

# 문제해결기법 4주차 과제

202033762 장민호

## HW. Chapter 7.3 -Codes & 작성후기

```
/*  
Homework (30 points)
```

```
Textbook Chapter 7.3  
Programming Exercises 2, 3a and 3b  
Must follow the problem solving steps (including  
the drawing of memory diagrams).  
Chapter 7.3 Exercises
```

2. Write a C function named `change()` that accepts a single-precision number and the addresses of the integer variables named `quarters`, `dimes`, `nickels`, and `pennies`. The function should determine the number of quarters, dimes, nickels, and pennies in the number passed to it and write these values directly into the respective variables declared in its calling function.

3a. Write a function named `secs()` that accepts the time in hours, minutes, and seconds; and determines the total number of seconds in the passed data. Write this function so that the total number of seconds is returned by the function as an integer number.

3b. Repeat Exercise 3a but also pass the address of the variable `totSec` to the function `secs()`. Using this passed address, have `secs()` directly alter the value of `totSec`.

```
*/  
/*
```

문제해결기법 4주차 과제

Chapter 7.3

Exercise 2, 3a, 3b

202033762 장민호

1. 문제설명

- Exercise 2.

`change()` 라는 함수를 만든다. 해당 함수는 임의의 `integer` 타입 숫자 하나와, `integer` 타입 변수 `quarters`, `dimes`, `nickels`, `pennies` 들의 주소를 각각 받는다. `change()` 함수는 임의의 주어진 숫자를 `cent` 로 가정하여 `quarters`, `dimes`, `nickels`, `pennies`로 변환한다. 이때, 함수를 호출하면 주소값에 바로 값을 초기화하도록 한다.

- Exercise 3a.

`secs()` 라는 함수를 만든다. 해당 함수는 `hours`, `minutes`, `seconds`를 인자로 받고, 그 시간을 `seconds`로 변환하여 `int` 타입으로 리턴해준다.

- Exercise 3b.

3a와 동일한 문제인데, `totSec`라는 변수를 추가 선언하고, `secs()`함수에서 `int`값을 리턴하는 방식이 아니라 `totSec`의 주소값을 추가 인자로 받은 뒤 `totSec`에 바로 총 `seconds`를 입력하여 풀이한다.

2. 작성후기

세 문제 모두 Call by reference를 활용하는 법에 대하여 잘 이해할 수 있는 문제였다. 특히 3a, 3b를 통해 Call by value와 Call by reference 간의 차이점에 대해 코드로 느낄 수 있는 경험이 되었다.

```
*/
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <malloc.h>
```

```
void change(int cents, int *quarters, int *dimes, int *nickels, int *pennies)
```

```
{
```

```
    *quarters = 25 * cents;
```

```
    *dimes = 10 * cents;
```

```
    *nickels = 5 * cents;
```

```
    *pennies = cents;
```

```
}
```

```
int secs_3a(int hours, int minutes, int seconds)
```

```
{
```

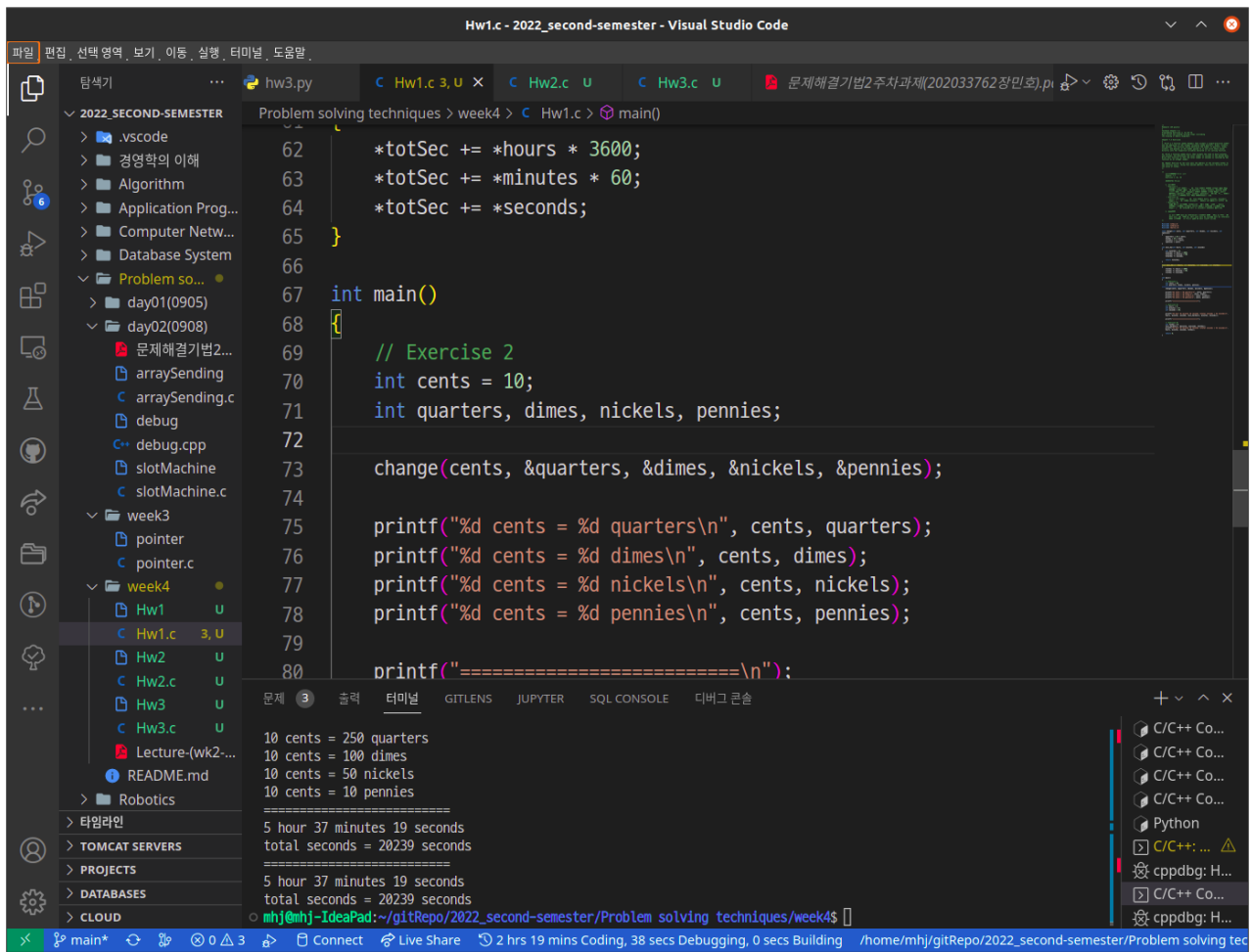
```
    int totalSec = 0;
```

```

        totalSec += hours * 3600;
        totalSec += minutes * 60;
        totalSec += seconds;
        return totalSec;
    }
    void secs_3b(int *hours, int *minutes, int *seconds, int *totSec)
    {
        *totSec += *hours * 3600;
        *totSec += *minutes * 60;
        *totSec += *seconds;
    }
    int main()
    {
        // Exercise 2
        int cents = 10;
        int quarters, dimes, nickels, pennies;
        change(cents, &quarters, &dimes, &nickels, &pennies);
        printf("%d cents = %d quarters\n", cents, quarters);
        printf("%d cents = %d dimes\n", cents, dimes);
        printf("%d cents = %d nickels\n", cents, nickels);
        printf("%d cents = %d pennies\n", cents, pennies);
        printf("=====\n");
        // Exercise 3a
        int hours = 5;
        int minutes = 37;
        int seconds = 19;
        printf("%d hour %d minutes %d seconds \ntotal seconds = %d seconds\n", hours, minutes,
seconds, secs_3a(hours, minutes, seconds));
        printf("=====\n");
        // Exercise 3b
        int totSec = 0;
        secs_3b(&hours, &minutes, &seconds, &totSec);
        printf("%d hour %d minutes %d seconds \ntotal seconds = %d seconds\n", hours, minutes,
seconds, totSec);
        return 0;
    }

```

- 실행결과



## HW. Chapter 9.4 -Codes & 작성 후기

/\*

Textbook Chapter 9.4

●

Programming Exercise #2

Write a C function named `liquid()` that is to accept an integer number and the addresses of the variables gallons, quarts, pints, and cups. The passed integer represents the total number of cups, and the function is to determine the number of gallons, quarts, pints, and cups in the passed value. Using the passed addresses, the function should directly alter the respective variables in the calling function. Use the relationships of 2 cups to a pint, 4 cups to a quart, and 16 cups to a gallon.

\*/

/\*

문제해결기법 4주차 과제

Chapter 9.4

Exercise 2

202033762 장민호

1. 문제설명

- Exercise 2.

`liquid()` 라는 함수를 만든다. 해당 함수는 총 5개의 인자를 갖는데, 하나는 `int` 타입 변수이며 나머지 4개는 gallons, quarts, pints, cups 변수의 주소값이다. `int` 타입 변수는 cup의 총 개수를 뜻하며 `liquid` 함수는 cup의 총 개수를 각각 몇 gallon, quart, pint, cup 인지 변환하면 된다. 이때 계산 시 주어진 주소값을 이용해 즉시 값을 반영하도록 한다. 16 cups = 1 gallon, 4 cups = 1 quart, 2 cups = 1 pint 임을 활용하여 계산하자.

2. 작성 후기

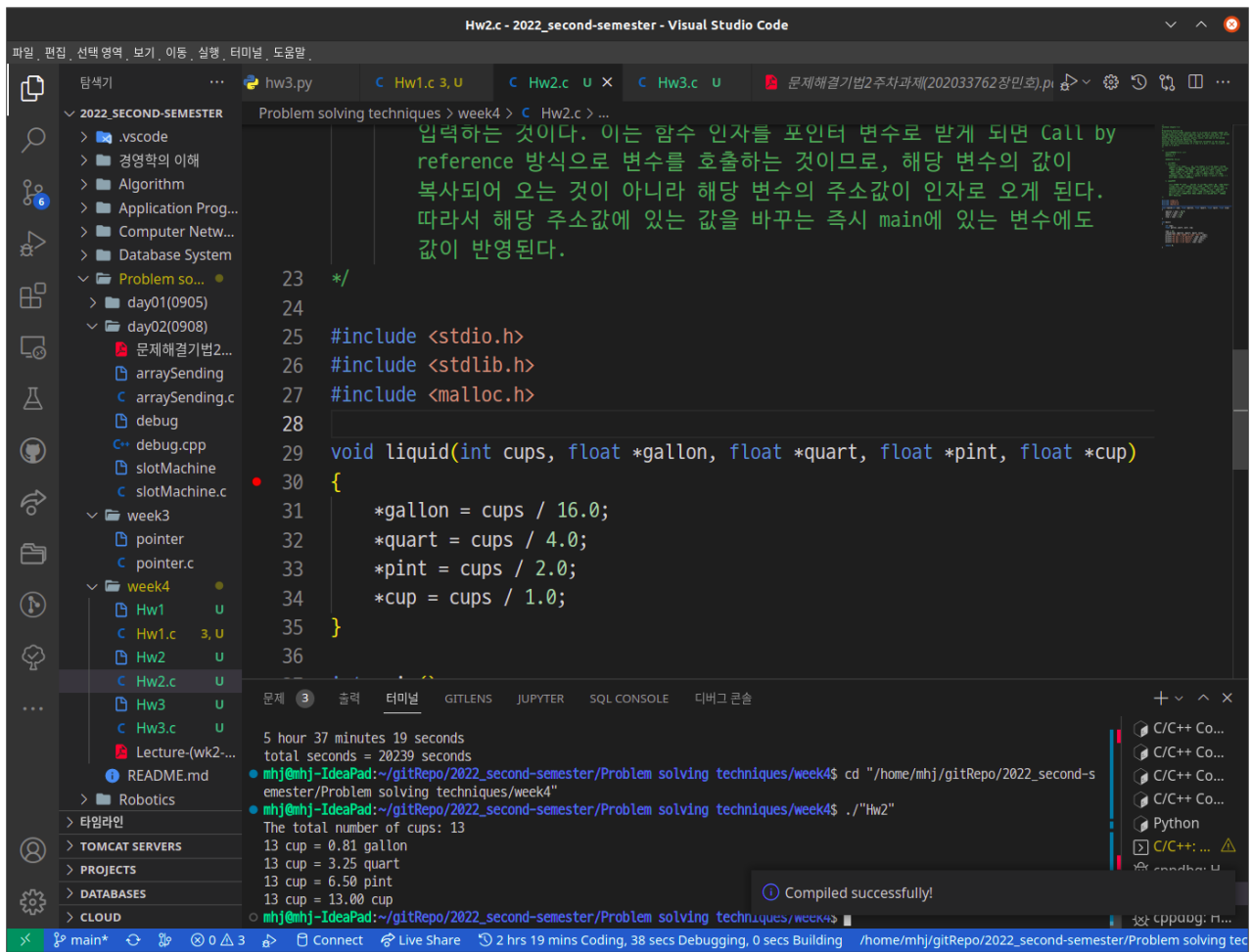
문제 해결에 큰 어려움은 없었다. 이 문제에서 중요한 점은 함수 인자로 주어진 cups 변수를 계산을 통해 포인터 변수에 즉시 계산된 값을 입력하는 것이다. 이는 함수 인자를 포인터 변수로 받게 되면 Call by reference 방식으로 변수를 호출하는 것이므로, 해당 변수의 값이 복사되어 오는 것이 아니라 해당 변수의 주소값이 인자로 오게 된다. 따라서 해당 주소값에 있는 값을 바꾸는 즉시 main에 있는 변수에도 값이 반영된다.

```

*/
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
void liquid(int cups, float *gallon, float *quart, float *pint, float *cup)
{
    *gallon = cups / 16.0;
    *quart = cups / 4.0;
    *pint = cups / 2.0;
    *cup = cups / 1.0;
}
int main()
{
    int cups;
    float gallon, quart, pint, cup;
    cups = 13;
    liquid(cups, &gallon, &quart, &pint, &cup);
    printf("The total number of cups: %d\n", cups);
    printf("%d cup = %.2f gallon\n", cups, gallon);
    printf("%d cup = %.2f quart\n", cups, quart);
    printf("%d cup = %.2f pint\n", cups, pint);
    printf("%d cup = %.2f cup\n", cups, cup);
    return 0;
}

```

## - 실행결과



## HW. Chapter 9.4

### -Codes & 작성 후기

```

/*
Homework (20 points)
Textbook Chapter 8.3
Programming Exercises 2 and 4
Must follow the problem solving steps (including
the drawing of memory diagrams).
2. Write a program that has a declaration in main() to store the following numbers into an array
named rates : 6.5, 8.2, 8.5, 8.3, 8.6, 9.4, 9.6, 9.8, 10.0. There should be a function call to
show() that accepts the rates array as a parameter named rates and then displays the numbers in
the array.
4. Write a program that declares three one-dimensional arrays named price, quantity, and amount.
Each array should be declared in main() and should be capable of holding 10 double-precision
numbers. The numbers that should be stored in price are 10.62, 14.89, 13.21, 16.55, 18.62, 9.47,
6.58, 18.32, 12.15, 3.98. The numbers that should be stored in quantity are 4, 8.5, 6, 8.35, 9,
15.3, 3, 5.4, 2.9, 4.8. Your program should pass these three arrays to a function called
extend(), which should calculate the elements in the amount array as the product of the
equivalent elements in the price and quantity arrays (for example, amount[1] = price[1] *
quantity[1]). After extend() has put values in the amount array, the values in the array should
be displayed from within main().
*/
/*
문제해결기법 4주차 과제
Chapter 8.3
Exercise 2, 4
202033762 장민호
1. 문제설명
- Exercise 2.

```

rates라는 이름의 배열을 main 함수 내부에 선언하고, 해당 배열에 주어진 숫자들을 넣어 초기화한다. 이후, rates 배열을 인자값으로 갖는 show() 라는 함수를 만든다. show() 함수는 인자값으로 받은 배열 내부에 있는 숫자들을 나열하여 보여주면 된다.

#### - Exercise 4.

3개의 1차원 배열을 선언한다. 각 배열의 이름은 price, quantity, 그리고 amount 이다. 각 배열은 main 함수 내부에서 선언되어야 하며 10개의 double 타입 숫자들을 받을 수 있도록 해야 한다. price와 quantity 배열에는 주어진 일련의 숫자들을 넣어 초기화 시키면 된다. 내가 할 일은 extend()라는 함수를 만드는 것인데, 그 함수는 세 개의 배열을 인자값으로 받아야 한다. 함수 내부에서는 amount[1] = price[1] \* quantity[1] 과 같은 연산을 수행하여 amount 배열에 데이터를 입력한다. extend() 함수가 amount 배열에 데이터를 다 넣었으면 해당 데이터를 main 함수 내부에서 출력하도록 한다.

#### 2. 작성후기

문제가 단순하여 해결에 큰 어려움이 없었다. 이 문제에서 중요한 점은 배열의 이름은 포인터이므로, 함수에서 배열을 받을 때에는 포인터로 받아도 된다는 것이며 실제로 Exercise 4에서 해당 방식으로 문제를 해결하였다.

```
*/
#include <stdio.h>
#include <malloc.h>
#define MAX 10
// Exercise 2
void show(float rates[])
{
    for (int i = 0; i < 9; i++)
    {
        printf("rates[%d] = %.1f\n", i, rates[i]);
    }
}
// Exercise 4
void extend(double *price, double *quantity, double *amount)
{
    for (int i = 0; i < MAX; i++)
    {
        amount[i] = price[i] * quantity[i];
    }
}
int main()
{
    printf("==== Exercise 2 =====\n");
    // ===== Exercise 2 Start =====
    float rates[9] = {6.5, 8.2, 8.5, 8.3, 8.6, 9.4, 9.6, 9.8, 10.0};
    show(rates);
    // ===== Exercise 2 End =====
    printf("==== Exercise 4 =====\n");
    // ===== Exercise 4 Start =====
    double price[MAX] = {10.62, 14.89, 13.21, 16.55, 18.62, 9.47, 6.58, 18.32, 12.15, 3.98};
    double quantity[MAX] = {4, 8.5, 6, 8.35, 9, 15.3, 3, 5.4, 2.9, 4.8};
    double amount[MAX];
    extend(price, quantity, amount);
    for (int i = 0; i < MAX; i++)
    {
        printf("amount[%d] = %.21f\n", i, amount[i]);
    }
    // ===== Exercise 4 End =====
    return 0;
}
```

## - 실행결과

Hw3.c - 2022\_second-semester - Visual Studio Code

파일, 편집, 선택 영역, 보기, 이동, 실행, 터미널, 도움말

탐색기

- 2022\_SECOND-SEMESTER
  - .vscode
  - 경영학의 이해
  - Algorithm
  - Application Prog...
  - Computer Netw...
  - Database System
  - Problem so...
    - day01(0905)
    - day02(0908)
      - 문제해결기법2...
        - arraySending
        - arraySending.c
        - debug
        - debug.cpp
        - slotMachine
        - slotMachine.c
      - week3
        - pointer
        - pointer.c
      - week4
        - Hw1
        - Hw1.c
        - Hw2
        - Hw2.c
        - Hw3
        - Hw3.c
      - Lecture-(wk2-...
      - README.md
    - Robotics
    - 타임라인
    - TOMCAT SERVERS
    - PROJECTS
    - DATABASES
    - CLOUD

hw3.py

C Hw1.c 3, U

C Hw2.c U

C Hw3.c U X

문제해결기법2 주차과제(202033762장민호).py

Problem solving techniques > week4 > C Hw3.c > ...

24

25

내부에 있는 숫자들을 나열하여 보여주면 된다.

- Exercise 4.

3개의 1차원 배열을 선언한다. 각 배열의 이름은 price, quantity, 그리고 amount 이다. 각 배열은 main 함수 내부에서 선언되어야 하며 10개의 double 타입 숫자들을 받을 수 있도록 해야 한다. price와 quantity 배열에는 주어진 일련의 숫자들을 넣어 초기화 시키면 된다. 내가 할 일은 extend()라는 함수를 만드는 것인데, 그 함수는 세 개의 배열을 인자값으로 받아야 한다. [함수 내부에서는 amount[1] = price[1] \* quantity[1] 과 같은 연산을 수행하여 amount 배열에 데이터를 입력한다. extend() 함수가 amount 배열에 데이터를 다 넣었으면 해당

문제 3 출력 터미널 GITLENS JUPYTER SQL CONSOLE 디버그 콘솔

• mhj@mhj-IdeaPad:~/gitRepo/2022\_second-semester/Problem solving techniques/week4\$ cd "/home/mhj/gitRepo/2022\_second-semester/Problem solving techniques/week4"  
 • mhj@mhj-IdeaPad:~/gitRepo/2022\_second-semester/Problem solving techniques/week4\$ ./"Hw3"  
 ===== Exercise 2 =====  
 rates[0] = 6.5  
 rates[1] = 8.2  
 rates[2] = 8.5  
 rates[3] = 8.3  
 rates[4] = 8.6  
 rates[5] = 9.4  
 rates[6] = 9.6  
 rates[7] = 9.8  
 rates[8] = 10.0  
 ===== Exercise 4 =====  
 amount[0] = 42.48  
 amount[1] = 126.56  
 amount[2] = 79.26  
 amount[3] = 138.19  
 amount[4] = 167.58  
 amount[5] = 144.89  
 amount[6] = 19.74  
 amount[7] = 98.93  
 amount[8] = 35.23  
 amount[9] = 19.10

Compiled successfully!

main\* 0 3 Connect Live Share 2 hrs 19 mins Coding, 38 secs Debugging, 0 secs Building /home/mhj/gitRepo/2022\_second-semester/Problem solving tec