Point of Sales System

Final Team presentation

T4

201211178 민경훈

201211187 배승현

201311283 송형선

201611299 정희승

INDEX

1. Self Test Result

1.1 failed tests

1.2 Modified code

1.1 failed cases by test

Identifier	Input specification	output specification	Pass/ Fail
T4_POS.STP.000.004	목록제거를 통해 과자를 모두 제거한다.	물1개, 라면1개로 캐션화면과 고객화면에 업데이트 되어있는지 확인한다.	Fail
T4_POS.STP.005.000	환불화면에서 임의의 영수증번호 111111을 입력한다.	환불이 불가능한 영수증이라는 메세지를 출력한다.	Fail

Failed Cases : 2

1.1 failed case (T4_POS.STP.000.004)

~/code_T4

```
==Customer Screen==================================
상 품              수 량

============================================

==Cashier Screen===================================
상 품              수 량
물                  1

============================================


상 품  입 력 을  계 속  하 시 겠 습 니 까  (y/n) : y
바 코 드 를  입 력 해 주 세 요  : 100

물  상 품 을  2개  선 택 하 였 습 니 다 . ( 재 고  : 98 )
+: 수 량  추 가 , -: 수 량  감 소 , d: 품 목  제 거 , f: 목 록  확 정
명 령 어 를  입 력 해 주 세 요  : d
```

품목 제거 버튼(d)로 입력중인 상품 입력은 제거 하나,
앞서 입력된 상품은 제거하지 않음.

~/code_T4

```
==Customer Screen==================================
상 품              수 량

============================================

==Cashier Screen===================================
상 품              수 량
물                  1

============================================


상 품  입 력 을  계 속  하 시 겠 습 니 까  (y/n) :
```

1.1 failed case (T4_POS.STP.000.004)

~/code_T4_before

```
                continue;
        }
        index = checkRecipt(SalesData, listLen, Item_code);

        //수 량  조 절
        int choiceNum;
        if ((itd + Item_code)->stock == 0) {
                printf("\n%s 개 수 가  부 족 합 니 다 \n\n", (itd + Item_code)->item);
                choiceNum = 0;
        }
        else {
                if (index == -1) choiceNum = adjustQuantity(itd, Item_code, 1);
                else choiceNum = adjustQuantity(itd, Item_code, SalesData[index][1] + 1);
                if (choiceNum == 0) printf("%s 상 품 이  삭 제 되 었 습 니 다 .\n", (itd + Item_code)->item);
                else printf("%s 상 품 을  %d개  선 택 하 였 습 니 다 . ( 재 고  : %d )\n", (itd + Item_code)->item, ch
oiceNum, ((itd + Item_code)->stock) - choiceNum);

                //판 매  목 록  저 장
                if (choiceNum != 0) {
                        if (index == -1) {
                                SalesData[listLen][0] = Item_code;
                                SalesData[listLen][1] = choiceNum;
                                listLen++;
                        }
                        else SalesData[index][1] = choiceNum;
                }
                printSaleProcessCustomer(itd,SalesData, listLen);
        }
        // 추 가  입 력 을  위 한  옵 션
        char op;
        printf("\n상 품  입 력 을  계 속  하 시 겠 습 니 까 ");
        while (1) {
                printf(" (y/n) : ");
                getchar();
                scanf("%c", &op);
                                                                197,16-30     52%
```

main.c

품목 제거 버튼(d)로 제거한 경우 choiceNum으로 0을 반환함.

하지만 choiceNum이 0일 때 기존 SalesData를 검사하지 않음.

1.2 Modified code (T4_POS.STP.000.004)

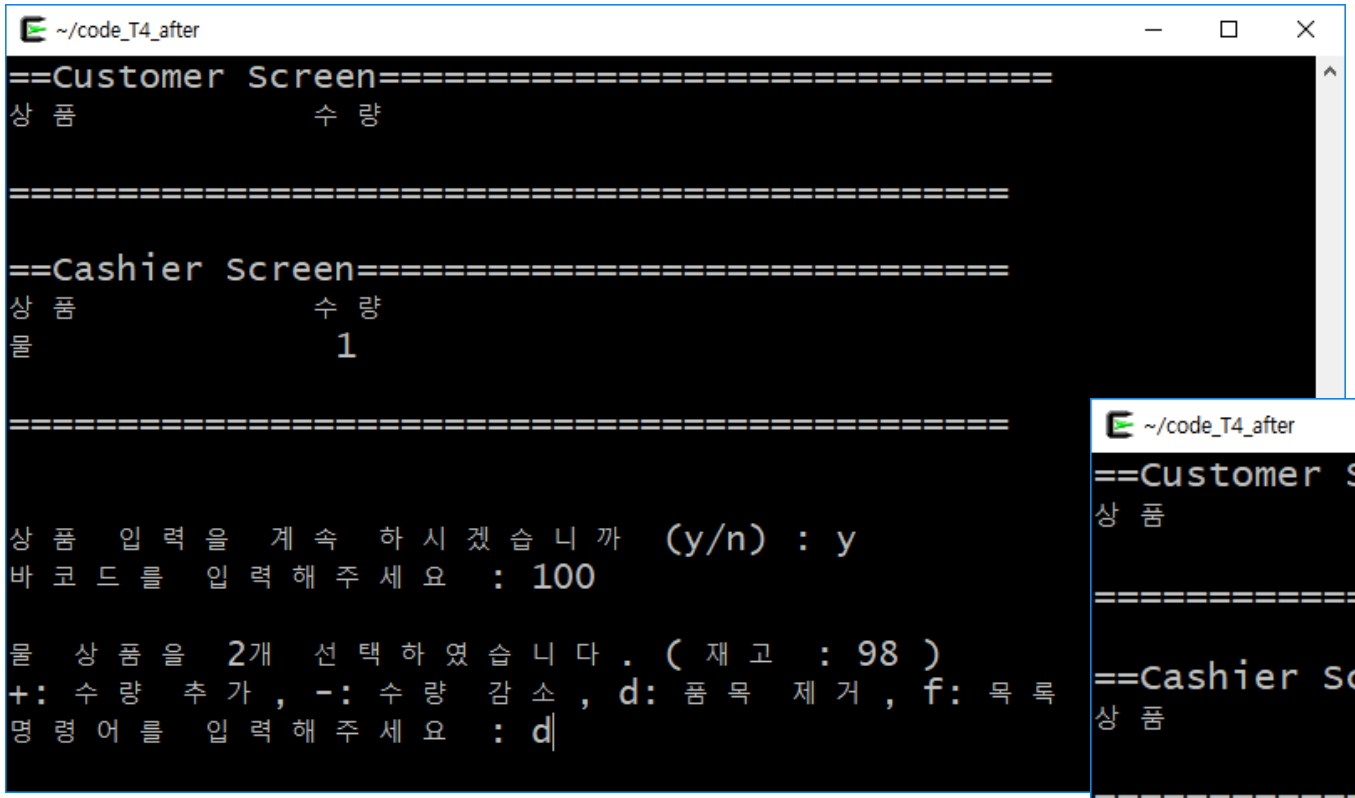
~/code_T4

```
                Print_Error_Message("Wrong Bar Code Input!");
                continue;
        }
        index = checkRecipt(SalesData, listLen, Item_code);

        //수 량  조 절
        int choiceNum;
        if ((itd + Item_code)->stock == 0) {
                printf("\n%s 개 수 가  부 족 합 니 다 \n\n", (itd + Item_code)->item);
                choiceNum = 0;
        }
        else {
                if (index == -1) choiceNum = adjustQuantity(itd, Item_code, 1);
                else choiceNum = adjustQuantity(itd, Item_code, SalesData[index][1] + 1);
                if (choiceNum == 0){
                        printf("%s 상 품 이  삭 제 되 었 습 니 다 .\n", (itd + Item_code)->item);
                        if(index != -1){
                                for(int remove_index = index; remove_index < listLen - 1 ; remove_index ++){
                                        SalesData[remove_index][1] = SalesData[remove_index + 1][1];
                                        SalesData[remove_index][0] = SalesData[remove_index + 1][0];
                                }
                                listLen --;
                                SalesData[listLen][0] = -1;
                                SalesData[listLen][1] = -1;
                        }
                }
                else printf("%s 상 품 을  %d개  선 택 하 였 습 니 다 .  ( 재 고  : %d )\n", (itd + Item_code)->item, choiceNum, ((itd + Item_code)->stoc
k) - choiceNum);

                //판 매  목 록  저 장
                if (choiceNum != 0) {
                        if (index == -1) {
                                SalesData[listLen][0] = Item_code;
                                SalesData[listLen][1] = choiceNum;
                                listLen++;
                        }
                        else SalesData[index][1] = choiceNum;
                }
                printSaleProcessCustomer(itd,SalesData, listLen);
        }
        // 추 가  입 력 을  위 한  옵 션
        char op;
        printf("\n상 품  입 력 을  계 속  하 시 겠 습 니 까 ");
                                                                207,16-44      51%
```

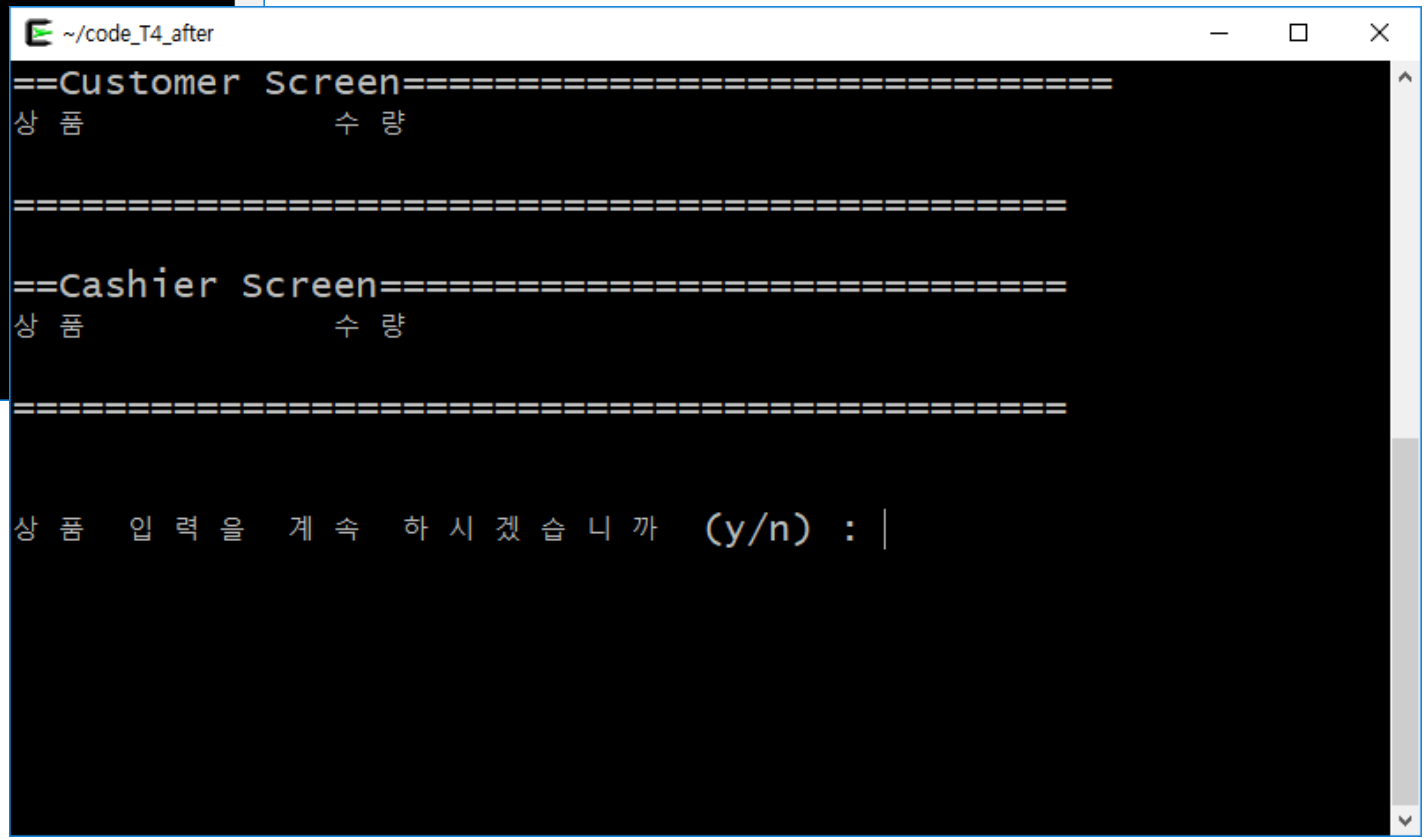
품목 제거 버튼(d)로 제거한 경우 choiceNum으로 0을 반환함.
choiceNum이 0일 때 판매 목록을 조회한 index가 존재할 때, 해당 판매 목록도 제거.

main.c

1.2 Modified code (T4_POS.STP.000.004)



품목 제거 버튼(d)로 입력했던 상품을
제거했을 때, 품목이 제거되는 것을 확인.



POS.STP.000.004
Pass

1.1 failed case (T4_POS.STP.005.000)

~/code_T4

```
$ ./main.exe
```

상 품	단 가	바 코 드
과 자	1000	001
아 이 스 크 림	1500	010
과 일	3000	011
물	500	100
라 면	800	101
음 료 수	1200	110
커 피	2000	111

```
1 은  종 료 , 2 는  판 매 , 3 은  환 불 , 4 는  재 고 확 인  : 3
영 수 증  번 호 를  입 력  : 1000
Segmentation fault (core dumped)

User@LAPTOP-3FQUR4PE ~/code_T4
$
```

영수증 번호가 영수증 바코드 형식 또는 Integer가 아닐 경우 Segmentation fault

1.1 failed case (T4_POS.STP.005.000)

main.c

```
~/code_T4_before
        //영 수 증  번 호  입 력  받 아  조 회
        printf("영 수 증  번 호 를  입 력  : ");
        TimeName * rtn = (TimeName *)malloc(sizeof(TimeName));
        scanf("%s", receipt_code);
        if(!LookupRefundList(receipt_code)){
                Print_Error_Message("이 미  환 불  처 리  완 료 된  영 수 증  입 니 다 .");
                return;
        }
        setRefundDate(rtn, receipt_code);
        setSaleManagementServerName(rtn, saleFileServer);
        SoldData * sold_data = (SoldData *)malloc(sizeof(SoldD
@@@
```

refundItem.c

```
~/code_T4_before
/*
 환 불  날 짜  저 장
*/
void setRefundDate(TimeName * tn, char * code)
{
        int YYYY, MM, DD, hh, mm;
        char date[30];
        strcpy(date, code);
        char * ptr = strtok(date, ".");

        tn->year = atoi(ptr);
        ptr = strtok(NULL, ".");
        tn->mon = atoi(ptr);
        ptr = strtok(NULL, ".");
        tn->day = atoi(ptr);
        ptr = strtok(NULL, ".");
        tn->hour = atoi(ptr);
        ptr = strtok(NULL, ".");
        tn->min = atoi(ptr);
}

/*
                                                23,1-8        9%
```

영수증 코드를 receipt_code 라는 문자열 변수로 입력 받아 저장.

setRefundDate 함수를 통해 시간 변수 rtn과 recipt_code를 넘겨줌.

함수 내에서 code를 Integer 값들인 YYYY, MM, DD, hh, mm 으로 변환.
(strtok: 문자열 분리, atoi: 문자열을 숫자로 변환)

이 과정에서 숫자가 아닌 값이나 변수 숫자가 맞지않으면 에러.

1.2 Modified code (T4_POS.STP.005.000)

~/code_T4_after

```c
int isNumString(char * s,int len)
{
        size_t size = strlen(s);
        if(len != (int)size) return 0;
        for(int i=0;i<(int) size;i++){
                if(s[i] == '.') continue;
                if(s[i] < '0' || s[i] > '9') return 0;
        }
        return 1;
}
```

isNumString이라는 새로운 함수 생성.
(문자열과 숫자를 입력 받아, 문자열의 길이가 숫자와 맞는지와 문자열이 “.”을 제외한 숫자 값인지 검사.)

문자를 받아온 순간과 strtok으로 분리 시킬 때마다 isNumString 함수를 통해 길이와 숫자 검사를 시행한다.

만일 반환 값이 0(false)인 경우, 함수 종료. (예외처리)

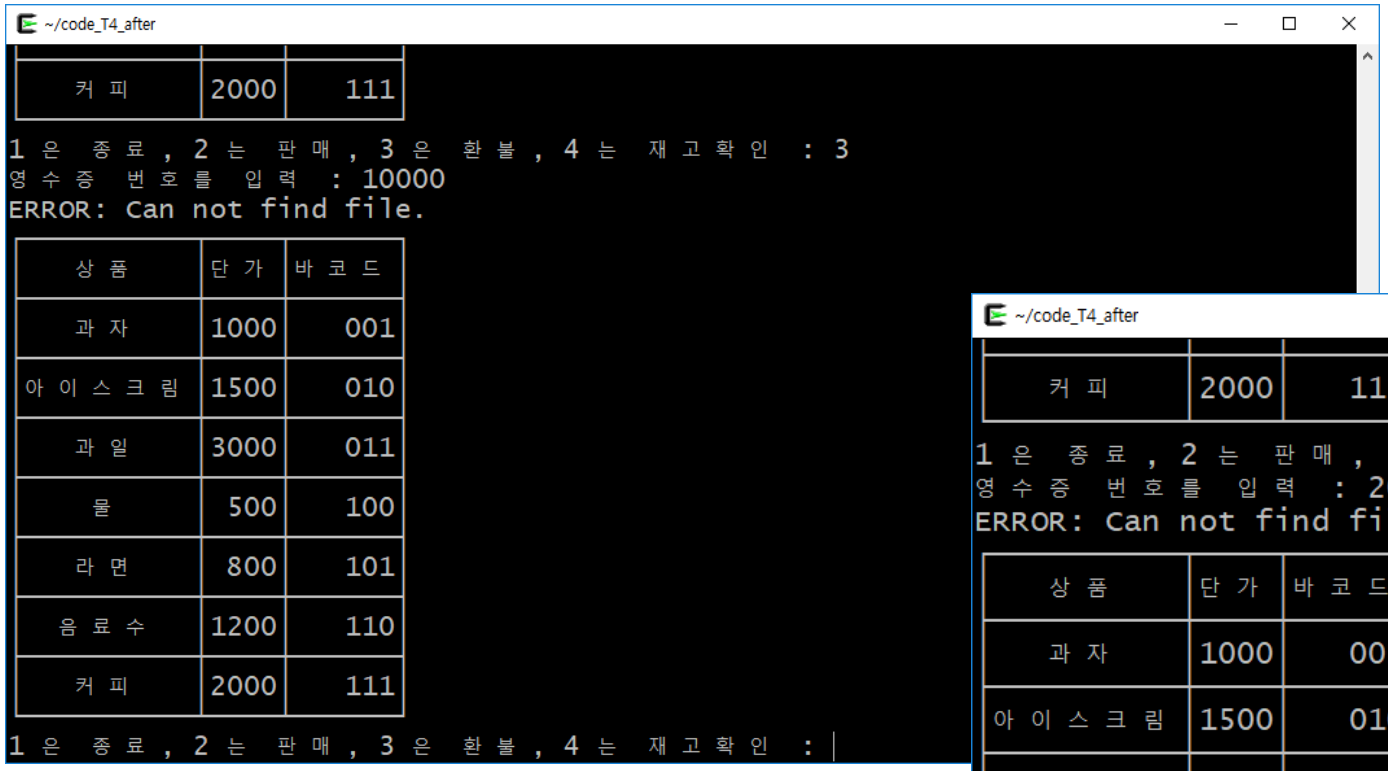
refundItem.c

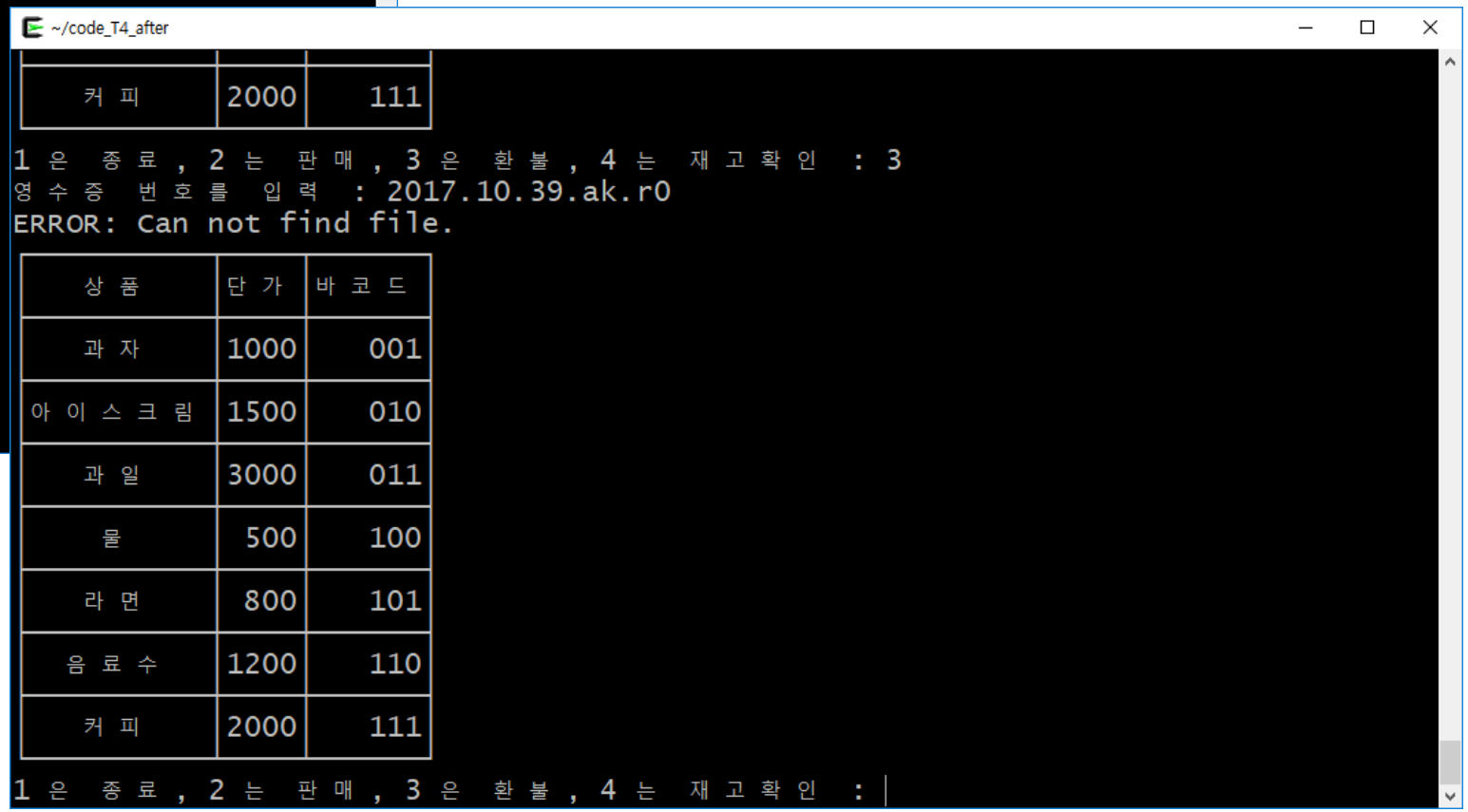
~/code_T4_after

```c
/*
 환 불  날 짜  저 장
*/
int setRefundDate(TimeName * tn, char * code)
{
        if(!isNumString(code,(int)strlen("YYYY.MM.DD.hh.mm"))) return 0;
        int YYYY, MM, DD, hh, mm;
        char date[30];
        strcpy(date, code);
        char * ptr = strtok(date, ".");

        if(!isNumString(ptr,4)) return 0;
        tn->year = atoi(ptr);

        ptr = strtok(NULL, ".");
        if(!isNumString(ptr,2)) return 0;
        tn->mon = atoi(ptr);

        ptr = strtok(NULL, ".");
        if(!isNumString(ptr,2)) return 0;
        tn->day = atoi(ptr);

        ptr = strtok(NULL, ".");
        if(!isNumString(ptr,2)) return 0;
        tn->hour = atoi(ptr);

        ptr = strtok(NULL, ".");
        if(!isNumString(ptr,2)) return 0;
        tn->min = atoi(ptr);
        return 1;
}
```

32,1-8 5%

1.2 Modified code (T4_POS.STP.005.000)





1. Result

Identifier	Input specification	output specification	Pass/ Fail
T4_POS.STP.000.004	목록제거를 통해 과자를 모두 제거한다.	물1개, 라면1개로 캐션화면과 고객화면에 업데이트 되어있는지 확인한다.	Fail -> Pass
T4_POS.STP.005.000	환불화면에서 임의의 영수증번호 111111을 입력한다.	환불이 불가능한 영수증이라는 메세지를 출력한다.	Fail -> Pass

Fixed Cases : 2

2. Test Result by Team 2

2.1 failed cases

2.2 Coped with Failure & Modified code

2.1 failed cases

Identifier	expect	actual	Pass/ Fail
T2_1.2.2	고객 화면에 업데이트	고객 화면 자체가 없음	Fail
T2_1.3.4	품목확정	segmentation fault("품목제거" 를 누르면, 빈 화면이 뜨고 "상품을 더 입력하시겠습니까?" 선택란에 "N" 을 입력하는 프로세스를 거쳐야 정상 작동합니다. 처음 테스트 때 품목 확정을 눌렀을 때 프로그램이 꺼져서 놀랐네요.)	Fail
T2_1.3.6.2	파일 이름 맞는지 refund_YYMMDDhhmm	refund_YYMMDDhhmm	Fail
T2_2.1.2	유효하지 않은 바코드	segmentation fault(영수증 번호 길이가 아예 다르면 바로 터집니다: 2017.11.16.xx.xx 가 없으면 정상작동 하지만 막 11223344 따위의 길이 다른 입력은 소화하지 못하고 바로 꺼집니다)	Fail
T2_2.5.2	날짜가 지난 후 환불	상품 파일이 재 생성 되지 않아 테스트 자체가 불가	Fail
T2_3.3.1	상품 파일이 재생성 되는지	생성되지 않음	Fail

T4_POS.STP.000.004 와 동일한 T2_1.3.3는 Pass

T2_2.1.2는 T4_POS.STP.005.000 와 동일한 Fail case

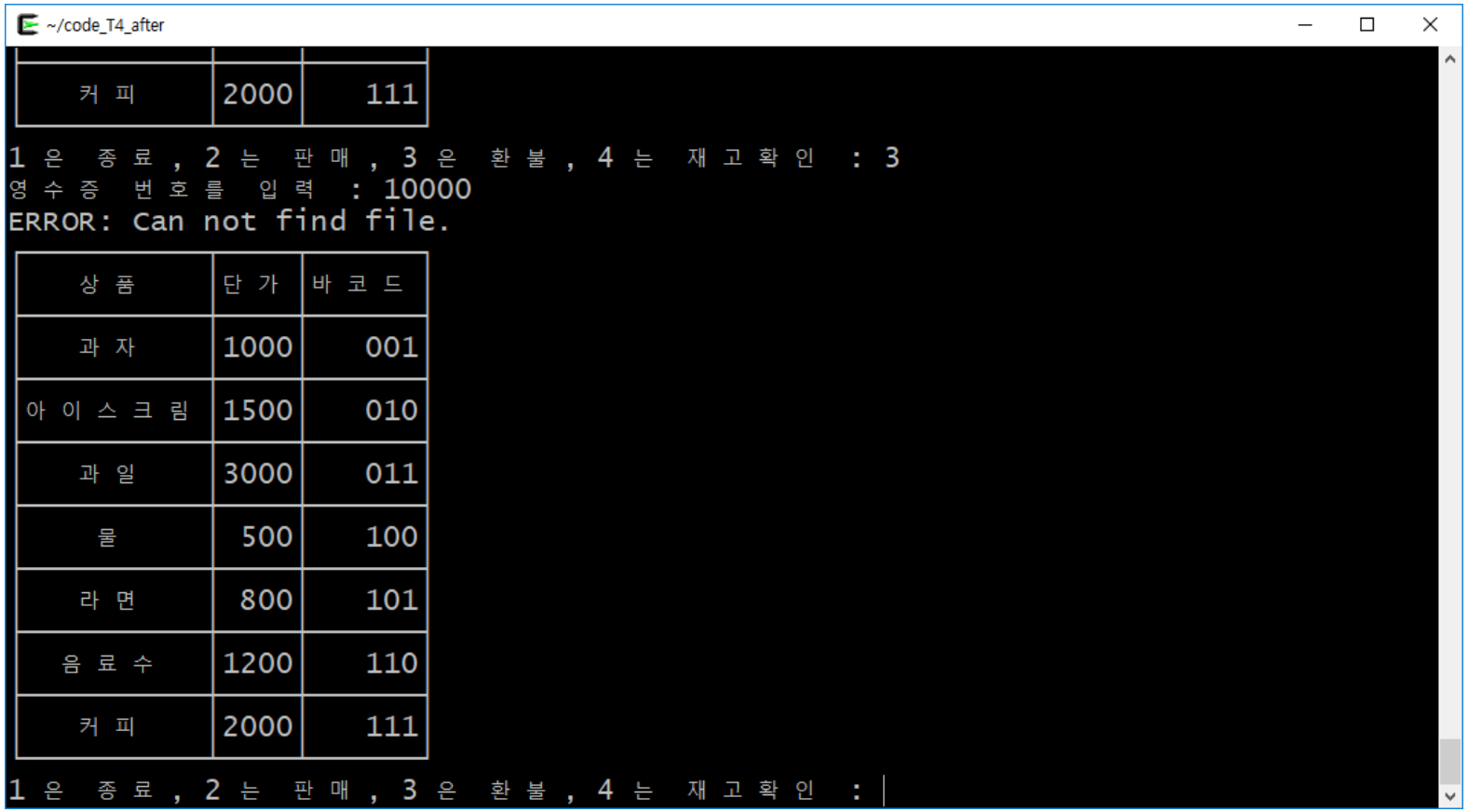
Failed Cases : 6

2.1 failed case (T2_1.2.2)

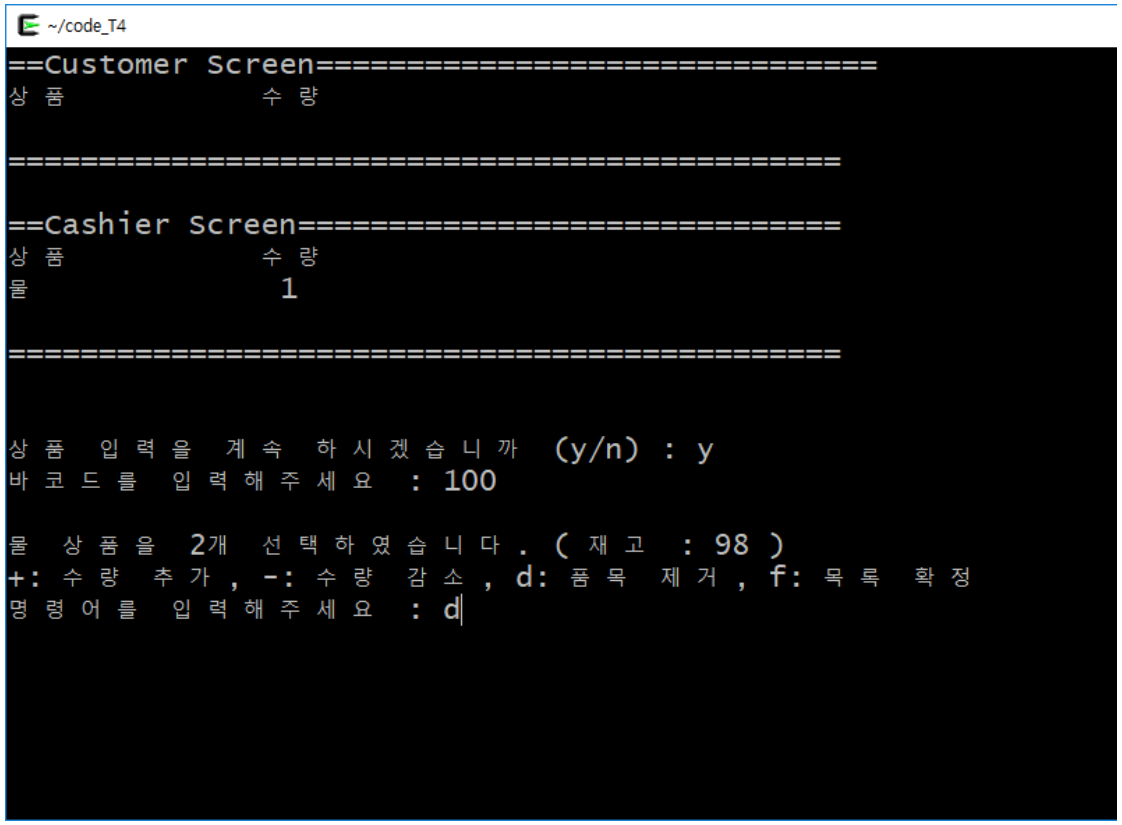
1.2 정보 업데이트

1.2.1 캐셔 화면에 업데이트 -- n: 실시간 반영이 안됨

1.2.2 고객 화면에 업데이트 -- fail: 고객 화면 자체가 없음



2.2 refutation (T2_1.2.2)



SRS에서 판매, 환불, 재고확인 가 완료된 후 스크린을 띄운다고 명시.

3.2.2 환불

3.2.2.1 Function

POST는 사용자가 캐셔 화면(터치 스크린)의 기본 화면에서 환불 버튼을 누를 시 환불을 진행할 수 있다.

POST는 바코드 센서에서 영수증 바코드를 읽었을 때 바코드로 읽은 영수증이 이전에 발급 했던 영수증인지 재고 서버에 확인을 요청한다.

재고 서버에서 환불이 가능하다는 응답이 올 경우, 터치스크린을 통해 환불을 완료할 수도 있고 환불을 취소할 수도 있다.

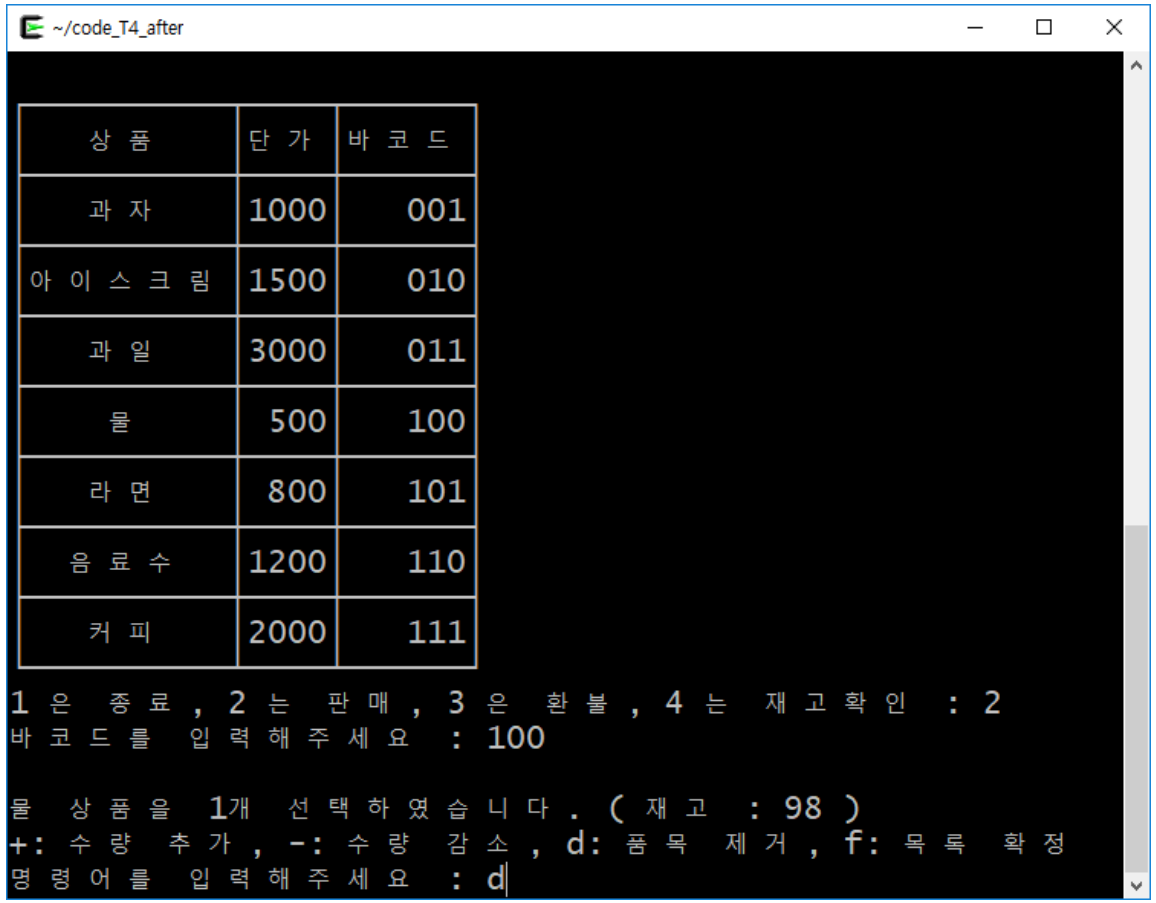
재고 서버에서 환불이 불가능하다는 응답이 올 경우, 캐셔 화면(터치 스크린)와 고객 화면에 각각 "해당 영수증은 환불이 불가능 합니다." 라는 메시지를 띄운다.

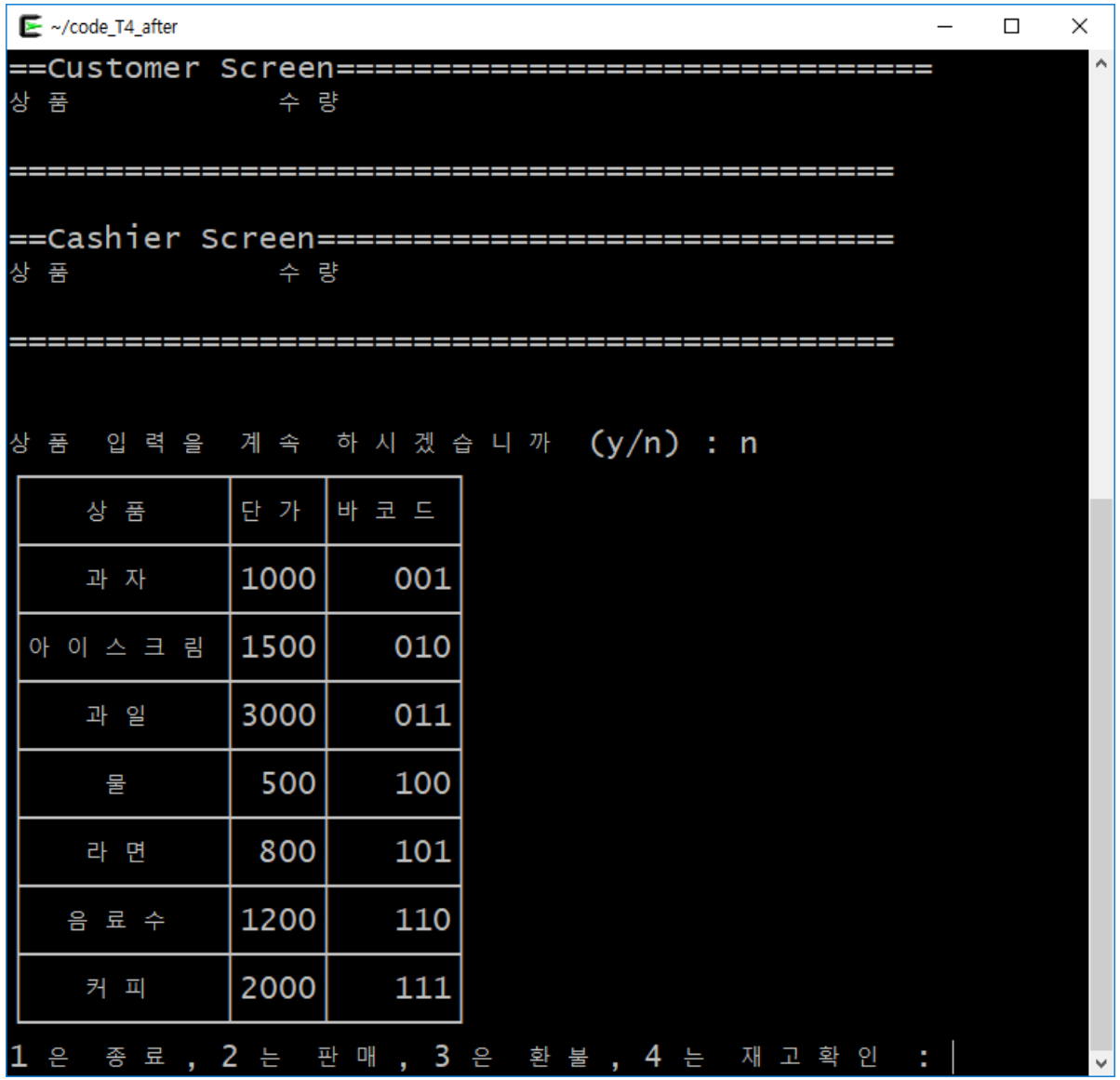
POST는 환불이 완료되면 캐셔 화면(터치 스크린)와 고객 화면에 각각 환불 정보를 띄운다.

2.1 failed case (T2_1.3.4)

1.3.4 품목확정 -- fail: segmentation fault

("품목제거"를 누르면, 빈 화면이 뜨고 "상품을 더 입력하시겠습니까?" 선택란에 "N"을 입력하는 프로세스를 거쳐야 정상 작동합니다. 처음 테스트 때 품목 확정을 눌렀을 때 프로그램이 꺼져서 놀랐네요)





2.2 refutation (T2_1.3.4)

~/code_T4_after

과 자	1000	001
아 이 스 크 림	1500	010
과 일	3000	011
물	500	100
라 면	800	101
음 료 수	1200	110
커 피	2000	111

```
1 은  종 료 , 2 는  판 매 , 3 은  환 불 , 4 는  재 고 확 인  : 2
바 코 드 를  입 력 해 주 세 요  : 100

물  상 품 을  1개  선 택 하 였 습 니 다 . ( 재 고  : 98 )
+: 수 량  추 가 , -: 수 량  감 소 , d: 품 목  제 거 , f: 목 록  확 정
명 령 어 를  입 력 해 주 세 요  : f
```

~/code_T4_after

```
==Customer Screen=================================
상 품            수 량

==================================================

==Cashier Screen==================================
상 품            수 량
물               1

==================================================


상 품  입 력 을  계 속  하 시 겠 습 니 까  (y/n) : n
```

~/code_T4_after

```
==Customer Screen=================================
total : 500
현 금  : 800
거 스 름  돈  : 300

==================================================

==Cashier Screen==================================

상 품            수 량            가 격
물               1                500
total : 500
현 금  : 800
거 스 름  돈  : 300

==================================================
```

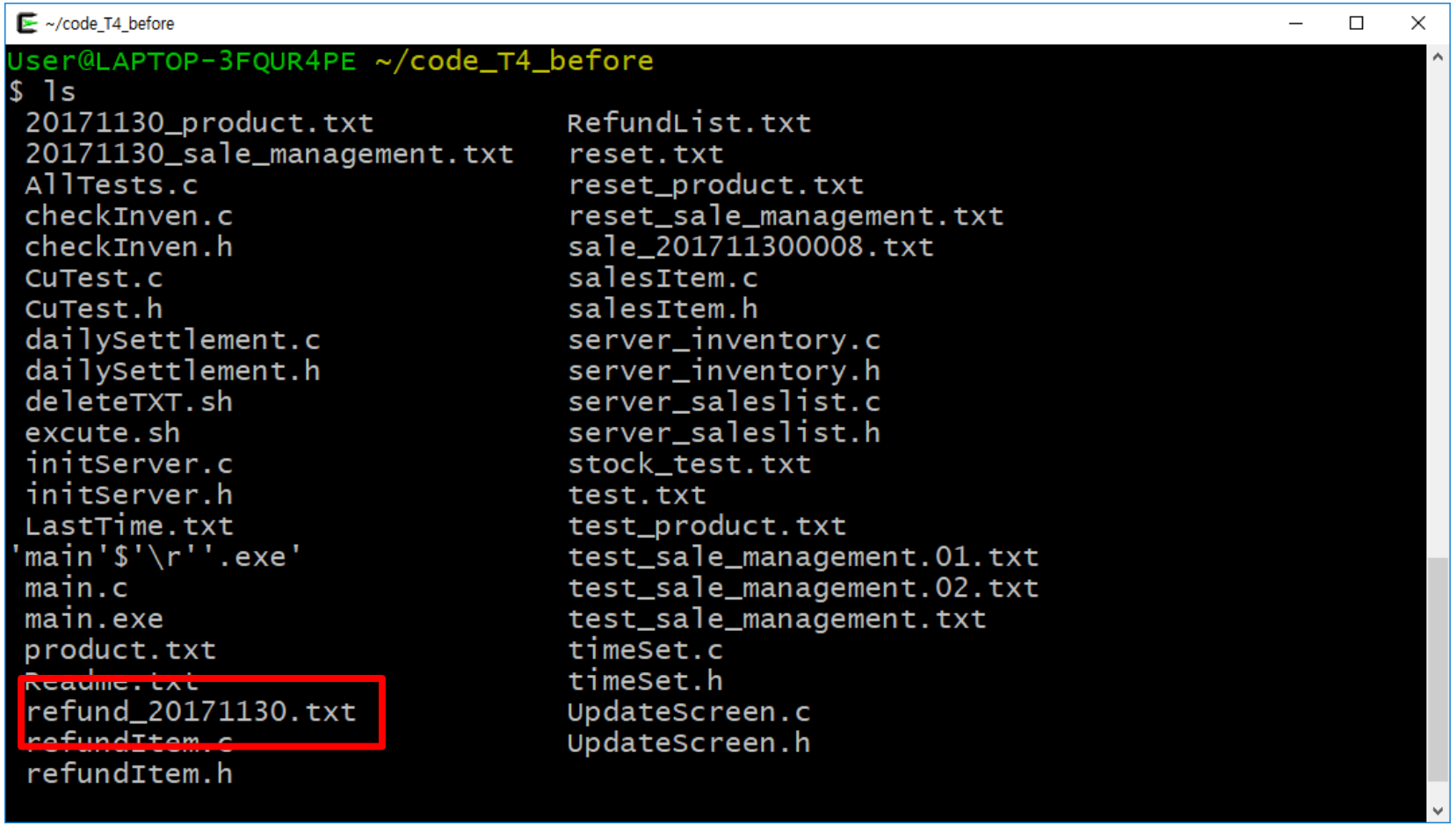
품목 제거, 품목 확정 후 Segmentation fault가 확인 되지 않음.

2.1 failed case (T2_1.3.6.2)

1.3.6 영수증 출력

1.3.6.1 날짜 시간 포맷 맞는지 -- pass

1.3.6.2 파일 이름 맞는지 -- fail: 시간 없음(refund_YYMMDDhhmm 여야 하지만 refund_YYMMDD 까지만 생성)



2.2 Modified Code (T2_1.3.6.2)

~/code_T4_before

```
        strcat(result, ".txt");
        free(today);
}

void namingRefundRecipt(TimeName* tn, char * result) {
        char * today = (char*)malloc(strlen("YYYYMMDD") + 1);
        setYYYYMMDD(tn, today);
        strcpy(result, "refund_");
        strcat(result, today);
        strcat(result, ".txt");
        free(today);
}
                                                    152,1-8       80%
```

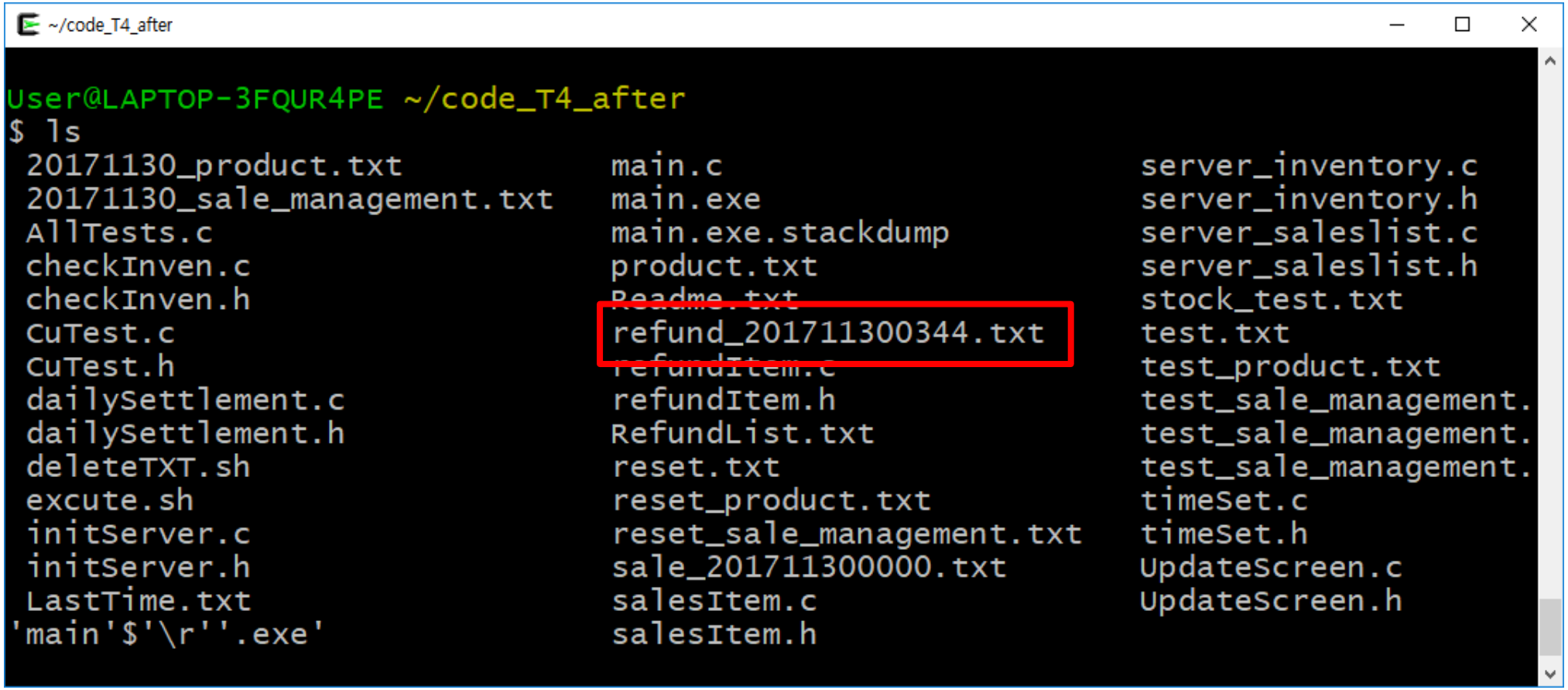
~/code_T4_after

```
}

void namingRefundRecipt(TimeName* tn, char * result) {
        char * today = (char*)malloc(strlen("YYYYMMDDhhmm") + 1);
        setYYYYMMDDhhmm(tn, today);
        strcpy(result, "refund_");
        strcat(result, today);
        strcat(result, ".txt");
        free(today);
}
                                                    154,1-8       80%
```

환불 영수증 형식은
“refund_YYYYMMDDhhmm.txt”

그에 맞게 문자열 길이와 형식 수정

2.2 Modified Code (T2_1.3.6.2)



2.1 failed case (T2_2.1.2)

2.1 바코드 읽기

2.1.1 유효한 바코드 -- pass

2.1.2 유효하지 않은 바코드 -- fail: segmentation fault(영수증 번호 길이가 아예 다르면 바로 터집니다: 2017.11.16.xx.xx 가 없으면 정상작동하지만 막 11223344 따위의 길이 다른 입력은 소화하지 못하고 바로 꺼집니다)

T2_2.1.2는 T4_POS.STP.005.000와 동일한 Fail case

2.1 failed case (T2_2.5.2 & T2_3.3.1)

2.5.2 날짜가 지난 후 환불 – fail: 상품 파일이 재생성 되지 않아 테스트 자체가 불가

~/code_T4_before

아 이 스 크 림	1500	010
과 일	3000	011
물	500	100
라 면	800	101
음 료 수	1200	110
커 피	2000	111

```
1 은  종 료 , 2 는  판 매 , 3 은  환 불 , 4 는  재 고 확 인  :
++++++++++++++++++++++++++++
+  하 루 가  지 나 갔 습 니 다 .  +
++++++++++++++++++++++++++++
1
시 스 템 을  종 료 합 니 다 ..

User@LAPTOP-3FQUR4PE ~/code_T4_before
$
```

~/code_T4_before

```
User@LAPTOP-3FQUR4PE ~/code_T4_before
$ ls
 20171130_product.txt            reset.txt
 20171130_sale_management.txt    reset_product.txt
 20171131_product.txt            reset_sale_management.txt
 AllTests.c                      sale_201711300000.txt
 checkInven.c                    salesItem.c
 checkInven.h                    salesItem.h
 CuTest.c                        server_inventory.c
 CuTest.h                        server_inventory.h
 dailySettlement.c               server_saleslist.c
 dailySettlement.h               server_saleslist.h
 deleteTXT.sh                    settle_20171130.txt
 excute.sh                       stock_test.txt
 initServer.c                    test.txt
 initServer.h                    test_product.txt
 LastTime.txt                    test_sale_management.01.txt
'main'$'\r''.exe'                test_sale_management.02.txt
 main.c                          test_sale_management.txt
 main.exe                        timeSet.c
 product.txt                     timeSet.h
 Readme.txt                      UpdateScreen.c
 refundItem.c                    UpdateScreen.h
 refundItem.h
```

상품 파일(서버)은 재생성 됨.

2.1 failed case (T2_2.5.2 & T2_3.3.1)

~/code_T4_before

```
 refundItem.c                    UpdateScreen.h
 refundItem.h

User@LAPTOP-3FQUR4PE ~/code_T4_before
$ ./main.exe
```

상 품	단 가	바 코 드
과 자	1000	001
아 이 스 크 림	1500	010
과 일	3000	011
물	500	100
라 면	800	101
음 료 수	1200	110
커 피	2000	111

```
1 은  종 료 , 2 는  판 매 , 3 은  환 불 , 4 는  재 고 확 인  : 3
영 수 증  번 호 를  입 력  : 2017.11.30.00.00
```

~/code_T4_before

```
==Customer Screen================================

판 매  날 짜  : 2017년  11월  30일  0시  0분
환 불  금 액  : 500

==================================================

==Cashier Screen=================================

판 매  날 짜  : 2017년  11월  30일  0시  0분
상 품            수 량            단 가            금 액
물                1               500              500
환 불  금 액  : 500

==================================================

환 불 을  정 말  진 행 하 시 겠 습 니 까  (y/n) : y
```

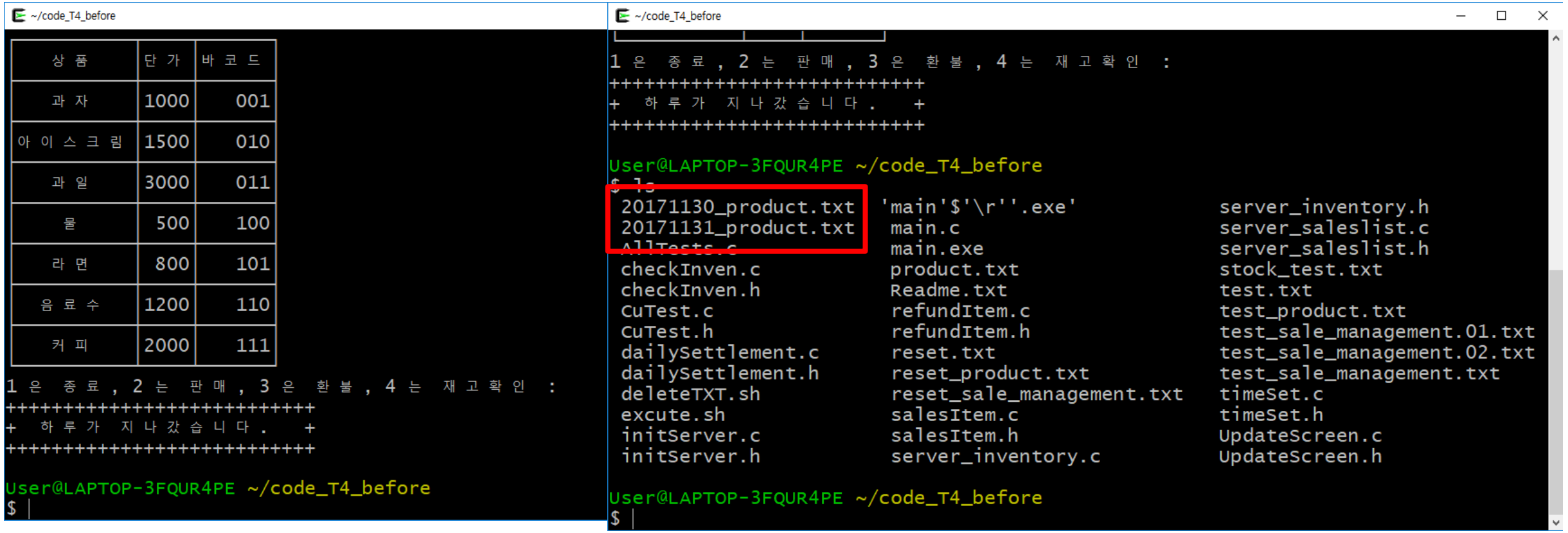
하루가 지난 후의 영수증도 환불이 됨.

2. Result

Identifier	expect	actual	Pass/ Fail
T2_1.2.2	고객 화면에 업데이트	고객 화면 자체가 없음	Not Fix
T2_1.3.4	품목확정	segmentation fault(“품목제거” 를 누르면, 빈 화면이 뜨고 “상품을 더 입력하시겠습니까?” 선택란에 “N” 을 입력하는 프로세스를 거쳐야 정상 작동합니다. 처음 테스트 때 품목 확정을 눌렀을 때 프로그램이 꺼져서 놀랐네요.)	Not Fix
T2_1.3.6.2	파일 이름 맞는지 refund_YYMMDDhhmm	refund_YYMMDDhhmm	Fail -> Pass
T2_2.1.2	유효하지 않은 바코드	segmentation fault(영수증 번호 길이가 아예 다르면 바로 터집니다: 2017.11.16.xx.xx 가 없으면 정상작동 하지만 막 11223344 따위의 길이 다른 입력은 소화하지 못하고 바로 꺼집니다)	Fail -> Pass
T2_2.5.2	날짜가 지난 후 환불	상품 파일이 재 생성 되지 않아 테스트 자체가 불가	Not Fix
T2_3.3.1	상품 파일이 재생성 되는지	생성되지 않음	Not Fix

Fixed Cases : 2

failed case (additional case)



서버 초기화 이후, 첫 실행 시 판매 목록 파일(서버)이 초기화 되지 않아, 판매 과정을 수행하지 않을 시 정산시 비정상 종료 발견.

failed case (additional case)

~/code_T4_before

```
        free(productPrice);
        free(productBarcode);
        return 1;
}

int Reset_Saleslist(char * FileName) {
        FILE * f;
        if (strlen(FileName) == 0) {
                return 0;
        }
        if ((f = fopen(FileName, "r")) != NULL) {

                if ((f = fopen(FileName, "w")) != NULL) {

                }
                else {
                        printf("reset saleslist fail!\n");
                        return 0;
                }
        }
        else {
                return 0;
        }
        fclose(f);
        return 1;
}
                                                        58,1-8          49%
```

~/code_T4_after

```
int Reset_Saleslist(char * FileName) {
        FILE * f;
        if (strlen(FileName) == 0) {
                return 0;
        }
        if ((f = fopen(FileName, "r")) == NULL) {

                if ((f = fopen(FileName, "w")) != NULL) {

                }
                else {
                        printf("reset saleslist fail!\n");
                        return 0;
                }
        }
        else {
                return 0;
        }
        fclose(f);
        return 1;
}
                                                        69,2-9          51%
```

initServer.c

파일이 이미 존재할 때 불러오는 과정.

Modified Case (additional case)

~/code_T4_after

상 품	단 가	바 코 드
과 자	1000	001
아 이 스 크 림	1500	010
과 일	3000	011
물	500	100
라 면	800	101
음 료 수	1200	110
커 피	2000	111

```
1 은  종 료 , 2 는  판 매 , 3 은  환 불 , 4 는  재 고 확 인  :
++++++++++++++++++++++++++
+  하 루 가  지 나 갔 습 니 다 .   +
++++++++++++++++++++++++++
```

~/code_T4_after

```
++++++++++++++++++++++++++
1
시 스 템 을  종 료 합 니 다 ..

User@LAPTOP-3FQUR4PE ~/code_T4_after
$ ls
 20171130_product.txt            refundItem.c
 20171130_sale_management.txt    refundItem.h
 20171131_product.txt            reset.txt
 20171131_sale_management.txt    reset_product.txt
 AllTests.c                      reset_sale_management.txt
 checkInven.c                    salesItem.c
 checkInven.h                    salesItem.h
 CuTest.c                        server_inventory.c
 CuTest.h                        server_inventory.h
 dailySettlement.c               server_saleslist.c
 dailySettlement.h               server_saleslist.h
 deleteTXT.sh                    settle_20171130.txt
 excute.sh                       stock_test.txt
 initServer.c                    test.txt
 initServer.h                    test_product.txt
 LastTime.txt                    test_sale_management.01.txt
```

상품 파일(서버)과 판매 목록 파일(서버) 모두 재생성 됨.

따라서 비정상 종료 해결.

3. Demo

3. Demo

In Cygwin

./main.exe
or
./excute.sh
or
gcc *.c –o main
./main.exe