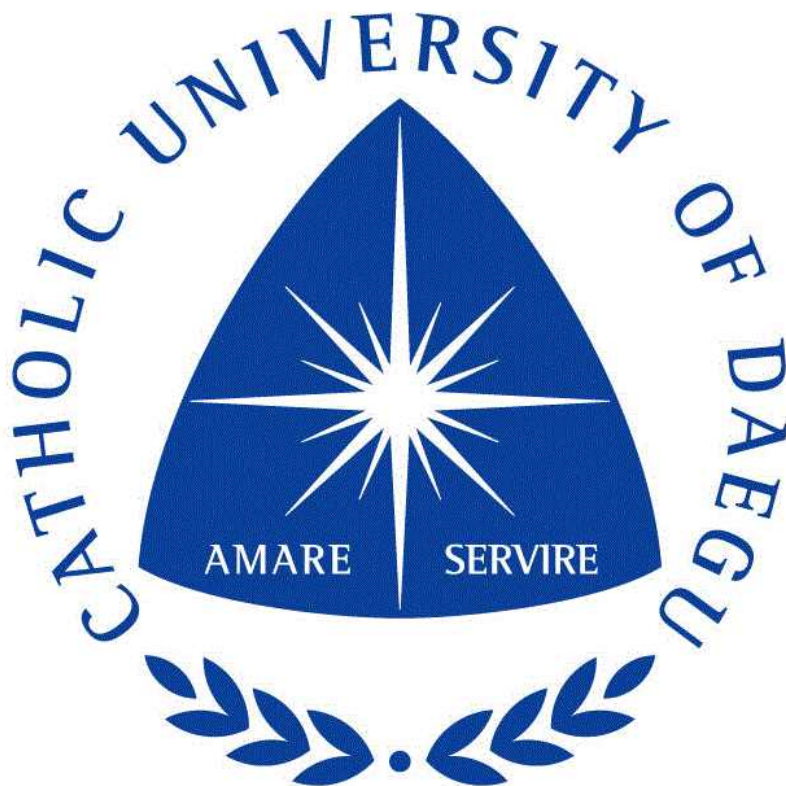




C를 이용한 자판기 프로그램의 구현

C 프로그래밍(06)



| | |
|-------|--------------|
| 학과 | 컴퓨터소프트웨어학부 |
| 학번 | 20116742 |
| 이름 | 권양하 |
| 담당 교수 | 김미혜 교수님 |
| 제출일 | 2020. 12. 11 |

목 차

| | |
|-----------------------------|----|
| 1. 서론 | 3 |
| 2. 관련 연구 | 3 |
| 3. 프로그램 설계 | 4 |
| 3.1. 프로그램 수행 계획 | |
| 3.2. 프로그램 기획 | |
| 3.3. 프로그램 구성 | |
| 4. 프로그램 구현 | 5 |
| 4.1. 헤더파일 및 매크로 상수 | |
| 4.2. 주요 변수 | |
| 4.3. 소스코드 설명 | |
| 5. 프로그램의 동작과 결과 | 8 |
| 6. 결과 분석 및 토의 | 9 |
| 6.1. 금액 투입 | |
| 6.2. 음료 선택 및 거스름돈 반환 | |
| 6.3. 음료 재고 | |
| 6.4 프로그램 종료 | |
| 7. 결론 | 11 |
| 7.1. 보고서의 결론 | |
| 7.2. 프로젝트를 하면서 느낀 점 | |
| 7.3. 향후 자신의 프로그래밍에 대한 짧은 소견 | |
| 8. 참고문헌 | 12 |
| 9. 별 첨(소스 코드와 주석) | 12 |

1. 서론

자판기 프로그램을 C 언어로 구현하여 자판기에서 음료가 나오기 위한 실행 결과와 과정을 다루고 있다.

1. 관련 연구
2. 프로그램 설계
3. 프로그램 구현
4. 프로그램 동작과 결과
5. 결과 분석 및 토의
6. 결론
7. 참고 문헌
8. 별첨(소스코드와 주석)

2. 관련 연구

자판기 프로그램에서 원하는 음료를 선택하기 위해 정수형 int를 이용해 음료마다 숫자를 매겨 숫자를 입력할 경우, 숫자에 해당되는 음료가 출력될 수 있도록 설정한다. 각 음료마다 해당되는 가격이나 재고가 다를 수 있으므로 반복문 while문과 조건문 if문, switch문을 이용해 여러 가정을 통해 옳은 가정일 경우, break문을 이용해 출력이 중지되고 옳지 않은 가정일 경우, continue문을 통해 다시 입력받을 수 있도록 설정한다. 사용자가 자판기를 본인 의지에 따라 종료하기 위해 반복문 if 문을 이용해 해당 결과를 출력하도록 하였다.

3. 프로그램 설계

3.1 프로그램 수행 계획

자판기 프로그램을 수행하기 위해 해당 소스코드에 필요한 함수나 변수 등을 공부하며 C프로그래밍 프로그램 Microsoft Visual Studio Community 2019와 GDB online Debugger를 이용해 소스코드를 입력하고 실행해 볼 것이다.

3.2 프로그램 기획

자판기 프로그램에서 코드의 가장 처음에 오는 헤더파일 #include <stdio.h>를 이용해 기본적인 입력, 출력을 도와주는 함수를 불러오고 매크로 함수 #define을 통해 음료와 가격을 정의한다. 정수형 int를 이용해 메뉴선택, 금액, 음료개수와

금액의 총합, 거스름돈, 음료재고를 출력한다. 반복문 while문과 조건문 if문, switch문을 이용해 투입 금액을 설정하고, 음료종류에 따라 다른 조건에 알맞은 선택을 하여 원하는 음료를 뽑아 먹을 수 있도록 기획하였다.

3.2 프로그램 구성

자판기 프로그램의 구성은 간단히 메뉴선택, 금액, 음료개수, 투입 금액의 합, 거스름 돈, 음료재고 등으로 구성되며 마지막으로 자판기 프로그램 운영과 종료를 사용자가 직접 선택할 수 있도록 설정해 종료하다면 자판기 프로그램이 종료되도록 구성하였다.

4. 프로그램 구현

4.1 헤더파일 및 매크로 상수

- 1) #include <stdio.h>
 - 코드의 가장 처음에 오는 전처리부로 표준 입출력을 담당하는 헤더파일로 입력과 출력을 도와주는 함수
- 2) #include <windows.h>
 - system("cls");를 사용하기 위한 헤더파일
- 2) #define/음료/가격
 - 매크로 함수 뒤에 원하는 음료와 해당 가격을 정의한 함수

4.2 주요 변수

- 1) int MenuChoice;, int Money;,
 - int로 메뉴선택, 금액, 음료개수, 투입 금액의 합, 거스름 돈 등의 변수를 설정
- 2) int 음료 = 음료 재고 개수;
 - 정수형 int를 이용해 각 음료의 재고를 원하는 개수만큼 설정해 소스코드를 실행할 경우, 남은 재고를 출력

4.3 소스코드 설명

- 1) 밑의 int형 변수는 각 음료의 종류와 재고를 나타내 소스코드 실행 시, 음료의 재고량을 출력할 수 있다.

```
int cider = 10;  
int cancoffee = 10;  
int fanta = 10;  
int blackcoffee = 10;  
int milkcoffee = 10;
```

2) 각 음료마다 해당되는 조건이 다르므로 조건문 switch문을 이용해 음료를 case로 나누어 모든 음료의 조건을 다르게 설정하였다. case1부터 case5까지 존재한다.

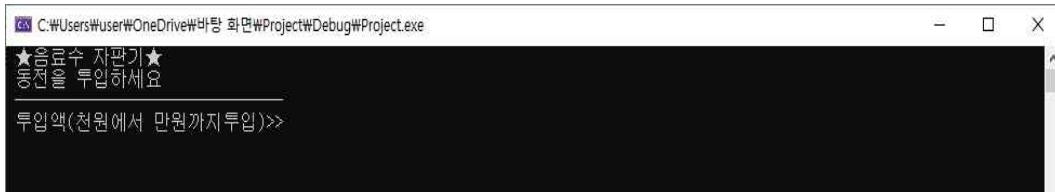
```
switch (MenuChoice) {  
case 1:  
    if (cider == 0) {  
        printf("사이다 재고가 부족합니다.\n");  
        continue;  
    }  
    else if ((Cider * Number) > Money) {  
        printf("돈이 부족합니다.\n");  
        continue;  
    }  
    else {  
        Hap = Cider * Number;  
        Nameozi = Money - Hap;  
        Money = Nameozi;  
        cider -= Number;  
        printf("돈 %d원을 쓰셨습니다.\n\n", Hap);  
        printf("거스름돈 %d원 입니다.\n\n", Nameozi);  
        printf("사이다를 맛있게 드십시오.\n\n");  
        printf("사이다 재고가 %d개 남았습니다.\n\n", cider);  
    }  
    break;
```

3) 사용자가 원하는 음료를 뽑아 먹고 난 후, 자판기 프로그램을 계속 운영할지, 종료할지 사용자가 직접 선택할 수 있도록 조건문 if문을 이용해 소스코드를 구현하였다.

```
Jong:  
int yn;  
fflush(stdin); //입력 버퍼를 지우는 함수  
printf("자판기를 종료하시겠습니까?\n");  
printf("(종료 하시려면 1, 종료 하지 않으시려면 2를 입력하세요)>> ");  
scanf_s("%d", &yn); //사용자로부터 y/n을 입력받는다.  
printf("\n");  
  
if (yn == 1) {  
    printf("자판기 종료합니다.\n");  
    break;  
}  
else if (yn == 2) {  
    printf("자판기 계속 운영합니다.\n");  
    continue;  
}  
else {  
    printf("잘못 입력하셨습니다.\n");  
    goto Jong;  
}  
}  
return 0;
```

5. 프로그램 동작과 결과

1) 음료수 자판기에 동전을 투입



2) system("cls");를 입력해 처음의 음료수 자판기 입력 문자가 사라지고 5000원이 투입되었다는 입력 문자가 나타나며 음료수 자판기가 출력되고 1번부터 5번 중 원하는 음료선택



3) 음료3번 환타를 선택하면 품목개수 선택 (단, 투입한 금액에 비해 가격이 높은 음료를 선택할 경우, 다시 선택의 기회제공)



4) 품목개수 4개를 입력하면 음료금액과 거스름돈 음료의 재고가 출력 (단, 입력한 금액에 비해 더 많은 음료의 개수를 선택할 경우, 다시 선택의 기회제공)

```
C:\Users\User\OneDrive\바탕 화면\Project1\Debug\Project1.exe
5000원 투입되었습니다.

★음료수 자판기★

1. 사이다(700원)
2. 캔커피(700원)
3. 쿼타(800원)
4. 블랙커피(300원)
5. 밀크커피(300원)

품목선택>> 2
품목개수>> 3

돈 2100원을 쓰셨습니다.
거스름돈 2900원 입니다.
캔커피를 맛있게 드십시오.
캔커피 재고가 7개 남았습니다.
```

5) 사용자가 원하는 음료를 뽑고 나면 자판기를 종료할 것인지 묻는 문장이 출력

```
C:\Users\User\OneDrive\바탕 화면\Project1\Debug\Project1.exe
5000원 투입되었습니다.

★음료수 자판기★

1. 사이다(700원)
2. 캔커피(700원)
3. 쿼타(800원)
4. 블랙커피(300원)
5. 밀크커피(300원)

품목선택>> 2
품목개수>> 3

돈 2100원을 쓰셨습니다.
거스름돈 2900원 입니다.
캔커피를 맛있게 드십시오.
캔커피 재고가 7개 남았습니다.

자판기를 종료하시겠습니까?
(종료 하시려면 1 , 종료 하지 않으시려면 2를 입력하세요)>>
```

6) 자판기를 종료하기 위해 1을 입력하면 자판기 프로그램 종료

```
Microsoft Visual Studio 디버그 콘솔

★음료수 자판기★

1. 사이다(700원)
2. 캔커피(700원)
3. 쿼타(800원)
4. 블랙커피(300원)
5. 밀크커피(300원)

품목선택>> 2
품목개수>> 3

돈 2100원을 쓰셨습니다.
거스름돈 2900원 입니다.
캔커피를 맛있게 드십시오.
캔커피 재고가 7개 남았습니다.

자판기를 종료하시겠습니까?
(종료 하시려면 1 , 종료 하지 않으시려면 2를 입력하세요)>> 1

자판기 종료합니다.

C:\Users\User\OneDrive\바탕 화면\Project1\Debug\Project1.exe(프로세스 17848개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

6. 결과 분석 및 토의

6.1 금액 투입

MenuChoice 함수를 통해 원하는 음료를 선택하기 전, Money 함수를 통해 사용자가 자판기에 투입하고 싶은 금액의 돈을 입력한다. 천원에서 만원 사이 금액이라면 다음 소스코드가 실행되지만 그렇지 않을 경우 “잘못 투입되었습니다.”라는 출력 문장과 함께 다시 투입액을 입력해야 한다. 이후, `system("cls");`을 입력하여 금액 투입 문장 콘솔 창에서 사라지게 한다.

6.2 음료선택 및 거스름돈 반환

MenuChoice 함수를 통해 원하는 음료를 선택하기 위해 각 음료에 해당되는 번호를 입력하면 사용자가 원하는 음료의 종류가 출력된다. 음료의 개수를 선택하기 위한 Number 함수로 설정해둔 재고 내에서 원하는 개수를 선택해 입력할 수 있다. 음료의 가격과 개수를 곱해 지불해야 하는 금액을 계산하여 총 금액을 출력한다. 그 후, 처음 투입했던 금액보다 음료 금액이 적다면 거스름돈을 계산해 출력하며, 부족하다면 “잘못 개수를 입력하셨습니다.”라고 출력된다. 각 음료마다 종류, 금액, 재고가 다르기 때문에 조건문 switch를 이용해 5가지 종류의 음료를 case1부터 case5까지 구분해 그에 알맞은 조건이 입력된다.

6.3 음료 재고

int형 변수를 통해 각 음료의 재고를 직접 설정할 수 있다. 예를 들어, 사이다를 3개를 자판기에서 뽑는다면 뽑고 난 후의 재고를 표시해야 한다. 그러기 위해 `cider -= Number;` 소스코드를 입력해 입력한 음료 개수를 빼주게 되면 음료 재고를 출력할 수 있다. `printf()`를 이용해 몇 개가 남았는지 출력한다. 또한 처음부터 음료의 재고가 0개라면 조건문 if를 통해 나타내고 `printf()`를 이용해 재고가 부족하다고 출력한다.

6.4 프로그램 종료

자판기에서 원하는 음료를 뽑았다면 자판기 운영을 종료할 것인지 운영할 것인지 사용자가 직접 선택할 수 있는 질문이 입력된다. 사용자가 ‘1’을 입력할 경우, 자판기 운영은 종료되며 ‘2’를 선택할 경우, 자판기 운영은 계속된다. 단, 다른 단어를 입력하면 “잘못 입력하셨습니다.”라는 문장이 등장하며 다시 질문한다.

7. 결론

7.1 보고서의 결론

C언어로 구현한 자판기 프로그램을 통해 음료선택, 금액투입, 음료개수, 거스름돈, 재고를 입력, 출력할 수 있는 소스코드를 구현하여 실행할 수 있다.

7.2 프로젝트를 하면서 느낀 점

처음에는 단순히 학교 수업시간에 하는 실습이나 과제와 같다고 생각해 쉽게 할 수 있을 것이라 생각해 여유를 부렸다. 그런데 여러 문제들 중 무엇이 가장 재미있으면서 과제 같지 않도록 정말 내가 하고 싶어서 할 수 있는 걸까? 정말 많이 고민하게 되어 문제를 고르고 모두 실행시켜 보는 시간만 3일이 걸렸다. 물론 모든 문제를 풀어보아 뿌듯하긴 했지만 다른 친구들에 비해 많은 시간이 들었기 때문에 그만큼 다른 부분에 소홀했다고 느낀다. 이러한 시행착오를 겪어 선택하게 된 문제가 바로 자판기 프로그램이다. 처음에 타자 연습 프로그램을 선택했지만 생각보다 깊이 있는 문제는 아니라 생각되어 자판기 프로그램으로 변경하게 되었다. 그래서 1년 동안 내가한 코딩 프로젝트 중에 가장 재미있으면서 한편으로는 힘든 프로젝트가 되었다. 그리고 가장 힘들었던 소스코드는 음료의 재고를 출력하는 부분이었다. 재고가 0개일 때와 음료를 뽑고 난 후의 재고를 출력해야 해서 시작을 어떻게 해야 할지 고민이었다. 나는 이 코드를 구현하며 어렵게 생각하면 더 어렵다는 말이 생각났다. 그래서 간단하게 생각하며 코드를 작성했더니 생각보다 쉽게 해결되어 내가 원하는 소스코드를 얻을 수 있었다. 이렇게 안 풀리던 부분을 해결한 뒤에 오는 희열감 때문에 코딩을 계속하게 되는 것 같다. 앞으로도 지금보다 성장하기 위해 많은 소스코드를 작성해보며 성장하는 사람이 되고 싶다.

7.3 향후 자신의 프로그래밍에 대한 짧은 소견

아직 나는 처음부터 끝까지 스스로 소스코드를 작성하기에는 힘든 실력을 가지고 있다. 그렇기에 2020년 남은 시간동안 C언어에 대해 더 공부해 언젠가 누군가 작성해둔 소스코드를 참고하지 않고도 원하는 프로그램을 만들어 볼 수 있었으면 한다. 그래서 지금부터 매일 꾸준히 공부해 남에게 알려줄 수 있는 실력으로 향상되길 희망한다.

8. 참고문헌

- [1] 강환수 외 2명, “Perfect C”, infinity books.
- [2] 서동만, 전수빈, “C프로그래밍 기초”, infinity books.
- [3] C언어 goto문에 대해 참고

<https://namu.wiki/w/GOTO>

[4] 음료 자판기 프로그램 소스 코드 예시

<https://kldp.org/node/161064>

9. 별첨(소스 코드와 주석)

```
#include <stdio.h>
#include <windows.h> //system("cls")를 사용하기 위한 헤더파일
#pragma warning(disable : 4996)
#define TRUE 1
#define Cider 700
#define Cancoffee 700
#define Fanta 600
#define Blackcoffee 300
#define Milkcoffee 300

int MenuChoice: //메뉴선택
int Money: //투입할 금액
int Number: //음료개수
int Hap: //음료개수와 투입도니 돈의 합
int Nameozi: //거스름 돈
int cider = 10;
int cancoffee = 10;
int fanta = 10;
int blackcoffee = 10;
int milkcoffee = 10;

int MainMenu()
{
    while (TRUE) { //while문을 이용해 반복시킨다.
        printf(" ★음료수 자판기★ \n");
        printf(" 동전을 투입하세요. \n");
        printf(" _____\n");
        printf(" 투입액(천원에서 만원까지 투입)>> ");
        scanf_s("%d", &Money); //사용자로부터 투입할 금액을 입력받는다.

        if ((Money < 1000) || (Money > 10000)) { //조건문 if문을 이용해 조건에 해당하는 입력 값을 받는다.
            printf("잘못 투입되었습니다.\n");
            system("cls"); //화면 갱신 코드
            continue;
        }
        else {
            system("cls");
            printf("%d원 투입되었습니다.\n\n", Money);
            return 0;
        }
    }
}
```

```

int Menu()
{
    while (TRUE) { //while문을 이용해 반복시킨다.
        printf(" ★음료수 자판기★ \n");
        printf(" _____ \n");
        printf(" | 1. 사이다(700원)   | \n");
        printf(" | 2. 캔커피(700원)   | \n");
        printf(" | 3. 환 타(600원)    | \n");
        printf(" | 4. 블랙커피(300원) | \n");
        printf(" | 5. 밀크커피(300원) | \n");
        printf(" _____ \n\n");
        printf("품목선택>> ");
        scanf_s("%d", &MenuChoice);

        if ((MenuChoice < 0) || (MenuChoice > 5)) {
            //조건문 if문을 이용해 조건에 해당되는 입력 값을 받는다.
            printf("잘못 입력하셨습니다.\n");
            continue;
        }
        fflush(stdin); //입력 버퍼를 지우는 함수
        printf("품목개수>> ");
        scanf_s("%d", &Number);
        printf("\n");
        //조건문 if문을 이용해 조건에 해당되는 입력 값을 받는다.
        if (Number < 0) {
            printf("잘못 개수 입력하셨습니다.\n");
            continue;
        }

        switch (MenuChoice) {

        case 1:
            if (cider == 0) {
                printf("사이다 재고가 부족합니다.\n");
                continue;
            }
            else if ((Cider * Number) > Money) {
                printf("돈이 부족합니다.\n");
                continue;
            }
            else {
                Hap = Cider * Number;
                Nameozi = Money - Hap;
                Money = Nameozi;
                cider -= Number;
                printf("돈 %d원을 쓰셨습니다.\n\n", Hap);
                printf("거스름돈 %d원 입니다.\n\n", Nameozi);
                printf("사이다를 맛있게 드십시오.\n\n");
                printf("사이다 재고가 %d개 남았습니다.\n\n", cider);
            }
            break;

```

case 2:

```
if (cancoffee == 0) {
    printf("캔커피 재고가 부족합니다.\n");
    continue;
}
else if ((Cancoffee * Number) > Money) {
    printf("돈이 부족합니다.\n");
    continue;
}
else {
    Hap = Cancoffee * Number;
    Nameozi = Money - Hap;
    Money = Nameozi;
    cancoffee -= Number;
    printf("돈 %d원을 쓰셨습니다.\n\n", Hap);
    printf("거스름돈 %d원 입니다.\n\n", Nameozi);
    printf("캔커피를 맛있게 드십시오.\n\n");
    printf("캔커피 재고가 %d개 남았습니다.\n\n", cancoffee);
}
break;
```

case 3:

```
if (fanta == 0) {
    printf("환타 재고가 부족합니다.\n");
    continue;
}
if ((Fanta * Number) > Money) {
    printf("돈이 부족합니다.\n");
    continue;
}
else {
    Hap = Fanta * Number;
    Nameozi = Money - Hap;
    Money = Nameozi;
    fanta -= Number;
    printf("돈 %d원을 쓰셨습니다.\n\n", Hap);
    printf("거스름돈 %d원 입니다.\n\n", Nameozi);
    printf("환타를 맛있게 드십시오.\n\n");
    printf("환타 재고가 %d개 남았습니다.\n\n", fanta);
}
break;
```

case 4:

```
if (blackcoffee == 0) {
    printf("블랙커피 재고가 부족합니다.\n");
    continue;
}
if ((Blackcoffee * Number) > Money) {

    printf("돈이 부족합니다.\n");
```

```

        continue;

    }

    else {
        Hap = Blackcoffee * Number;
        Nameozi = Money - Hap;
        Money = Nameozi;
        blackcoffee -= Number;
        printf("돈 %d원을 쓰셨습니다.\n\n", Hap);
        printf("거스름돈 %d원 입니다.\n\n", Nameozi);
        printf("블랙커피를 맛있게 드십시오.\n\n");
        printf("블랙커피 재고가 %d개 남았습니다.\n\n", blackcoffee);
    }
    break;

case 5:
    if (milkcoffee == 0) {
        printf("밀크커피 재고가 부족합니다.\n");
        continue;
    }
    if ((Milkcoffee * Number) > Money) {
        printf("돈이 부족합니다.\n");
        continue;
    }
    else {
        Hap = Milkcoffee * Number;
        Nameozi = Money - Hap;
        Money = Nameozi;
        milkcoffee -= Number;
        printf("돈 %d원을 쓰셨습니다.\n\n", Hap);
        printf("거스름돈 %d원 입니다.\n\n", Nameozi);
        printf("밀크커피를 맛있게 드십시오.\n\n");
        printf("밀크커피 재고가 %d개 남았습니다.\n\n", milkcoffee);
    }
    break;
}

Jong:
int yn;
fflush(stdin); //입력 버퍼를 지우는 함수
printf("자판기를 종료하시겠습니까?\n");
printf("(종료 하시려면 1 , 종료 하지 않으시려면 2를 입력하세요)>> ");
scanf_s("%d", &yn); //사용자로부터 y/n을 입력받는다.
printf("\n");

if (yn == 1) {
    printf("자판기 종료합니다.\n");
    break;
}
else if (yn == 2) {

```

```

        printf("자판기 계속 운영합니다.\n");
        continue;
    }
    else {
        printf("잘못 입력하셨습니다.\n");
        goto Jong;
    }
}
return 0;
}

int main()
{
    MainMenu();
    Menu();
    return 0;
}

```