/\*

#include <stdio.h>

int SumArray(int\* pA, int Size);

int main(void)

{

int a[] = { 10,5,15,25,7 };

int Sum;

Sum = SumArray(a, 5);

printf("배열의 합 : %d \n", Sum);

return 0;

}

int SumArray(int\* pA, int Size)

{

int result = 0, i;

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

// result += pA[i];

// result += \*(pA + i);

// result += \*pA ++;

return result;

}

//Clemente font

\*/

/\*

#include <stdio.h>

int main(void)

{

char \*pC = "C programming";

printf("문자열을 변환기호를 이용해서 출력\n");

printf("%s\n", pC);

printf("반복문을 이용해 한 글자씩 출력\n");

while (\*pC)

printf("%c", \*pC++);

printf("\n");

return 0;

}

\*/

/\* while문 조건식 안에서 문자열을 가리킨다면

null문자를 만날 때까지 다음 식을 진행하라는 뜻임.\*/

/\*

#include <stdio.h>

#include <ctype.h>

int main(void)

{

char \*pC = "c language";

while (\*pC)

{

printf("%c", toupper(\*pC++));

}

printf("\n");

return 0;

}

\*/

/\*

#include <stdio.h>

int strleng(char \*ch);

int main(void)

{

int len;

char str[] = "c language and data structure";

len = strleng(str);

printf("문자열의 크기는 %d다.\n", len);

return 0;

}

int strleng(char\*ch)

{

int index;

for (index = 0; \*(ch + index) != '\0'; index++);

return index;

}

\*/

/\* strcpy 많이 씀.

java string에서 특정 문자로 배열 쪼개기? -> split. \*/

/\*

#include <stdio.h>

#include <string.h>

int main(void)

{

char cmp1[40] = "C programming";

char cmp2[] = "Java programming";

char cmp3[] = "C programming";

char str[] = "is easy";

int length, i;

length = strlen(cmp1);

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

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

printf("\n");

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

printf("cmp1과 cmp2는 서로 %s\n", strcmp(cmp1, cmp3) ? "같지 않다." : "같다.");

printf("cmp1에 cmp2를 복사하면\" %s\"이 출력된다.\n", strcpy(cmp1, cmp2));

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

return 0;

}

\*/

/\*변수만 들어가 있는 class라고 이해하기\*/

/\*

#include <stdio.h>

struct list{

char name;

char gender;

int age;

};

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));

return 0;

}

\*/

/\*

#include <stdio.h>

struct students {

char name[20];

char address[30];

int age;

};

typedef struct students student;

int main(void)

{

student st1 = { "Moon", "Seoul",25 };

struct students st2 = { "John","London",30 };

printf("학생 1은 성이 %s이고 %s에 살며 %d살이다.\n", st1.name, st1.address, st1.age);

printf("학생 2는 성이 %s이고 %s에 살며 %d살이다.\n", st2.name, st2.address, st2.age);

return 0;

}

\*/

//typedef 를 앞에 붙힐 수도 있지만 붙히면 객체 이름을 빠지고 밑에 이름만 선언함.

//점 간의 거리 구하기 안봐도 됨. 이해를 돕기 위한 것.

//화살표를 붙히면 내부로 접근 가능.

/\*

#include <stdio.h>

struct ThreeDime {

double x;

double y;

double z;

};

typedef struct ThreeDime ThreeDime;

int main(void)

{

ThreeDime A1 = { 3,1,8 };

ThreeDime\*pA1 = &A1;

printf("3차원 점 A1의 x : %.2lf, y : %2If, z : %2If이다. \n", pA1->x, pA1->y, pA1->z);

return 0;

}

\*/

// 구조체를 인자로 받는 함수 -> callbyvalue형식임.

// 구조체 포인터를 인자로 받는 함수 -> callbyref형식임.

/\*열거형은 struct와 거의 흡사함.

상수값화 함.\*/

/\*

#include <stdio.h>

int main(void)

{

enum { yellow, red, blue, green } color;

printf("원하는 색을 입력하세요.\n");

printf("0번 : 노란색, 1번 : 빨간색\n");

printf("2번 : 파란색, 3번 : 초록색\n");

scanf("%d", &color);

if (color = = yellow) printf("노란색입니다.\n");

else if (color = = red) printf("빨간색입니다.\n");

else if (color = = blue) printf("파란색입니다.\n");

else if (color = = green) printf("초록색입니다.\n");

return 0;

}

\*/