학번: 2019202052

이름:김호성

249쪽 도전 4번 문제

초반 코드

6과제1. 교재 249p. 도전4

소스 코드

#include <stdio.h>

int main(void)

{

int chong = 5000; // 처음에 5000원

int dvd = 1500; // dvd값 1500원

int cream = 500; // 크림빵 개당 500원

int shrimp = 700; // 새우 개당 700원

int coke = 400; // 콜라 개당 400원

int hyunjae1 = chong - dvd; // 현재 5000-1500 = 3500원

int x = 1; // 500x + 700y + 400z = 3500

int y = 1;

int z = 1;

printf("현재 당신이 소유하고 있는 금액 : %d \n", hyunjae1);

for (x = 1; hyunjae1/cream > x; x++) // 3500 / 크림빵비용이 x보다 클때 x하나씩 증가하고 반복

{

for (y = 1; hyunjae1 / shrimp > y; y++) // 3500 / 새우 비용이 y보다 클때 y하나씩 증가하고 반복

{

for (z = 1; hyunjae1 / coke > z; z++) // 3500 / 콜라 비용이 z보다 클때 z하나씩 증가하고 반복

{

if (hyunjae1 == cream \* x + shrimp \* y + coke \* z) // 500x + 700y + 400z = 3500

printf("크림빵 %d개, 새우깡 %d개, 콜라 %d개 \n", x, y, z);

}

}

}

}

소스코드 변형 후

#include<stdio.h>

const int BREAD=500; //빵의 가격 고정

const int SNACK=700; //과자의 가격 고정

const int DRINK=400; //음료수의 가격 고정

int main(void)

{

int bread, snack, drink; //빵, 과자, 음료수의 개수를 측정하기 위해 변수 생성

int money; //현재 보유한 금액을 입력받기 위해 변수 생성

printf("현재 당신이 소유하고 있는 금액 : ");

scanf("%d", &money); //현재 자본 입력받음

for(bread=1; bread<money/BREAD; bread++) //빵의 개수 < 보유금액/빵 값

{

for(snack=1; snack<money/SNACK; snack++) //과자의 개수 < 보유금액/과자의 값

{

for(drink=1; drink<money/DRINK; drink++)//음료수의 개수 < 보유금액/음료수의 값

{

if(money==BREAD\*bread+SNACK\*snack+DRINK\*drink)

{

printf("크림빵 %d개, ", bread);

printf("새우깡 %d개, ", snack);

printf("콜 라 %d개 \n", drink);

}

}

}

}

printf("어떻게 구입하시겠습니까? \n");

return 0;

}

/\*

소소코드 진행(3500원일때 가정)

빵:1 과자:1 음료수:1

빵:1 과자:1 음료수:2

●●●

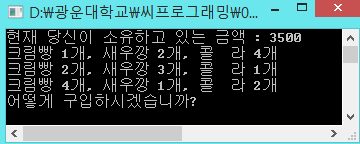
빵:1 과자:1 음료수7 (3500/8은 8.xx지만 int형이므로 소수점이 사라지므로 7까지만 증가)

빵:1 과자2: 음료수1 (for문 하나 나와서 과자로 넘어갔으므로 음료수값 1로 초기화)

●●●

빵:1 과자2: 음료수4(if조건에 맞으므로 출력)

\*/



초반에는 전역변수는 무조건 쓰면 코드가 꼬인다고 생각했다. 그러나 실상은 2~4개 정도 사용하는데 런 타임에 그리 차이 있다는 것을 느끼지 못해서 전역변수를 사용했다.(물론 전역변수는 많이 쓸수록 좋지 않은 것 또한 안다.) 현재 같은 경우에는 전역변수가 많아질수록 스파게티코드가 되는 걸 알고 있지만, 일단 3개 밖에 되지 않기 때문에 그냥 사용했지만, 정말 프로그램을 만들고 싶을 때는 전역변수를 줄이는 게 중요하다는 것을 느꼈다.(전역변수가 적을수록 런 타임이 낮은 건 사실이긴 때문에 이렇게 생각했다.)