/\*

240320 구조체 예시

\*/

#include <stdio.h>

#pragma warning(disable: 4996)

typedef struct studentTag {

char name[10]; // 문자배열로 된 이름

int age; // 나이를 나타내는 정수값

double gpa; // 평균평점을 나타내는 실수값

} student;

int main(void)

{

student a = { "kim", 20, 4.3 };

student b = { "park", 21, 4.2 };

printf("%s\t %d\t %f\n", a.name, a.age, a.gpa);

printf("%s\t %d\t %f\n", b.name, b.age, b.gpa);

return 0;

}

/\*

240320 포인터 예시

\*/

#include <stdio.h>

#pragma warning(disable: 4996)

void swap(int\* px, int\* py)

{

int tmp;

tmp = \*px;

\*px = \*py;

\*py = tmp;

}

int main(void)

{

int a = 1, b = 2;

printf("swap을 호출하기 전: a=%d, b=%d\n", a, b);

swap(&a, &b);

printf("swap을 호출한 다음: a=%d, b=%d\n", a, b);

return 0;

}

/\*

240320 배열과 포인터 예시

\*/

#include <stdio.h>

#pragma warning(disable: 4996)

#define SIZE 6

void get\_integers(int list[])

{

printf("6개의 정수를 입력하시오: ");

for (int i = 0; i < SIZE; ++i) {

scanf("%d", &list[i]);

}

}

int cal\_sum(int list[])

{

int sum = 0;

for (int i = 0; i < SIZE; ++i) {

sum += \*(list + i);

}

return sum;

}

int main(void)

{

int list[SIZE];

get\_integers(list);

printf("합 = %d \n", cal\_sum(list));

return 0;

}

/\*

240320 동적 메모리 할당

\*/

// MALLOC.C: malloc을 이용하여 정수 10를 저장할 수 있는 동적 메모리를

// 할당하고 free를 이용하여 메모리를 반납한다.

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

#pragma warning(disable: 4996)

#define SIZE 10

int main(void)

{

int\* p;

p = (int\*)malloc(SIZE \* sizeof(int));

if (p == NULL) {

fprintf(stderr, "메모리가 부족해서 할당할 수 없습니다.\n");

exit(1);

}

for (int i = 0; i < SIZE; i++)

p[i] = i;

for (int i = 0; i < SIZE; i++)

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

free(p);

return 0;

}

/\*

240320 동적 메모리 할당 2

\*/

// MALLOC.C: malloc을 이용하여 정수 10를 저장할 수 있는 동적 메모리를

// 할당하고 free를 이용하여 메모리를 반납한다.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct studentTag {

char name[10]; // 문자배열로 된 이름

int age; // 나이를 나타내는 정수값

double gpa; // 평균평점을 나타내는 실수값

} student;

int main(void)

{

student\* s;

s = (student\*)malloc(sizeof(student));

if (s == NULL) {

fprintf(stderr, "메모리가 부족해서 할당할 수 없습니다.\n");

exit(1);

}

strcpy(s->name, "Park");

s->age = 20;

s->gpa = 4.2;

printf("%s\t %d\t %f", s->name, s->age, s->gpa); // \t -> tab키 간격

free(s);

return 0;

}