

R E P O R T

학생식당 자판기 프로그램 제작



학 과	창의소프트웨어공학부
교수님	박유현
학 번	20193148
분 반	002
이 름	황진주
제출일	2019.10.

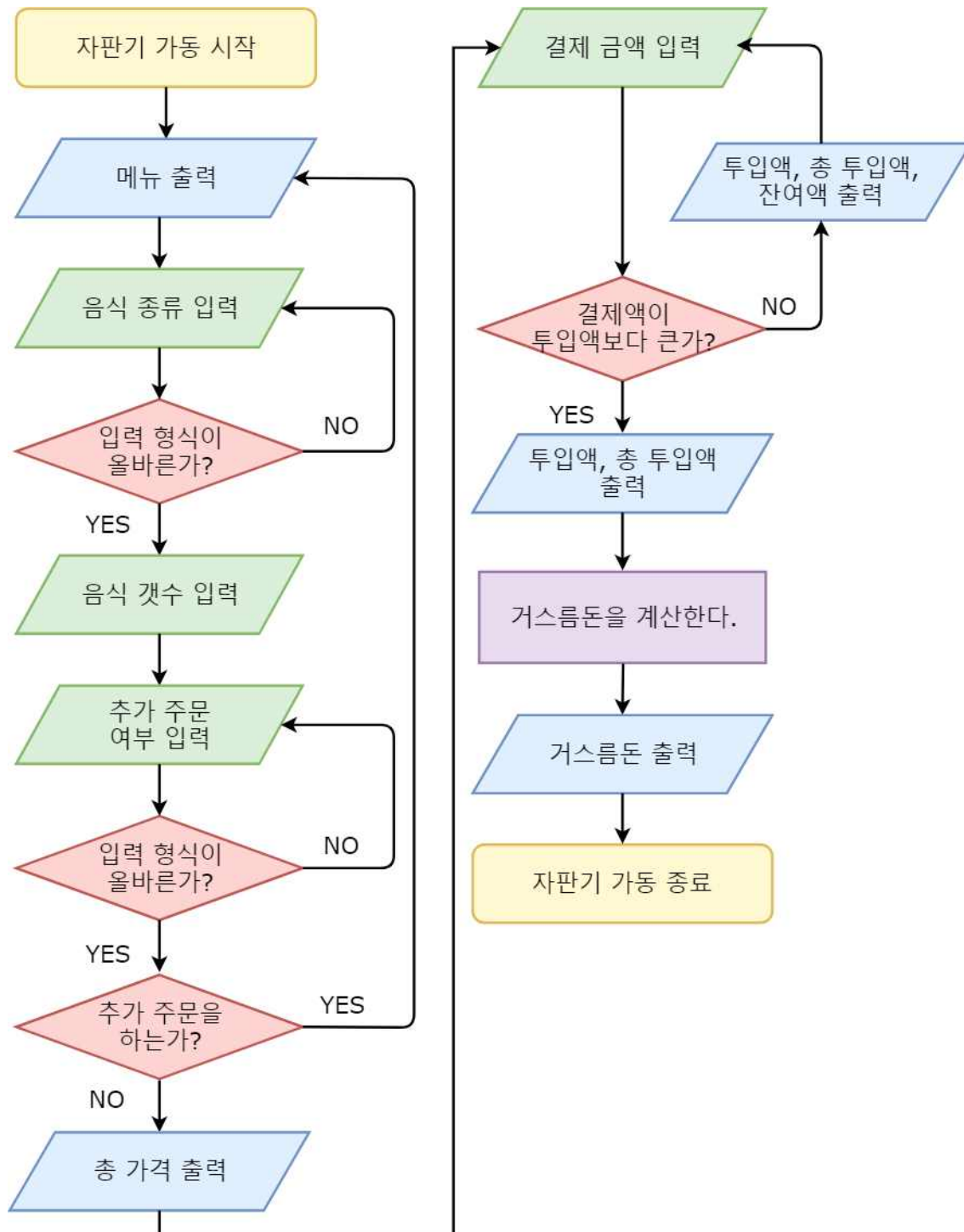


DONG-EUI UNIVERSITY
www.deu.ac.kr

목차

1. 프로그램 동작 과정 -----	p3
2. 함수별 동작 원리 -----	p4
3. 전체 코드 -----	p13
4. 구동 장면 -----	p17
5. 피드백 -----	p19
6. 초기 알고리즘 -----	p20

프로그램 동작 과정



함수별 동작 원리

001			
함수 이름	printDividingLine	반환형	void
사용 목적			
출력되는 결과값 사이를 구분해주는 선을 출력해주기 위해 사용되는 함수이다.			
동작원리			
<ol style="list-style-type: none"> 1. 선을 출력하기 전 enter키를 두 번 쳐줌으로 가독성을 높인다. 2. 구분선을 출력한다. 이 때, 반복문을 사용함으로 길이 조절이 쉽게 만들었다. 3. 다시 한 번 줄 바꿈을 두 번 하여, 후에 출력될 내용과도 공백을 둔다. 			
코드			
<pre>void printDividingLine() { printf("\n\n"); for (int i = 0; i < 60; i++) { printf("="); } printf("\n\n"); }</pre>			

002			
함수 이름	printMenu	반환형	void
사용 목적			
메뉴 출력을 위해 사용되는 함수이다.			
동작원리			
'Wt'와 공백을 적절히 넣어 출력해 보기 좋은 위치에 글이 자리 잡도록 출력한다.			
코드			
<pre>void printMenu() { printDividingLine(); printf("WtWt<<< 정보관 식당 메뉴 >>>WnWn"); printf("WtWt 1. 라면 (1500) Wn"); printf("WtWt 2. 라면밥 (2000) Wn"); printf("WtWt 3. 돈까스 (3000) Wn"); printf("WtWt 4. 부대찌개 (3500) Wn"); printDividingLine(); }</pre>			

003			
함수 이름	selectMenu	반환형	int
사용 목적			
문자 혹은 숫자를 입력해 메뉴를 선택하면 입력 내용을 판독해 음식 가격을 반환한다.			
동작원리			
<ol style="list-style-type: none"> 1. 문자열로 사용자에게 메뉴를 입력받는다. 2. string.h 헤더파일의 strcmp함수를 이용해 메뉴 이름 혹은 번호로 입력받은 문자열이 어떤 메뉴인지 판별 후 그 가격을 반환한다. 3. 판별하는 if문들을 무한루프를 돌려놓은 while문에 넣어두고 지정된 메뉴를 입력 할 때까지 계속 반복시킨다. 4. 지정된 메뉴 번호 혹은 이름이 아닌 경우 "올바른 형식의 메뉴를 입력해주세요."를 출력한다. 			
코드			
<pre> int selectMenu() { while(true) { char inputMenu[30] = { 0 }; printf("\t주문할 메뉴의 번호 혹은 이름을 입력해주세요 >> "); scanf("%s", inputMenu); printf("\n\n"); if (strcmp(inputMenu, "라면") == 0 strcmp(inputMenu, "1") == 0) { return 1500; } else if (strcmp(inputMenu, "라면밥") == 0 strcmp(inputMenu, "2") == 0) { return 2000; } else if (strcmp(inputMenu, "돈까스") == 0 strcmp(inputMenu, "3") == 0) { return 3000; } else if (strcmp(inputMenu, "부대찌개") == 0 strcmp(inputMenu, "4") == 0) { return 3500; } else { printf("올바른 형식의 메뉴를 입력해주세요."); printDividingLine(); } } } </pre>			

004

함수 이름	returnPrice	반환형	int
사용 목적			
가격의 총 합계를 반환한다.			
동작원리			
1. 매개변수로 음식의 가격(price)을 입력받는다. 2. 주문할 메뉴의 개수(numberOfDish)를 int형으로 입력받는다. 3. 만약 음수를 입력 시 -1을 곱하여 양수로 변환시킨다. 4. 총가격인 가격(price)과 개수를(numberOfDish)를 곱한 값을 반환한다.			
코드			
<pre>int returnPrice(int price) { int numberOfDish = 0; printf("\n\n\t메뉴의 개수를 입력해주세요 >> "); scanf("%d", &numberOfDish); if (numberOfDish < 1) { numberOfDish *= -1; } return numberOfDish * price; }</pre>			

005			
함수 이름	checkNextOrder	반환형	int
사용 목적			
<p>추가 주문 여부를 확인한다.</p>			
동작원리			
<ol style="list-style-type: none"> 1. 사용자에게 추가 주문 여부를 입력받는다. 2. strcmp함수를 다시 한 번 이용하여 'y'혹은 'Y'입력 시 1을 반환하고 'n' 혹은 'N'을 입력할 시에는 0을 반환한다. 3. 올바른 형식이 아닌 것이 입력되었을 때는 값을 반환하지 않고 다시 입력 받기 위해 조건이 참으로 고정된 while문에 넣었다. 			
코드			
<pre>int checkNextOrder() { while (true) { printDividingLine(); char check[50] = { 0 }; printf("\n\n\n추가 주문을 하시겠습니까? (Y / N) >> "); scanf("%s", check); if (strcmp(check, "Y") == 0 strcmp(check, "y") == 0) { return 1; } else if (strcmp(check, "N") == 0 strcmp(check, "n") == 0) { printDividingLine(); return 0; } else { printf("\n\n\n올바른 형식의 답변을 입력해주세요."); } printDividingLine(); } }</pre>			

006			
함수 이름	printFesultPrice	반환형	void
사용 목적			
가격의 총 합계를 출력한다.			
동작원리			
<ol style="list-style-type: none"> 1. 매개변수로 음식의 총 가격(price)을 입력받는다. 2. 가격과 구분선을 출력한다. 			
코드			
<pre>void printResultPrice(int price) { printf("결제하실 총 금액은 %d원 입니다.", price); printDividingLine(); }</pre>			

008			
함수 이름	calculateChange	반환형	void
사용 목적			
거스름돈을 계산하고 출력한다.			
동작원리			
<ol style="list-style-type: none"> 1. 매개변수로 거스름돈을 입력받는다. 2. int배열(money)에 거슬러줄 돈의 단위를 넣어둔다. 3. 큰 단위 돈을 거슬러 주고 남은 돈을 저장하는 변수(temp)를 만든다. 4. for문을 sizeof를 이용해 배열(money)의 크기만큼 작동하게 만든다. 5. 현재의 금액(temp) / 거슬러줄 돈의 단위(money[i])를 하였을 때, 0이 나오면 거슬러 줄 수 없다. 이 값이 자연수인 경우 거슬러 줄 수 있다는 의미이다. 이 경우 현재 계산하고 있는 단위(money[i])와 그 개수(temp / money[i])를 출력한다. 			
코드			
<pre> void calculateChange(int change) { int temp; int money[] = {50000, 10000, 5000, 1000, 500, 100, 50, 10}; temp = change; printDividingLine(); printf("WtWt아래의 거스름돈이 출력됩니다.WnWn"); for (int i = 0; i < sizeof(money) / sizeof(int); i++) { if (temp / money[i]) { printf("Wt>> %d원 : %d개Wn", money[i], temp / money[i]); temp %= money[i]; } } srand(time(NULL)); int orderNumber = rand()%100+1; printf("WnWnWtWt 주문번호는 %d번입니다.Wn", orderNumber); printDividingLine(); } </pre>			

함수 이름	main	반환형	int
사용 목적			
함수들을 조합하여 원하는 메뉴를 입력받아 주문을 하는 자판기이다.			
동작원리			
<ol style="list-style-type: none"> 무한루프를 도는 while문에 주문을 받는 함수들을 넣는다. <ul style="list-style-type: none"> - inputMenu : 메뉴의 가격을 저장한다. - price : 총 가격을 저장한다. - nextOrder : 추가 주문의 여부를 저장한다. - printMenu() : 메뉴를 출력한다. - selectMenu() : 선택된 메뉴의 가격을 반환해준다. - checkNextOrder() : 추가 주문의 여부를 반환해준다. 추가주문이 없어 nextOrder값이 0으로 변하면 while문이 종료된다. 계산을 위한함수들이 나온다. <ul style="list-style-type: none"> - change : 거스름돈을 저장한다. - printResultPrice() : 주문으로 얻은 총 가격을 출력해준다. - inputMoney() : 돈을 받고 거스름돈을 반환한다. - calculateChange : 각 단위별 거스름돈을 출력한다. 			
코드			
<pre> int main() { // nextOrder : 추가 주문 확인용 int nextOrder = 1, price = 0, change = 0; while (nextOrder) { // 추가 주문이 없을 때까지 반복 int inputMenu; printMenu(); inputMenu = selectMenu(); // 메뉴 선택 price += returnPrice(inputMenu); // 총 가격 : (메뉴 개수 X 가격) 값 nextOrder = checkNextOrder(); // 추가 주문 여부 : 참, 거짓 } printResultPrice(price); // 총액 출력 : 총 가격 출력 change = inputMoney(price); // 금액 투입 : 거스름돈 리턴 calculateChange(change); // 거스름돈 계산 : 거스름돈 계산, 출력 } </pre>			

작성 코드

```
#include <stdio.h>
#include <string.h>
#include <random>
#include <time.h>
#pragma warning (disable : 4996)

/*
=====
학과 : 창의소프트웨어공학부
학번 : 20193148
분반 : 2분반
이름 : 황진주
제출 일자 : 19.09.30
=====
*/

int countMenu[5];

// 구분선 출력
void printDividingLine() {
    printf("\n\n");
    for (int i = 0; i < 60; i++) {
        printf("=");
    }
    printf("\n\n");
}

// 메뉴 출력
void printMenu() {
    printDividingLine();
    printf("\t\t\t<<< 정보관 식당 메뉴 >>>\n\n");
    printf("\t\t\t1. 라면 (1500) \n");
    printf("\t\t\t2. 라면밥 (2000) \n");
    printf("\t\t\t3. 돈까스 (3000) \n");
    printf("\t\t\t4. 부대찌개 (3500) \n");
    printDividingLine();
}

// 메뉴 선택 및 가격 리턴
int selectMenu() {
    while(true) {
        char inputMenu[30] = { 0 };    // 입력값 저장 장소
```

```

        printf("\t주문할 메뉴의 번호 혹은 이름을 입력해주세요 >> ");
        scanf("%s", inputMenu);
        printf("\n\n");
        if (strcmp(inputMenu, "라면") == 0 || strcmp(inputMenu, "1") == 0) {
            return 1500;
        }
        else if (strcmp(inputMenu, "라면밥") == 0 || strcmp(inputMenu, "2") ==
0) {
            return 2000;
        }
        else if (strcmp(inputMenu, "돈까스") == 0 || strcmp(inputMenu, "3") ==
0) {
            return 3000;
        }
        else if (strcmp(inputMenu, "부대찌개") == 0 || strcmp(inputMenu, "4")
== 0) {
            return 3500;
        }
        else {
            printf("올바른 형식의 메뉴를 입력해주세요.");
            printDividingLine();
        }
    }
}

// 선택한 메뉴의 개수 입력
int returnPrice(int price) {
    int numberOfDish = 0;
    printf("\n\n\t메뉴의 개수를 입력해주세요 >> ");
    scanf("%d", &numberOfDish);
    if (numberOfDish < 1) {          // 음수 처리
        numberOfDish *= -1;
    }
    return numberOfDish * price;    // 메뉴 총 가격 반환
}

// 추가 주문 여부 확인
int checkNextOrder() {
    while (true) {
        printDividingLine();
        char check[50] = { 0 };    // 입력값 저장 장소
        printf("\n\n\t추가 주문을 하시겠습니까? (Y / N) >> ");
        scanf("%s", check);
    }
}

```

```

        if (strcmp(check, "Y") == 0 || strcmp(check, "y") == 0) {
            return 1;
        }
        else if (strcmp(check, "N") == 0 || strcmp(check, "n") == 0) {
            printDividingLine();
            return 0;
        }
        else {
            printf("WnWnWt올바른 형식의 답변을 입력해주세요.");
        }
        printDividingLine();
    }
}

// 총 가격 출력
void printResultPrice(int price) {
    printf("결제하실 총 금액은 %d원 입니다.", price);
    printDividingLine();
}

// 금액 투입
int inputMoney(int price) {
    int input = 0;
    while (true) {
        printf("Wt결제할 금액을 투입해주세요. >> ");
        int temp;
        scanf("%d", &temp);

        if (temp < 1) {           // 음수 처리
            temp *= -1;
        }

        printf("WnWn");
        input += temp;
        if (input < price) {
            printf("WtWtWt투입한 금액 : %dWnWnWtWtWt 총 투입 금액 : %dWnWnWtWtWt 남은 금액 : %dWnWn", temp, input, price-input);
        }
        else {
            printf("WtWtWt투입한 금액 : %dWnWnWtWtWt 총 투입 금액 : %dWnWn", temp, input);
            return input - price;
        }
    }
}

```

```

}

// 거스름돈 계산
void calculateChange(int change) {
    int temp;
    int money[] = {50000, 10000, 5000, 1000, 500, 100, 50, 10};

    temp = change;
    printDividingLine();
    printf("₩₩₩아래의 거스름돈이 출력됩니다.₩₩₩\n");

    for (int i = 0; i < sizeof(money) / sizeof(int); i++) {
        if (temp / money[i]) {
            printf("₩₩>> %d원 : %d개₩₩\n", money[i], temp / money[i]);
            temp %= money[i];
        }
    }
    printDividingLine();
}

int main() {
    countMenu[5] = {0};
    // nextOrder : 추가 주문 확인용
    int nextOrder = 1, price = 0, change = 0;

    while (nextOrder) {
        int inputMenu;
        printMenu();

        inputMenu = selectMenu(); // 메뉴 선택
        :           메뉴의 가격을 리턴
        price += returnPrice(inputMenu); // 총 가격          :
(메뉴의 갯수 X 가격) 값 리턴
        nextOrder = checkNextOrder(); // 추가 주문 여부      :
참, 거짓

    }
    printResultPrice(price); // 메뉴 출력          :
        총가격 출력
    change = inputMoney(price); // 금액 투입
    :           거스름돈 리턴
    calculateChange(change); // 거스름돈 계산 :
    거스름돈 개수 계산 및 출력

}

```


구동장면

<사용 모습>

=====
<<< 정보관 식당 메뉴 >>>

1. 라면 (1500)
2. 라면밥 (2000)
3. 돈까스 (3000)
4. 부대찌개 (3500)

=====
주문할 메뉴의 번호 혹은 이름을 입력해주세요 >> 돈까스

메뉴의 갯수를 입력해주세요 >> 6

=====
추가 주문을 하시겠습니까? (Y / N) >> n

=====
결제하실 금액은 18000원 입니다.

=====
결제할 금액을 투입해주세요. >> 83560

투입한 금액 : 83560

총 투입 금액 : 83560

=====
아래의 거스름돈이 출력됩니다.

>> 50000원 : 1개
>> 10000원 : 1개
>> 5000원 : 1개
>> 500원 : 1개
>> 50원 : 1개
>> 10원 : 1개

=====
계속하려면 아무 키나 누르십시오 . . .

<예외처리>

적용 : 메뉴선택, 추가주문

```
=====
<<< 정보관 식당 메뉴 >>>
1. 라면 (1500)
2. 라면밥 (2000)
3. 돈까스 (3000)
4. 부대찌개 (3500)

=====

주문할 메뉴의 번호 혹은 이름을 입력해주세요 >> 짜장면

올바른 형식의 메뉴를 입력해주세요.

=====

주문할 메뉴의 번호 혹은 이름을 입력해주세요 >> 6

올바른 형식의 메뉴를 입력해주세요.

=====

주문할 메뉴의 번호 혹은 이름을 입력해주세요 >> _
```

<추가주문 선택 시 화면>

```
=====

추가 주문을 하시겠습니까? (Y / N) >> y

=====

<<< 정보관 식당 메뉴 >>>
1. 라면 (1500)
2. 라면밥 (2000)
3. 돈까스 (3000)
4. 부대찌개 (3500)

=====

주문할 메뉴의 번호 혹은 이름을 입력해주세요 >> _
```

피드백

처음 과제에 임할 때에 자판기의 처리 과정은 간단할 것이니 깊게 고민하지 않고 코드를 바로 짜기 시작했었습니다. 그런데 코드를 짜다보니 새로운 기능들과 생각지 못했지만 필요했던 기능들이 떠오르기 시작했습니다. 이 것들이 한데 엉키고 정리가 되지 않자 첫 코드를 완성하고 실행을 하니 오류들이 많이 발생했었습니다. 마구잡이로 코드를 수정하다 결국 저는 작성한 코드를 모두 지우고 찬찬히 알고리즘을 작성했습니다. 알고리즘을 작성하다가 생긴 의문이나 오류가 있었던 부분도 따로 적어 해결방안을 간단히 적으며 생각을 정리했습니다. (내용이 길어 [내용](#)은 따로 공간을 만들어 첨부되어 있습니다.) 먼저 각각의 큰 틀을 나누고 그 큰 틀을 어떻게 구성하여 구현할 것인지를 고민하여 적었더니, 함수간의 꼬임현상과 전역 변수는 사라지게 되었고, 이전에 이상이 없던 코드는 더욱 간결하고 명확하게 작성이 되었습니다.

각각의 함수가 구현이 어렵지 않은 것이었지만 그것들이 모이게 되면 복잡해 질 수 있다는 것, 많은 내용의 코드가 아닐 때는 미련 없이 모두 지우고 해보는 것도 좋은 방법이라는 것을 배울 수 있었습니다.

현재의 코드는 예외처리가 부족한 것같이 미숙한 면이 많습니다. 메뉴 추가 삭제와 같이 더 추가해보고 싶은 기능이 있었지만 구현하지 못 하기도 했습니다. 현재의 제 지식의 한계가 있어 못 한 것과 좋은 알고리즘이 떠오르지 않아 구현해내지 못 한 것들을 앞으로 배워나가며 추가해보고 싶습니다.

초기 알고리즘

< 나의 식권 만들기 >

1. 메뉴 출력

- 출력

2. 메뉴 선택

- 문자열로 받음 : 번호를 선택 or 메뉴 이름 기입
- 예외
 - 1) 없는 번호 선택
 - 2) 없는 메뉴 선택처리 -> 다시 입력을 받게 함

3. 추가 주문 여부 확인

- Y / N
- 만약 Y : 2번으로 돌아감
- 만약 N : 4번 실행
- 예외
 - 1) 이상한 거 입력처리 -> 다시 입력 받게

4. 가격 고지

- 각 메뉴의 갯수 파악
- 각 메뉴의 종류 파악
- 종류에 따른 가격 파악
- 출력 : 메뉴 이름 / 주문한 갯수 / 갯수에 따른 가격 / 총 가격

5. 입금

- 입력 받음
- 예외 :
 - 1) 입력 가격이 총 가격 보다 적음
 - > 현재 입금액 / 부족한 금액 출력
 - > 입금액 <= 총 가격 때 까지

6. 거스름돈 계산

- %랑 /계산으로
50000 / 10000 / 5000 / 1000 / 500 / 100
/ 50 / 10 / 5 / 1원 계산

7. 거스름돈 출력

- 출력 조건 (나와야할 지폐가 1개 이상일 때)

++ 주문 수정, 주문 취소 기능을 넣어보고 싶다!

- 스택으로 쌓으면 되지 않을까?
- 메뉴마다 배열을 만들어서 카운트를 하자!

< ★내가 막힌 부분★ >

1. 각 메뉴와 그 가격을 어떻게 저장할 까?
 - 구조체 배열
 - if문 덩어리
2. 각 메뉴가 몇 개가 입력 되었는지 어떻게 파악하나?
 - 인트 배열을 이용
3. 메뉴 입력시 판별을 어떻게 해야하는가?
 - strcmp
 - if문 덩어리
4. 입력받은 메뉴와 기존 메뉴 정보를 어디에 저장하나?
 - 기존 메뉴
: 구조체 배열에 일단 넣어두자
 - 입력 메뉴
: 캐릭터 배열에 입력