// 16-1

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int\* pi;

double\* pd;

pi = (int\*)malloc(sizeof(int));

if (pi == NULL) {

printf("# 메모리가 부족합니다.\n");

exit(1);

}

pd = (double\*)malloc(sizeof(double));

\*pi = 10;

\*pd = 3.4;

printf("정수형으로 사용 : %d\n", \*pi);

printf("실수형으로 사용 : %.1lf\n", \*pd);

free(pi);

free(pd);

return 0;

}

// 16-2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int\* pi;

int i, sum = 0;

pi = (int\*)malloc(5 \* sizeof(int));

if (pi == NULL)

{

printf("메모리가 부족합니다!\n");

exit(1);

}

printf("다섯 명의 나이를 입력하세요 : ");

for (i = 0; i < 5; i++)

{

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

sum += pi[i];

}

printf("다섯 명의 평균 나이 : % .1lf\n", (sum / 5.0)); // 평균 나이 출력

free(pi);

return 0;

}

// 16-3

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int\* pi;

int size = 5;

int count = 0;

int num;

int i;

pi = (int\*)calloc(size, sizeof(int));

while (1)

{

printf("양수만 입력하세요 = > ");

scanf("%d", &num);

if (num <= 0) break;

if (count == size

{

size += 5;

pi = (int\*)realloc(pi, size \* sizeof(int));

}

pi[count++] = num;

}

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

{

printf("%5d", pi[i]);

}

free(pi);

return 0;

}

// 16-4

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(void)

{

char temp[80];

char\* str[3];

int i;

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

{

printf("문자열을 입력하세요 : ");

gets(temp);

str[i] = (char\*)malloc(strlen(temp) + 1);

strcpy(str[i], temp);

}

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

{

printf("%s\n", str[i]);

}

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

{

free(str[i]);

}

return 0;

}

// 16-5

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void print\_str(char\*\* ps);

int main(void)

{

char temp[80];

char\* str[21] = { 0 };

int i = 0;

while (i < 20)

{

printf("문자열을 입력하세요 : ");

gets(temp);

if (strcmp(temp, "end") == 0) break;

str[i] = (char\*)malloc(strlen(temp) + 1);

strcpy(str[i], temp);

i++;

}

print\_str(str);

for (i = 0; str[i] != NULL; i++)

{

free(str[i]);

}

return 0;

}

void print\_str(char\*\* ps)

{

while (\*ps != NULL)

{

printf("%s\n", \*ps);

ps++;

}

}

// 16-6

#include <stdio.h>

int main(int argc, char\*\* argv)

{

int i;

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

{

printf("%s\n", argv[i]);

}

return 0;

}

// 17-1

#include <stdio.h>

struct student

{

int num;

double grade;

};

int main(void)

{

struct student s1;

s1.num = 2;

s1.grade = 2.7;

printf("학번 : %d\n", s1.num);

printf("학점 : %.1lf\n", s1.grade);

return 0;

}

// 17-2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct profile

{

char name[20];

int age;

double height;

char\* intro;

};

int main(void)

{

struct profile yuni;

strcpy(yuni.name, "서하윤");

yuni.age = 17;

yuni.height = 164.5;

yuni.intro = (char\*)malloc(80);

printf("자기소개 : ");

gets(yuni.intro);

printf("이름 : %s\n", yuni.name);

printf("나이 : %d\n", yuni.age);

printf("키 : %.1lf\n", yuni.height);

printf("자기소개 : %s\n", yuni.intro);

free(yuni.intro);

return 0;

}

// 17-3

#include <stdio.h>

struct profile

{

int age;

double height;

};

struct student

{

struct profile pf;

int id;

double grade;

};

int main(void)

{

struct student yuni;

yuni.pf.age = 17;

yuni.pf.height = 164.5;

yuni.id = 315;

yuni.grade = 4.3;

printf("나이 : %d\n", yuni.pf.age);

printf("키 : %.1lf\n", yuni.pf.height);

printf("학번 : %d\n", yuni.id);

printf("학점 : %.1lf\n", yuni.grade);

return 0;

}

// 17-4

#include <stdio.h>

struct student

{

int id;

char name[20];

double grade;

};

int main(void)

{

struct student s1 = { 315, "홍길동", 2.4 },

s2 = { 316, "이순신", 3.7 },

s3 = { 317, "세종대왕", 4.4 };

struct student max;

max = s1;

if (s2.grade > max.grade) max = s2;

if (s3.grade > max.grade) max = s3;

printf("학번 : %d\n", max.id);

printf("이름 : %s\n", max.name);

printf("학점 : %.1lf\n", max.grade);

return 0;

}

// 17-5

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

struct vision

{

double left;

double right;

};

struct vision exchange(struct vision robot);

int main(void)

{

struct vision robot;

printf("시력 입력 : ");

scanf("%lf%lf", &(robot.left), &(robot.right));

robot = exchange(robot);

printf("바뀐 시력 : %.1lf %.1lf\n", robot.left, robot.right);

return 0;

}

struct vision exchange(struct vision robot)

{

double temp;

temp = robot.left;

robot.left = robot.right;

robot.right = temp;

return robot;

}

// 17-6

#include <stdio.h>

struct score

{

int kor;

int eng;

int math;

};

int main(void)

{

struct score yuni = { 90, 80, 70 };

struct score\* ps = &yuni;

printf("국어 : %d\n", (\*ps).kor);

printf("영어 : %d\n", ps->eng);

printf("수학 : %d\n", ps->math);

return 0;

}

// 17-7

#include <stdio.h>

struct address

{

char name[20];

int age;

char tel[20];

char addr[80];

};

int main(void)

{

struct address list[5] = {

{"홍길동", 23, "111 - 1111", "울릉도 독도"},

{"이순신", 35, "222 - 2222", "서울 건천동"},

{"장보고", 19, "333 - 3333", "완도 청해진"},

{"유관순", 15, "444 - 4444", "충남 천안"},

{"안중근", 45, "555 - 5555", "황해도 해주"}

};

int i;

for (i = 0; i < 5; i++)

{

printf("%10s%5d%15s%20s\n",

list[i].name, list[i].age, list[i].tel, list[i].addr);

}

return 0;

}

// 17-8

#include <stdio.h>

struct address

{

char name[20];

int age;

char tel[20];

char addr[80];

};

void print\_list(struct address\* lp);

int main(void)

{

struct address list[5] = {

{"홍길동", 23, "111 - 1111", "울릉도 독도"},

{"이순신", 35, "222 - 2222", "서울 건천동"},

{"장보고", 19, "333 - 3333", "완도 청해진"},

{"유관순", 15, "444 - 4444", "충남 천안"},

{"안중근", 45, "555 - 5555", "황해도 해주"}

};

print\_list(list);

return 0;

}

void print\_list(struct address\* lp)

{

int i;

for (i = 0; i < 5; i++)

{

printf("%10s%5d%15s%20s\n",

(lp + i)->name, (lp + i)->age, (lp + i)->tel, (lp + i)->addr);

}

}

// 17-9

#include <stdio.h>

struct list

{

int num;

struct list\* next;

};

int main(void)

{

struct list a = { 10, 0 }, b = { 20, 0 }, c = { 30, 0 };

struct list\* head = &a, \* current;

a.next = &b;

b.next = &c;

printf("head->num : %d\n", head->num);

printf("head->next->num : %d\n", head->next->num);

printf("list all : ");

current = head;

while (current != NULL)

{

printf("%d ", current->num);

current = current->next;

}

printf("\n");

return 0;

}

// 17-10

#include <stdio.h>

union student

{

int num;

double grade;

};

int main(void)

{

union student s1 = { 315 };

printf("학번 : %d\n", s1.num);

s1.grade = 4.4;

printf("학점 : %.1lf\n", s1.grade);

printf("학번 : %d\n", s1.num);

return 0;

}

// 17-11

#include <stdio.h>

enum season { SPRING, SUMMER, FALL, WINTER };

int main(void)

{

enum season ss;

char\* pc = NULL;

ss = SPRING;

switch (ss)

{

case SPRING:

pc = "inline"; break;

case SUMMER:

pc = "swimming"; break;

case FALL:

pc = "trip"; break;

case WINTER:

pc = "skiing"; break;

}

printf("나의 레저 활동 = > %s\n", pc);

return 0;

}

// 17-12

#include <stdio.h>

struct student

{

int num;

double grade;

};

typedef struct student Student;

void print\_data(Student\* ps);

int main(void)

{

Student s1 = { 315, 4.2 };

print\_data(&s1);

return 0;

}

void print\_data(Student\* ps)-

{

printf("학번 : %d\n", ps->num);

printf("학점 : %.1lf\n", ps->grade);

}

// 181

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main(void)

{

FILE\* fp;

fp = fopen("a.txt", "r");

if (fp == NULL)

{

printf("파일이 열리지 않았습니다.\n");

return 1;

}

printf("파일이 열렸습니다.\n");

fclose(fp);

return 0;

}

// 18-2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main(void)

{

FILE\* fp;

int ch;

fp = fopen("a.txt", "r");

if (fp == NULL)

{

printf("파일이 열리지 않았습니다.\n");

return 1;

}

while (1)

{

ch = fgetc(fp);

if (ch == EOF)

{

break;

}

putchar(ch);

}

fclose(fp);

return 0;

}

// 18-3

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main(void)

{

FILE\* fp;

char str[] = "banana";

int i;

fp = fopen("b.txt", "w");

if (fp == NULL)

{

printf("파일을 만들지 못했습니다.\n");

return 1;

}

i = 0;

while (str[i] != '\0')

{

fputc(str[i], fp);

i++;

}

fputc('\n', fp);

fclose(fp);

return 0;

}

// 18-4

#include <stdio.h>

int main(void)

{

int ch;

while (1)

{

ch = getchar();

if (ch == EOF)

{

break;

}

putchar(ch);

}

return 0;

}

// 18-5

#include <stdio.h>

int main(void)

{

int ch;

while (1)

{

ch = fgetc(stdin);

if (ch == EOF)

{

break;

}

fputc(ch, stdout);

}

return 0;

}

// 18-6

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main(void)

{

FILE\* fp;

int ary[10] = { 13, 10, 13, 13, 10, 26, 13, 10, 13, 10 };

int i, res;

fp = fopen("a.txt", "wb");

for (i = 0; i < 10; i++)

{

fputc(ary[i], fp);

}

fclose(fp);

fp = fopen("a.txt", "rt");

while (1)

{

res = fgetc(fp);

if (res == EOF) break;

printf("%4d", res);

}

fclose(fp);

return 0;

}

// 18-7

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

int main(void)

{

FILE\* fp;

char str[20];

fp = fopen("a.txt", "a+");

if (fp == NULL)

{

printf("파일을 만들지 못했습니다.\n");

return 1;

}

while (1)

{

printf("과일 이름 : ");

scanf("%s", str);

if (strcmp(str, "end") == 0)

{

break;

}

else if (strcmp(str, "list") == 0)

{

fseek(fp, 0, SEEK\_SET);

while (1)

{

fgets(str, sizeof(str), fp);

if (feof(fp))

{

break;

}

printf("%s", str);

}

}

else

{

fprintf(fp, "%s\n", str);

}

}

fclose(fp);

return 0;

}

// 18-8

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

int main(void)

{

FILE\* ifp, \* ofp;

char str[80];

char\* res;

ifp = fopen("a.txt", "r");

if (ifp == NULL)

{

printf("입력 파일을 열지 못했습니다.\n");

return 1;

}

ofp = fopen("b.txt", "w");

if (ofp == NULL)

{

printf("출력 파일을 열지 못했습니다.\n");

return 1;

}

while (1)

{

res = fgets(str, sizeof(str), ifp);

if (res == NULL)

{

break;

}

str[strlen(str) - 1] = '\0';

fputs(str, ofp);

fputs(" ", ofp);

}

fclose(ifp);

fclose(ofp);

return 0;

}

// 18-9

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main(void)

{

FILE\* ifp, \* ofp;

char name[20];

int kor, eng, math;

int total;

double avg;

int res;

ifp = fopen("a.txt", "r");

if (ifp == NULL)

{

printf("입력 파일을 열지 못했습니다.\n");

return 1;

}

ofp = fopen("b.txt", "w");

if (ofp == NULL)

{

printf("출력 파일을 열지 못했습니다.\n");

return 1;

}

while (1)

{

res = fscanf(ifp, "%s%d%d%d", name, &kor, &eng, &math);

if (res == EOF)

{

break;

}

total = kor + eng + math;

avg = total / 3.0;

fprintf(ofp, "%s%5d%7.1lf\n", name, total, avg);

}

fclose(ifp);

fclose(ofp);

return 0;

}

// 18-10

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main(void)

{

FILE\* fp;

int age;

char name[20];

fp = fopen("a.txt", "r");

fscanf(fp, "%d", &age);

fgets(name, sizeof(name), fp);

printf("나이 : %d, 이름 : %s", age, name);

fclose(fp);

return 0;

}

// 18-11

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main(void)

{

FILE\* afp, \* bfp;

int num = 10;

int res;

afp = fopen("a.txt", "wt");

fprintf(afp, "%d", num);

bfp = fopen("b.txt", "wb");

fwrite(&num, sizeof(num), 1, bfp);

fclose(afp);

fclose(bfp);

bfp = fopen("b.txt", "rb");

fread(&res, sizeof(res), 1, bfp);

printf("%d", res);

fclose(bfp);

return 0;

}

// 19-1

// 소스 파일 – main.c

#include <stdio.h>

#include "student.h"

int main(void)

{

Student a = { 315, "홍길동" };

printf("학번 : %d, 이름 : %s\n", a.num, a.name);

return 0;

}

// 19-2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#define PI 3.14159

#define LIMIT 100.0

#define MSG "passed!"

#define ERR\_PRN printf("범위를 벗어났습니다!\n")

int main(void)

{

double radius, area;

printf("반지름을 입력하세요(100 이하) : ");

scanf("%lf", &radius);

area = PI \* radius \* radius;

if (radius > LIMIT) ERR\_PRN;

else printf("원의 면적 : %.2lf(%s)\n", area, MSG);

return 0;

}

// 19-3

#include <stdio.h>

#define SUM(a, b) ((a) + (b))

#define MUL(a, b) ((a) \* (b))

int main(void)

{

int a = 10, b = 20;

int x = 30, y = 40;

int res;

printf("a + b = %d\n", SUM(a, b));

printf("x + y = %d\n", SUM(x, y));

res = 30 / MUL(2, 5);

printf("res : %d\n", res);

return 0;

}

// 19-4

#include <stdio.h>

void func(void);

int main(void)

{

printf("컴파일 날짜와 시간 : %s, %s\n\n", \_\_DATE\_\_, \_\_TIME\_\_);

printf("파일명 : %s\n", \_\_FILE\_\_);

printf("함수명 : %s\n", \_\_FUNCTION\_\_);

printf("행번호 : %d\n", \_\_LINE\_\_);

#line 100 "macro.c"

func();

return 0;

}

void func(void)

{

printf("\n");

printf("파일명 : %s\n", \_\_FILE\_\_);

printf("함수명 : %s\n", \_\_FUNCTION\_\_);

printf("행번호 : %d\n", \_\_LINE\_\_);

}

// 19-5

#include <stdio.h>

#define PRINT\_EXPR(x) printf(#x " = %d\n", x)

#define NAME\_CAT(x, y) (x ## y)

int main(void)

{

int a1, a2;

NAME\_CAT(a, 1) = 10; // (a1) = 10;

NAME\_CAT(a, 2) = 20; // (a2) = 20;

PRINT\_EXPR(a1 + a2); // printf("a1 + a2" " = %d\n", a1 + a2);

PRINT\_EXPR(a2 - a1); // printf("a2 - a1" " = %d\n", a2 - a1);

return 0;

}

// 19-6

#include <stdio.h>

#define VER 7

#define BIT16

int main(void)

{

int max;

#if VER >= 6

printf("버전 %d입니다.\n", VER);

#endif

#ifdef BIT16

max = 32767;

#else

max = 2147483647;

#endif

printf("int형 변수의 최댓값 : %d\n", max);

return 0;

}

// 19-7

#include <stdio.h>

#pragma pack(push, 1)

typedef struct

{

char ch;

int in;

} Sample1;

#pragma pack(pop)

typedef struct

{

char ch;

int in;

} Sample2;

int main(void)

{

printf("Sample1 구조체의 크기 : %d바이트\n", sizeof(Sample1));

printf("Sample2 구조체의 크기 : %d바이트\n", sizeof(Sample2));

return 0;

}v

// 19-8

#include <stdio.h>

void input\_data(int\*, int\*);

double average(int, int);

int main(void)

{

int a, b;

double avg;

input\_data(&a, &b);

avg = average(a, b);

printf("%d와 %d의 평균 : %.1lf\n", a, b, avg);

return 0;

}

// 19-9

#include <stdio.h>

int input\_data(void);

double average(void);

void print\_data(double);

int count = 0; static int total = 0;

int main(void)

{

double avg;

total = input\_data();

avg = average();

print\_data(avg);

return 0;====

}

void print\_data(double avg)

{

printf("입력한 양수의 개수 : %d\n", count);

printf("전체 합과 평균 : %d, %.1lf\n", total, avg);

}

// 19-10

#include <stdio.h>

#include "point.h"

#include "line.h"

int main(void)

{

Line a = { {1, 2}, {5, 6} };

Point b;

b.x = (a.first.x + a.second.x) / 2;

b.y = (a.first.y + a.second.y) / 2;

printf("선의 가운데 점의 좌표 : (%d, %d)\n", b.x, b.y);

return 0;

}