", "문자열2", "문자열3"};

문자열 = english, math, korean

char \*pStr[] = {"english", "math", "korean"};

int i;

- 포인터 배열을 이용해서 문자열 출력

for(i=0; i<3; i++)

printf("pStr[%d] = %s\n", i, pStr[i]);

- 2차원 배열을 이용해서 문자열 출력

for(i=0; i<3; i++)

printf("subject[%d] = %s\n", i, subject[i]);

예제 10-14

핵심

strlen()함수는 ￦0을 제외한 문자열의 크기를 반환

strcmp()함수는 두 문자열을 비교

strcpy()함수는 두 문자열을 복사

strcat()함수는 첫 번째 문자열에 두 번째 문자열을 연결

char cmp1[40] = " C programming";

char cmp2[ ] = "Java programming";

char cmp3[ ] = "C programming";

char str[ ] = "is easy";

//문자열을 입력

int length, i;

length = strlen(cmp1);

//strlen 함수 이용

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

15 printf("%c", cmp1[i]);

printf("cmp1과 cmp2는 서로 %s\n", strcmp(cmp1, cmp2) ? "같지 않다." : "같다.");

printf("cmp1과 cmp3은 서로 %s\n", strcmp(cmp1, cmp3) ? "같지 않다." : "같다.");

//strcmp 함수 이용해서 비교 후 같지 않다와 같다로 출력

printf("cmp1에 cmp2를 복사하면\"%s\"이 출력된다.\n",

strcpy(cmp1, cmp2));

//strcpy 함수를 이용해서 복사 후 출력

21 printf("cmp1에 str을 연결하면\"%s\"가 출력된다.\n", strcat(cmp1, str));

//strcat 함수를 이용해서 연결 후 출력

ch11

예제 11-1

핵심

구조체의 정의, 구조체의 선언

-구조체의 정의

struct list {

char name;

char gender;

int age;

};

구조체 키워드 -> struct

구조체명 -> list

char name;

char gender;

-> 문자형 멤버

int age;

-> 정수형 멤버

- 구조체의 선언

int main(void)

{

struct list st1 = {'T', 'M', 25};

printf("구조체 list의 크기는 %d이다.\n", sizeof(struct list));

printf("구조체 객체 st1의 크기는 %d이다.\n", sizeof(st1));

예제 11-3

핵심

-구조체에서 할당 연산자 사용

struct list {

char name;

char gender;

int age;

} ;

//구조체의 정의

int main(void)

{

struct list st1 = {'T', 'M', 25};

struct list st2, st3 = {'P', 'F', 30};

//구조체의 선언

st2 = st1;

st3.name = st1.name;

st3.gender = st1.gender;

//할당 연산자 사용

printf(" 이름 성별 나이\n");

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

printf("st1 %c %c %d\n", st1.name, st1.gender, st1.age);

printf("st2 %c %c %d\n", st2.name, st2.gender, st2.age);

printf("st3 %c %c %d\n", st3.name, st3.gender, st3.age);

예제 11-7

핵심

- 구조체를 인자로 받는 함수

- 3차원상의 한 점과 원점의 거리를 구하는 함수

double Distance1(struct ThreeDime a)

{

double d;

d = sqrt(a.x \* a.x + a.y \* a.y + a.z \* a.z);

return d;

}

- 3차원상의 두 점 간 거리를 구하는 함수

double Distance2(struct ThreeDime a, struct ThreeDime b)

{

double d;

d = sqrt((a.x - b.x) \* (a.x - b.x) + (a.y - b.y) \* (a.y - b.y) + (a.z - b.z) \*

(a.z - b.z));

return d;

}
3차원상의 한 점을 원점에 대칭시킨 점을 구하는 함수

struct ThreeDime SymOri(struct ThreeDime a)

{

a.x = -a.x;

a.y = -a.y;

a.z = -a.z;

return a;

}

struct ThreeDime

{

double x;

double y;

double z;

};

typedef struct ThreeDime ThreeDime;

//포인터 선언

double Distance1(ThreeDime a)

{

double d;

d = sqrt(a.x \* a.x + a.y \* a.y + a.z \* a.z);

return d;

}

//3차원상의 한 점과 원점의 거리를 구하는 함수 사용