

가계부.txt

```
#include <stdio.h>          // 표준 입출력 헤더파일
#include <stdlib.h>          // exit(), rand(), system() 함수등이 담긴 헤더파일. 이번 프
로그래머에서는 system("cls") 를 사용하기 위하여 사용된 헤더파일이다.
#include <string.h>          // 메모리 초기화 함수, memset을 사용하기 위해
사용한 헤더파일. memset에 대한 설명은 case2 부분에 사용한 부분에 주석을 달아놓음.

#define MAXLINE 100          // MAXLINE 을 100 으로 치환하는 전처리지시문

void f_tittle();              // 타이틀
출력을 불러오는 함수 선언

int print_menu();             // 메뉴출
력을 불러오는 함수 선언

int menufunction(int f_menunum); // 메뉴에 해당하는 switch
case를 실행시키기 위한 함수 선언

////////////////////////////////////
////////////////////////////////////

char prodName [MAXLINE][30] = {0}; // 입력할 상품의 "이름" 이 저장될 2차
원 배열

float prodValue [MAXLINE] = {0};    // 입력할 상품의 "가격" 이 저장될 1차
원 배열

int prodCount [MAXLINE] = {0};      // 입력할 상품의 "갯수" 가
저장될 1차원 배열

int main()          // main() 의 시작
{

    int f_menunum, menunum;
    float sum = 0;

    while(1)          // 전체메뉴 main() 을 반복하기 위한 무한반복문
    {

        f_tittle(); // 타이틀출력 함수 호출
```

가계부.txt

f_menunum = print_menu(); // 메뉴출력을 불러오는 함수 호출.
f_menunum 에 정수값 메뉴번호 전달

system("cls"); // 스크린
을 한번 초기화시킴

menufunction(f_menunum); // 메뉴에 해당하는 switch
case 함수 호출

system("cls"); // 스크린
을 한번 초기화시킴

if (f_menunum == 3){
printf("프로그램을 종료합니다.\n"); // 정수 3을 받아서
case3 으로 넘어갈 때, case 탈출 이후의 전체메뉴 무한반복 while(1) 을 탈출하기위한 조
건문
break;
}

}

} // main() 의 끝

////////////////////////////////////
////////////////////////////////////

void f_tittle() // 타이틀출력을 불러오는 함수 정의
{

printf("=====
=====Wn");
printf("Wt제작자 : 0960030 송두영 & 1260014 김철언WnWnWn");
printf("Wt프로그램 설명 : 메뉴와 가격, 수량을 입력받아 계산하여 출력하는
WnWtWtWt메모, 가계부 역할을 위한 프로그램.WnWn");

printf("=====
=====Wn");
}

////////////////////////////////////

가계부.txt

////////////////////////////////////

int print_menu() // 메뉴출력을 불러오는 함수 정의

{

int f_menunum; // f_menunum 에 정수를 받아서 이후에 case를
실행할 것임

while(1) // 메뉴출력을 반복하기위한 무한반복문
{

printf("1. 품목 입력 WnWn");
printf("2. 영수증 출력WnWn");
printf("3. 프로그램 종료WnWn");
printf("메뉴 번호를 입력 하세요(1 - 3) : ");

scanf("%d", &f_menunum);
getchar(); // 문자를 받아도 무한반복하지 않게 걸어놓음

if(f_menunum > 0 && f_menunum < 4) break; // 1~3만을 입력하게끔 조건
을 걸었음.

printf("Wn!Error! 메뉴 번호 1 - 3 중 선택하여 입력해주세요.
!Error!WnWn");

}

return f_menunum;

}

////////////////////////////////////
////////////////////////////////////

int menufunction(int f_menunum) // 메뉴에 해당하는 switch case를 실행
시키기 위한 함수 정의

{

int i, j;
float sum;

switch(f_menunum)

가계부.txt

```
{
    case 1 :

        // 입력 받기 전에 초기화 기본형태는, memset( 위치 , 초기
        화값, 사이즈 )
        // 즉, prodName을 sizeof(prodName) 크기만큼 16진수
        0값으로 0x00 (0) 으로 초기화 하겠다.

        memset(prodName , 0x00, sizeof(prodName));
        memset(prodValue, 0x00, sizeof(prodValue));
        memset(prodCount, 0x00, sizeof(prodCount));

        for (i = 0; i < MAXLINE; i++) // 주소값 0부터 100까지 반
        복시킬것임
        {

            printf("상품입력란에 x 를 입력시 품목입력을 종
            료합니다\n상품의 이름 : ");

            scanf("%s", &prodName[i][0]);          //
            prodName[0][0] 부터 쪽 저장시킬것임.

            if( prodName[i][0] == 'x' || prodName[i][0] ==
            'X') break;// 이름입력에 x 혹은 대문자 X를 받을 경우 저장을 중단함.

            printf("상품의 가격 : ");

            scanf("%f", &prodValue[i]); // 상품 가격배열에
            [0] 부터 저장

            printf("상품의 갯수 : ");

            scanf("%d", &prodCount[i]); // 상품 갯수에 [0]
            부터 저장

            printf("\n");
        }
    }
}
```

가계부.txt

```
}  
break;
```

case 2 :

```
system("cls"); // 화면 클리어
```

```
printf("=====  
=====");
```

```
printf("%20s%9s%5s\n","상품명","가격","수량\n");
```

// 표처럼 출력을 위해 맨 위에 상품명, 가격, 수량을 출력시킴

```
for ( j = 0; j < MAXLINE; j++) // 0번부터 MAXLINE(=100)
```

까지 반복함

```
{
```

```
    if( prodName[j][0] == 'x' || prodName[j][0] ==  
'X') break; // x로 메뉴입력을 중단한 부분에서 break.
```

```
    if ( prodName[j][0] == 0x00 ){
```

```
        printf("데이터가 없습니다.\n");
```

```
        break;
```

```
    } // 이것은 만약을 위해 만든 조건문인
```

데, 아무것도 입력 안하고 2번을 눌러서 영수증을 출력하게끔 웨이크를 거는 사람들을 위
해 데이터가 없다고 출력하기 위한것임.

```
printf("%20s\n", prodName[j]); // prodName 출
```

력

```
printf("%9.2f\n", prodValue [j]); // prodValude 출
```

력

```
printf("%5d\n", prodCount[j]); // prodCound 출
```

력

```
}
```

```
printf("=====  
=====");
```

가계부.txt

=====\\n");

sum = 0; // sum+= prodValue[j] * prodCount[j] 는 sum
에 *을 연속 적으로 하는 것이므로 초기에 sum = 0 으로 초기화 해야 함 <- 추가

// 최종 결과 총합을 계산하기 위한 수식으로, 상품 가격과
갯수의 연산을 sum 에 저장하면서 Name의 데이터가 없는, 즉 0값(16진수로 0x00)이 있는
곳까지 반복을 달림.

// Name이 0이 있는곳을 찾으면, 상품가격 * 갯수 연산을
당연히 마칠것임.

// 마친부분까지 sum 저장을 최종적으로 출력함

for(j = 0; j < MAXLINE; j++){

sum += prodValue[j] * prodCount[j];

if (prodName[j][0] == 0x00) //

prodNmae을 0일때, break로 멈추는 조건문. 데이터가 아예 없는 경우를 위함.

break;

}

printf("총 합 : %9.2f 원 입니다.", sum); // 연산한

총합 출력

getchar(); // 다시 메뉴로 돌아가는 무한반복을 방

지하기 위해 엔터를 한번 더 누르게 해서 메뉴로 돌아감.

break;

case 3 :

printf("프로그램을 종료합니다.\\n");

break;

}

return f_menunum;

}

가계부.txt

////////////////////////////////////
////////////////////////////////////