

## 2 部データ構造とアルゴリズム I レポート課題

18NC021

カトリ スザン

# Contents

<b>1</b>	<b>課題 1</b>	<b>2</b>
1.1	問題 1-1	2
1.1.1	shohin.h	2
1.1.2	shohin.c	2
1.1.3	テストプログラム 1-1	2
1.1.4	実行結果	3
1.1.5	プログラムの解説	3
1.2	問題 1-2	4
1.2.1	shohin.h	4
1.2.2	shohin.c	4
1.2.3	uriage.h	4
1.2.4	uriage.c	5
1.2.5	テストプログラム 1-2	5
1.2.6	実行結果	5
1.2.7	プログラムの解説	6
1.3	問題 1-3	7
1.3.1	shohin.h	7
1.3.2	shohin.c	7
1.3.3	uriage.h	7
1.3.4	uriage.c	8
1.3.5	テストプログラム 1-3	8
1.3.6	実行結果	9
1.3.7	プログラムの解説	9
1.4	問題 1-4	10
1.4.1	shohin.h	10
1.4.2	shohin.c	10
1.4.3	uriage.h	10
1.4.4	uriage.c	11
1.4.5	テストプログラム 1-4	11
1.4.6	実行結果	12
1.4.7	プログラムの解説	12
1.5	考察	13

## 1 課題 1

### 1.1 問題 1-1

#### 1.1.1 shohin.h

```
#ifndef C_ALGO_REPORT1_SHOHIN_H
#define C_ALGO_REPORT1_SHOHIN_H

#endif //C_ALGO_REPORT1_SHOHIN_H

#define ZEI (1.08)
typedef struct{
    char* name;
    int tanka;
    int sotozei;
} SHOHIN;
extern SHOHIN shohin[];
void printshohin(SHOHIN s);
```

#### 1.1.2 shohin.c

```
#define N 3

#include <stdio.h>
#include "shohin.h"

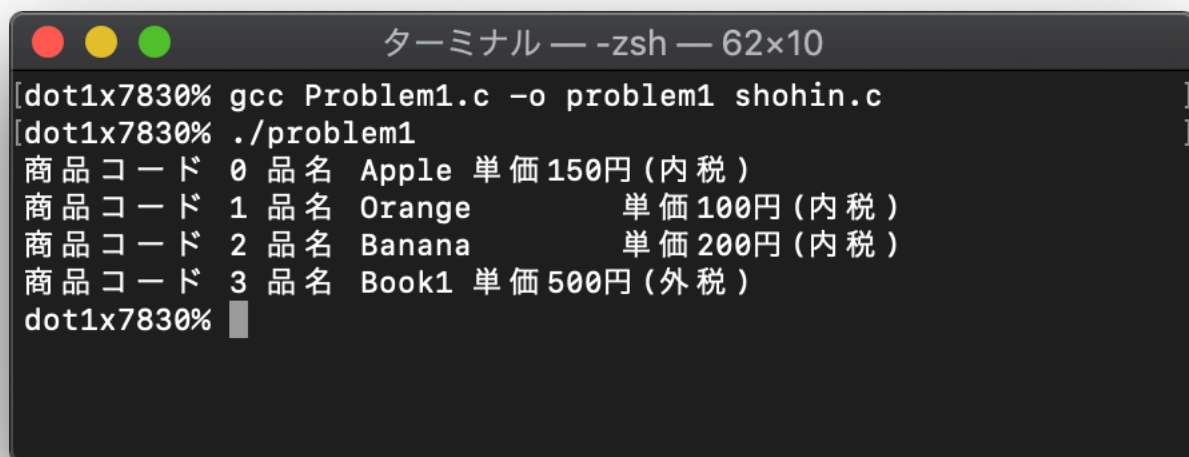
SHOHIN shohin[] = {{"Apple", 150},
                   {"Orange", 100},
                   {"Banana", 200},
                   {"Book1", 500, 1},
                   {"", 0}};

void printshohin(SHOHIN s) {
    printf("%s\t単価%d円(%s)", s.name, s.tanka, s.sotozei ? "外税" : "内税");
}
```

#### 1.1.3 テストプログラム 1-1

```
#include <stdio.h>
#include "shohin.h"
int main(void){
    int i;
    for(i=0;shohin[i].tanka!=0; i++){
        printf("商品コード %d 品名 ",i);
        printshohin(shohin[i]);
        printf("\n");
    }
    return 0;
}
```

#### 1.1.4 実行結果



```
ターミナル — -zsh — 62x10
[dot1x7830% gcc Problem1.c -o problem1 shohin.c ]
[dot1x7830% ./problem1 ]
商品コード 0 品名 Apple 単価 150円 (内税)
商品コード 1 品名 Orange 単価 100円 (内税)
商品コード 2 品名 Banana 単価 200円 (内税)
商品コード 3 品名 Book1 単価 500円 (外税)
dot1x7830% █
```

図 1: 問題 1-1 実行結果

#### 1.1.5 プログラムの解説

[shohin.c] の printshohin 関数は SHOHIN 型の `s` を引数とし、戻り値は `void` である。この関数は渡された構造体の `name(:string)` と `tanka(:int)` プロパティを `print` する。また最後に `stozei(:int)` プロパティの値が `truthy` 値の場合は '外税' を `print` し、`falsy` 値の場合は '内税' を `print` する。

## 1.2 問題 1-2

### 1.2.1 shohin.h

```
#ifndef C_ALGO_REPORT1_SHOHIN_H
#define C_ALGO_REPORT1_SHOHIN_H

#endif //C_ALGO_REPORT1_SHOHIN_H

#define ZEI (1.08)
typedef struct{
    char* name;
    int tanka;
    int sotozei;
} SHOHIN;
extern SHOHIN shohin[];
void printshohin(SHOHIN s);
```

### 1.2.2 shohin.c

```
#define N 3

#include <stdio.h>
#include "shohin.h"

SHOHIN shohin[] = {{ "Apple", 150},
                   { "Orange", 100},
                   { "Banana", 200},
                   { "Book1", 500, 1},
                   { "", 0}};

void printshohin(SHOHIN s) {
    printf("%s\t単価%d円(%s)", s.name, s.tanka, s.sotozei ? "外税" : "内税");
}
```

### 1.2.3 uriage.h

```
#ifndef C_ALGO_REPORT1_URIAGE_H
#define C_ALGO_REPORT1_URIAGE_H

#endif //C_ALGO_REPORT1_URIAGE_H

typedef struct {
    int code;
    int num;
} URIAGE;

int printUriage(URIAGE* q);
```

### 1.2.4 uriage.c

```
#include <stdio.h>
#include "uriage.h"
#include "shohin.h"

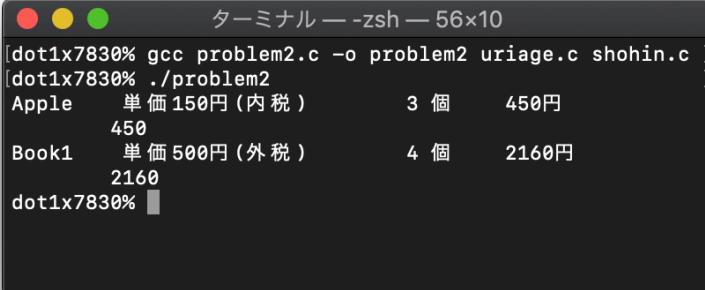
int printUriage(URIAGE *q) {

    char *const product_name = shohin[q->code].name;
    const int item_price = shohin[q->code].tanka;
    const int tax = shohin[q->code].sotozei;
    const int total_sales = q->num;
    const int total_price =
        tax ? (item_price * total_sales) * ZEI : item_price * total_sales;
    printf("%s\t 単価%d円(%s)\t%d 個\t%d円 \n", product_name, item_price,
        tax ? "外税" : "内税", total_sales, total_price);
    return total_price;
}
```

### 1.2.5 テストプログラム 1-2

```
#include <stdio.h>
#include "shohin.h"
#include "uriage.h"
int main(void){
    URIAGE u1={0,3};
    URIAGE u2={3,4};
    int shokei;
    shokei = printUriage(&u1);
    printf("\t%d\n", shokei);
    shokei = printUriage(&u2);
    printf("\t%d\n", shokei);
    return 0;
}
```

### 1.2.6 実行結果



```
ターミナル — zsh — 56x10
dot1x7830% gcc problem2.c -o problem2 uriage.c shohin.c ]
dot1x7830% ./problem2 ]
Apple  単価150円(内税)      3 個      450円
      450
Book1   単価500円(外税)     4 個     2160円
      2160
dot1x7830% █
```

図 2: 問題 1-2 実行結果

### 1.2.7 プログラムの解説

[uriage.c] の printUriage 関数は URIAGE 型のポインター (\*q) を引数とし、戻り値が total\_price(:int) の関数である。この関数に渡されるポインターには shohin(:SHOHIN) のインデックスを指定している code(:int) プロパティとそのインデックスの商品の売り上げ数を指定している num(:int) プロパティを持っている構造体のアドレスが格納されている。この関数は構造体 \*q で指定された shohin(:SHOHIN) のあるインデックスに入っている商品の売り上げ総額を計算して total\_price(:int) に格納する。そして商品の名前、単価、売り上げ数、売り上げ総額を print する。最後に total\_price(:int) を return する。shohin の sotozei プロパティの値が truthly の場合は単価に +8%(税金) を加算して計算する。falsy の場合は税金を加算せずに計算する。また、この関数では shohin のプロパティの値を二回以上参照していて、毎回構造体のプロパティを指定するのが面倒くさい為それぞれのプロパティ用に変数を用意してその値を格納して使っている。

## 1.3 問題 1-3

### 1.3.1 shohin.h

```
#ifndef C_ALGO_REPORT1_SHOHIN_H
#define C_ALGO_REPORT1_SHOHIN_H

#endif //C_ALGO_REPORT1_SHOHIN_H

#define ZEI (1.08)
typedef struct{
    char* name;
    int tanka;
    int sotozei;
} SHOHIN;
extern SHOHIN shohin[];
void printshohin(SHOHIN s);
```

### 1.3.2 shohin.c

```
#define N 3

#include <stdio.h>
#include "shohin.h"

SHOHIN shohin[] = {{ "Apple", 150},
                   { "Orange", 100},
                   { "Banana", 200},
                   { "Book1", 500, 1},
                   { "", 0}};

void printshohin(SHOHIN s) {
    printf("%s\t単価%d円(%s)", s.name, s.tanka, s.sotozei ? "外税" : "内税");
}
```

### 1.3.3 uriage.h

```
#ifndef C_ALGO_REPORT1_URIAGE_H
#define C_ALGO_REPORT1_URIAGE_H

#endif //C_ALGO_REPORT1_URIAGE_H

typedef struct {
    int code;
    int num;
}URIAGE;

int printUriage(URIAGE* q);
int printUriageArray(URIAGE u[]);
```



### 1.3.4 uriage.c

```
#include <stdio.h>
#include "uriage.h"
#include "shohin.h"

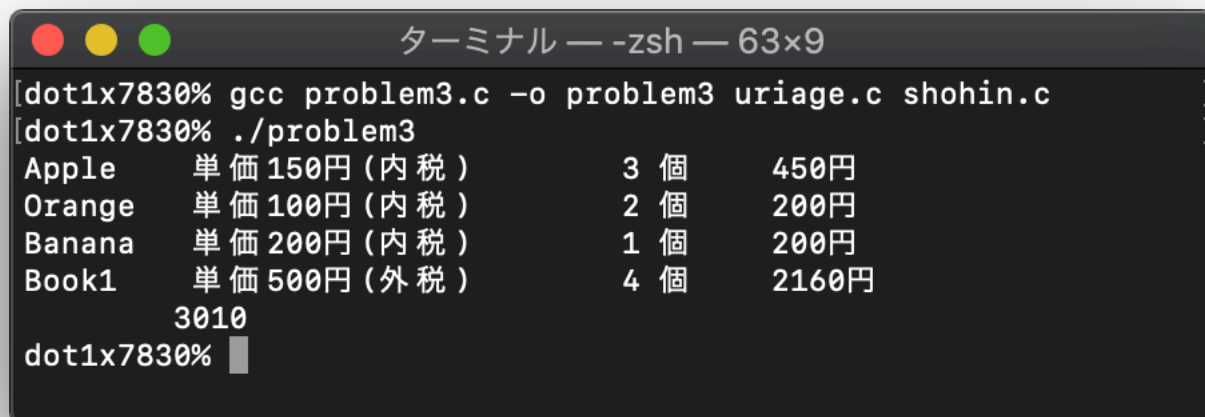
int printUriage(URIAGE *p) {
    char *const product_name = shohin[p->code].name;
    const int item_price = shohin[p->code].tanka;
    const int tax = shohin[p->code].sotozei;
    const int total_sales = p->num;
    const int total_price =
        tax ? (item_price * total_sales) * ZEI : item_price * total_sales;
    printf("%s\t 単価%d円(%s)\t%d 個\t%d円 \n", product_name, item_price,
        tax ? "外税" : "内税", total_sales, total_price);
    return total_price;
}

int printUriageArray(URIAGE u[]) {
    int shokei = 0;
    for (int i = 0; u[i].code != -1; i++) {
        shokei += printUriage(&u[i]);
    }
    return shokei;
}
```

### 1.3.5 テストプログラム 1-3

```
#include <stdio.h>
#include "shohin.h"
#include "uriage.h"
int main(void){
    URIAGE u[]={ {0,3},{1,2},{2,1},{3,4},{-1}};
    int shokei;
    shokei = printUriageArray(u);
    printf("\t%d\n", shokei);
    return 0;
}
```

### 1.3.6 実行結果



```
ターミナル — -zsh — 63x9
[dot1x7830% gcc problem3.c -o problem3 uriage.c shohin.c
[dot1x7830% ./problem3
Apple    単価 150円 (内税)      3 個      450円
Orange   単価 100円 (内税)     2 個      200円
Banana   単価 200円 (内税)     1 個      200円
Book1    単価 500円 (外税)     4 個     2160円
          3010
dot1x7830%
```

図 3: 問題 1-3 実行結果

### 1.3.7 プログラムの解説

[uriage.c] の `printUriageArray` 関数は `URIAGE` 型の配列 `u` を引数とし、戻り値が `shokei(:int)` の関数である。配列 `u` の各要素には `shohin(:SHOHIN)` のインデックスを指定している `code(:int)` プロパティーとそのインデックスの商品の売り上げ数を指定している `num(:int)` プロパティーを持っている構造体が格納されている。配列 `u` の要素の `code(:int)` プロパティーが `-1(:int)` でない限り `for` ループを繰り返す。`for` ループの中では各要素の売り上げ総額を求め、`shokei` に加算していく。売り上げ総額を求める方法は問題 1-2 の `printUriage` 関数と同じである。最後に商品の名前、単価、売り上げ数、売り上げ総額を `print` し、`total_price(:int)` を `return` する。

## 1.4 問題 1-4

### 1.4.1 shohin.h

```
#ifndef C_ALGO_REPORT1_SHOHIN_H
#define C_ALGO_REPORT1_SHOHIN_H

#endif //C_ALGO_REPORT1_SHOHIN_H

#define ZEI (1.08)
typedef struct {
    char* name;
    int tanka;
    int sotozei;
} SHOHIN;
extern SHOHIN shohin[];
void printshohin(SHOHIN s);
```

### 1.4.2 shohin.c

```
#define N 3

#include <stdio.h>
#include "shohin.h"

SHOHIN shohin[] = {{"Apple", 150},
                   {"Orange", 100},
                   {"Banana", 200},
                   {"Book1", 500, 1},
                   {"", 0}};

void printshohin(SHOHIN s) {
    printf("%s\t単価%d円(%s)", s.name, s.tanka, s.sotozei ? "外税" : "内税");
}
```

### 1.4.3 uriage.h

```
#ifndef C_ALGO_REPORT1_URIAGE_H
#define C_ALGO_REPORT1_URIAGE_H

#endif //C_ALGO_REPORT1_URIAGE_H

typedef struct {
    int code;
    int num;
} URIAGE;

int printUriage(URIAGE* q);
int printUriageArray(URIAGE u[]);
int printUriageTrans(URIAGE **p);
```

#### 1.4.4 uriage.c

```

#include <stdio.h>
#include "uriage.h"
#include "shohin.h"

int printUriage(URIAGE *p) {
    char *const product_name = shohin[p->code].name;
    const int item_price = shohin[p->code].tanka;
    const int tax = shohin[p->code].sotozei;
    const int total_sales = p->num;
    const int total_price =
        tax ? (item_price * total_sales) * ZEI : item_price * total_sales;
    printf("%s\t 単価%d円(%s)\t%d 個\t%d円 \n", product_name, item_price,
        tax ? "外税" : "内税", total_sales, total_price);
    return total_price;
}

int printUriageArray(URIAGE u[]) {
    int shokei = 0;
    for (int i = 0; u[i].code != -1; i++) {
        shokei += printUriage(&u[i]);
    }
    return shokei;
}

int printUriageTrans(URIAGE** u) {
    int sub_total = 0;
    int total = 0;
    for (int i = 0; *(u+i) != NULL; i++) {
        sub_total = printUriageArray(*(u+i));
        total += sub_total;
        printf("-----\n");
        sub_total = 0;
    }
    printf("合計 : \t%d円\n", total);
    return total;
}

```

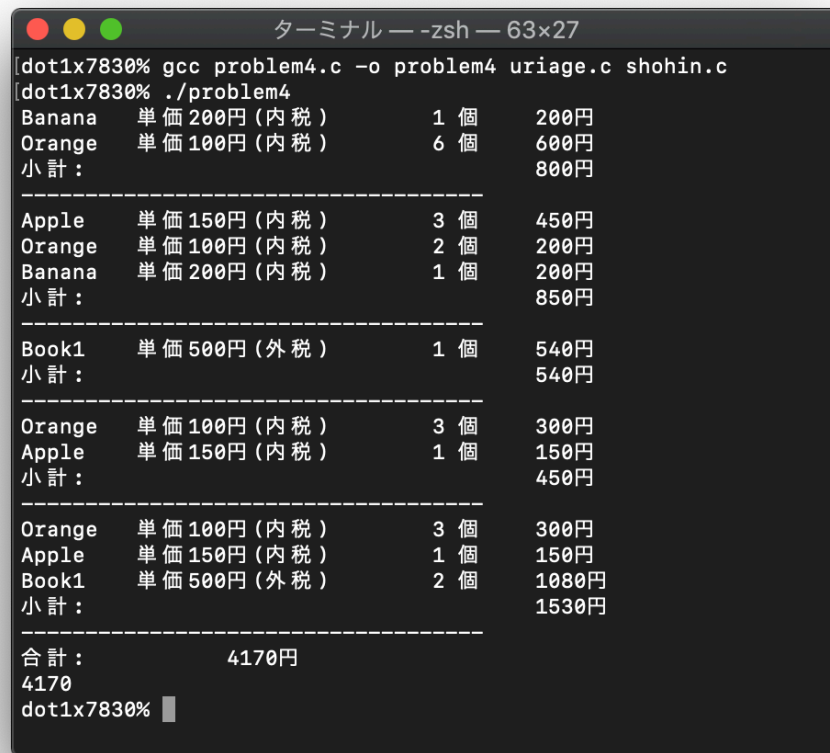
#### 1.4.5 テストプログラム 1-4

```

#include <stdio.h>
#include "shohin.h"
#include "uriage.h"
int main(void){
    URIAGE uriage0[]={ {2,1},{1,6},{-1}};
    URIAGE uriage1[]={ {0,3},{1,2},{2,1},{-1}};
    URIAGE uriage2[]={ {3,1},{-1}};
    URIAGE uriage3[]={ {1,3},{0,1},{-1}};
    URIAGE uriage4[]={ {1,3},{0,1},{3,2},{-1}};
    URIAGE* uriage[]={ uriage0, uriage1, uriage2, uriage3, uriage4, NULL};
    int total;
    total=printUriageTrans(uriage);
    printf("%d\n",total);
    return 0;
}

```

### 1.4.6 実行結果



```

[dot1x7830% gcc problem4.c -o problem4 uriage.c shohin.c ]
[dot1x7830% ./problem4 ]
Banana 単価 200円 (内税)      1 個      200円
Orange 単価 100円 (内税)     6 個      600円
小計:
-----
Apple  単価 150円 (内税)      3 個      450円
Orange 単価 100円 (内税)      2 個      200円
Banana 単価 200円 (内税)      1 個      200円
小計:
-----
Book1   単価 500円 (外税)      1 個      540円
小計:
-----
Orange 単価 100円 (内税)      3 個      300円
Apple  単価 150円 (内税)      1 個      150円
小計:
-----
Orange 単価 100円 (内税)      3 個      300円
Apple  単価 150円 (内税)      1 個      150円
Book1   単価 500円 (外税)      2 個      1000円
小計:
-----
合計:          4170円
4170
dot1x7830% █

```

図 4: 問題 1-4 実行結果

### 1.4.7 プログラムの解説

[uriage.c]printUriageTrans 関数は、URIAGE 型のポインタのポインタ変数を引数とし、戻り値が int の関数である。この関数には main() で uriage0[] ~ uriage4[] の先頭アドレスが格納されている URIAGE 型ポインタ配列 uriage[] が渡される。for 文では、main 関数内の uriage 配列に格納されている配列のアドレスを printUriageArray に渡し、戻り値を sub\_total に格納する。total には、total + sub\_total の値を代入する。この処理を uriage 配列の値が NULL になるまで繰り返す。

## 1.5 考察

今回の課題を解いてレポート書くにあたって構造体を宣言する方法や構造体のプロパティを参照する方法について詳しく理解できた。そしてアロー演算子を使って構造体ポインターからプロパティの値を参照する方法も理解できた。また、ポインタとは変数のアドレスを格納する特殊な変数で、変数のアドレスをメモリのどこかに保存されていてポインタ変数も独自のアドレスを持つことができ、ポインタのポインタから、参照しているポインタがさらに参照している大元の変数へアクセスすることもできるということが分かった。そしてポインタを連鎖は無限にできるが、ポインタのポインタ以降の連鎖はプログラマ自身が参照先がどこなのかを把握しにくくなるため、通常は使用されないということが分かった。