

C 프로그래밍 및 실습

기말고사 프로젝트

< 가계부 + 소득, 지출 분석 프로그램 >

전남대학교
소프트웨어공학과
175768 최재원

소득, 지출 분석 프로젝트

1. 만들고자 하는 소프트웨어 개요

1-1) 만들고자 하는 소프트웨어 요약

돈을 관리하는 것은 그 소득, 지출을 기록하고 현황을 확인하며 분석하는 것에서 시작된다. 사용자가 입력한 소득, 지출을 분석하고 사용자에게 어떠한 금전적 행동을 해야하는지 알려주는 프로그램이다. 사용자는 2020년도의 소득, 지출을 입력하여 이에 대한 분석 및 현황을 확인할 수 있다.

1-2) 만들고자 하는 소프트웨어 주요 기능

• 최초 설명 함수 < aboutProgram >

- 프로그램을 시작하면 프로그램의 전체적인 기능, 입력 데이터, 데이터 저장, 데이터 처리, 그리고 데이터 출력에 대한 간단한 설명문이 나오며 이를 통해 사용자는 프로그램의 목적과 기능을 알 수 있다.

• 메뉴 출력 함수 < showMenu >

- 사용자가 선택할 수 있는 메뉴 옵션들을 출력하여 보여준다.

• 메뉴 실행 함수 < performMenu >

- 사용자가 선택한 실행 옵션에 관한 함수를 호출하는 함수이다. 1 ~ 5번까지는 프로그램의 기능을 실행하기 위한 case이고 6번은 프로그램 종료 case이다. 이에 해당하는 메뉴를 선택하지 않으면 재선택하라는 문구가 출력된다.

• 소득 입력 함수 < income >

- 사용자가 입력한 월/일에 3가지의 내역(월급, 용돈, 기타) 중 하나를 선택하고 이에 대한 소득 금액을 입력할 수 있다. 한 번의 입력이 종료되면 동일한 달의 추가 소득을 입력하거나 다른 달을 선택하여 소득을 입력할 수 있다. 만약 더 이상의 소득 입력이 없으면 '소득 입력 종료'를 선택하여 메뉴 화면으로 돌아갈 수 있다.
- 해당 함수에는 월별 마지막 날을 반환하는 함수를 가지고 있다.
- 월, 일, 금액, 내역 입력에서 잘못된 입력이 있으면 MENU 화면으로 넘어간다.

• 지출 입력 함수 < spend >

- 사용자가 입력한 월/일에 9가지의 내역(식비, 식비외, 주거비, 의류비, 교통비, 교육비, 미용비, 의료비, 기타) 중 하나를 선택하고 이에 대한 지출 금액을 입력할 수 있다. 한 번의 입력이 종료되면 동일한 달의 추가 지출을 입력하거나 다른 달을 선택하여 지출을 입력할 수 있다. 만약 더 이상의 지출 입력이 없으면 '지출 입력 종료'를 선택하여 메뉴 화면으로 돌아갈 수 있다.
- 해당 함수에는 월별 마지막 날을 반환하는 함수를 가지고 있다.
- 월, 일, 금액, 내역 입력에서 잘못된 입력이 있으면 MENU 화면으로 넘어간다.

• 소득 출력 함수 < showIncome >

- 사용자가 월 소득을 확인하고 싶은 달을 입력하면 해당 월의 날짜, 내역별 소득을 출력한다. 적절하지 못한 달을 입력하면 MENU 화면으로 넘어간다.

• 지출 출력 함수 < showSpend >

- 사용자가 월 지출을 확인하고 싶은 달을 입력하면 해당 월의 날짜, 내역별 지출을 출력한다. 적절하지 못한 달을 입력하면 MENU 화면으로 넘어간다.

• 소득, 지출 분석 함수 < analyze >

- 사용자는 해당 함수를 통해 지금까지 입력한 소득, 지출의 종합 현황 및 분석을 확인할 수 있다. 함수는 사용자가 선택한 달의 소득, 지출 종합 현황을 출력하고 이후에 소득과 지출을 비교하여 흑자일 때의 경우와 적자일 때의 경우로 나누어서 설명한다. 소득과 지출의 차이 정도에 따라 흑자 또는 적자 단계가 적용이 되며 이에 따른 추천 행동 요령을 출력한다.

• 월말일 함수 < lastDay >

- 넘겨받은 월의 마지막일 반환한다. (2020년 기준으로 2월은 29일을 반환한다.)

2. 아이디어 개요

2-1) 아이디어의 구체성 및 우수성

☞ 아이디어의 우수성 및 이를 실현하기 위한 방법에 대해서 구체적으로 기재하세요. (그림 또는 표 활용가능)

1. 가계부로서의 기능

- 해당 프로그램에서는 소득과 지출의 분석을 위해 우선적으로 이를 입력하여야 한다. 이 때 이를 저장하는 배열이 main함수에 선언되어 있으며 이를 함수의 매개변수를 통해 전달되어 값을 저장하기 때문에 결국에는 모든 입력이 두 개의 3차원 배열 (소득 배열, 지출 배열)에 저장되어진다. 즉, 프로그램을 종료하지 않는다면 (스마트폰 어플리케이션에서 이를 구현하면 프로그램의 종료는 어플리케이션의 삭제라고 할 수 있다.) 소득과 지출에 대한 데이터는 유지된다.

2. 소득, 지출 입력의 자동화

- 이 프로그램에서는 소득과 지출을 사용자가 직접 입력하여야한다는 단점이 있다. 물론 가계부로서의 기능 또한 사용하고 있다면 소득과 지출이 발생할 때마다 입력을 하여 크게 상관하지 않아도 되는 단점이지만 분석으로서의 기능만을 사용하고자하는 사용자에게는 모든 소득과 지출을 입력하는 것은 시간이 오래 걸려서 불편할 것이다. 이를 보완하기 위해서는 사용자가 사용하는 카드사와의 연동을 통해 소득과 지출이 자동으로 입력되게 할 수 있다.

3. 프로그램의 역할

- 해당 프로그램은 사용자가 원활한 소비 체계를 계획하고 건강한 경제 생활을 하는데에 목적을 둔다. 이 프로그램에서는 단순히 소득과 지출의 금액만을 다루는 것이 아닌 날짜별 그리고 내역별 내용을 다루기 때문에 사용자는 이를 통해 자신의 경제적 상태를 실시간으로 확인하고 이에 따른 원활한 경제 활동을 할 수 있다.

2-2) 아이디어의 독창성

☞ 제안하는 아이디어의 독창성 및 차별화된 강점은 무엇입니까?

- 사실 가계부 기능을 행하는 프로그램들은 굉장히 다양합니다. 저 또한 가계부를 사용하고 있기 때문에 비슷한 기능을 하는 프로그램들을 많이 접해보았습니다. 하지만 이러한 프로그램들 모두 소비, 지출의 현황만을 보여주거나 통장 잔고를 보여주는 추가 기능만을 가지고 있어서 이후의 경제적 활동을 어떻게 해야하는지에 대한 대안을 주는 프로그램은 없었습니다. 이 프로그램을 만든 이유는 이러한 기능까지 가지는 가계부 / 경제 분석 프로그램을 만들고 싶었기 때문입니다. 그리고 이를 부족하지만 틀을 짜는 정도로 구현을 하였고 이후에 좀 더 세밀한 소득, 지출 분석과 이에 따른 행동 요령을 추가하여 더 발전된 프로그램을 만들 수 있을 것 같습니다. 즉, 다른 가계부 프로그램들과의 차별성이 이 아이디어의 독창성이라고 생각합니다.

3. 설계(알고리즘)

DEFINE. 상수 정의

- 이후의 알고리즘에서 상수를 다음과 같이 정의한다.

MONTH_V = 12 / DAY_V = 31 / INCOME_V = 3 / SPEND_V = 9

ARRAY. 배열 정의

- 이후의 알고리즘에서 사용되는 배열은 다음과 같이 정의한다.

int yearIncome{MONTH_V}[DAY_V][INCOME_V] : 월/일/내역별 소득을 저장하는 3차원 배열

int yearSpend{MONTH_V}[DAY_V][SPEND_V] : 월/일/내역별 지출을 저장하는 3차원 배열

```
char *contentI[INCOME_V] = {"월급", "용돈", "기타"} : 소득 내역을 나타내는 1차원 배열
```

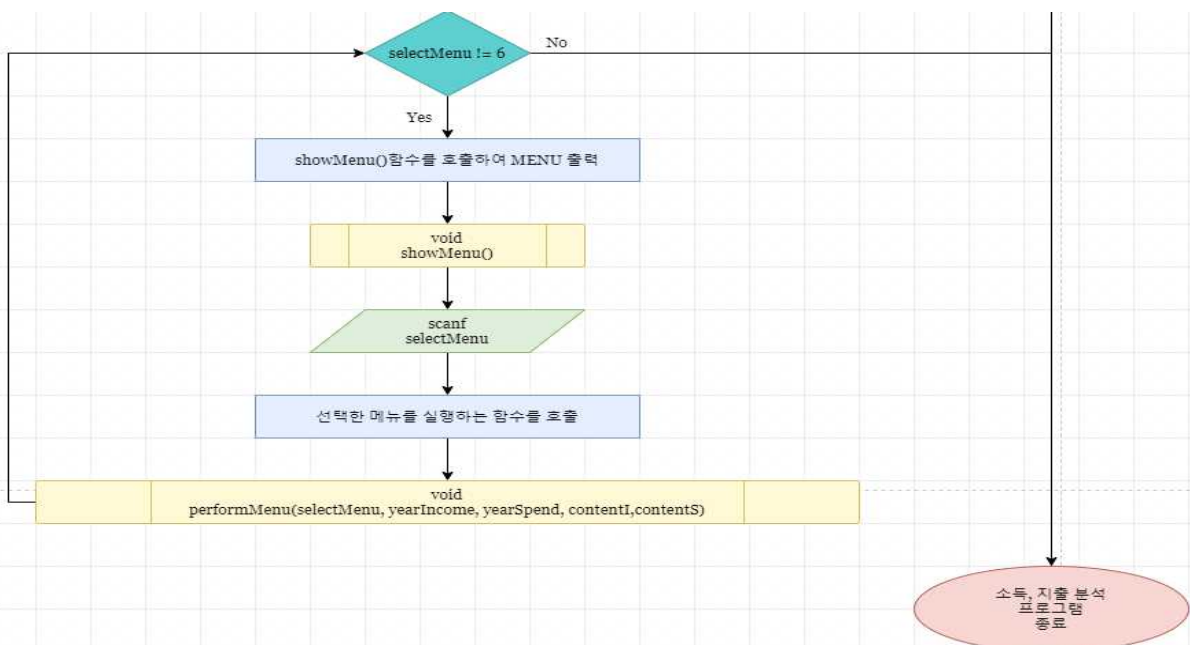
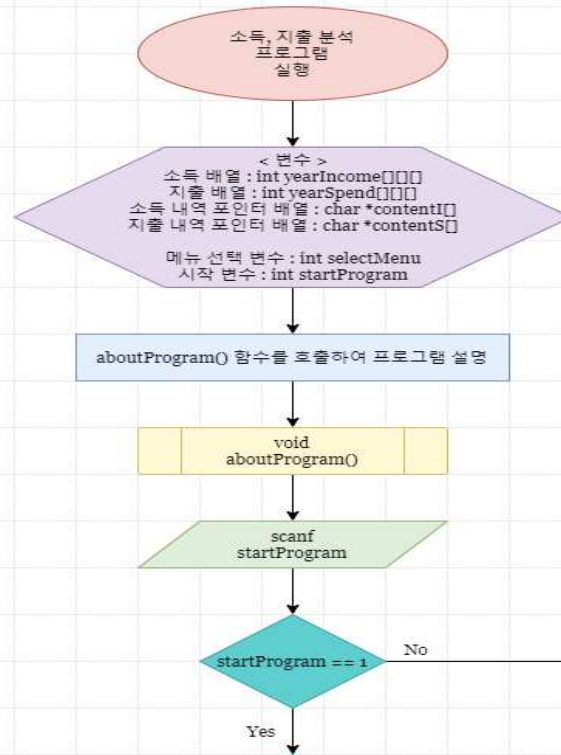
```
char *contentS[SPEND_V] ={"식비", "식비외", "주거비",
```

“의류비”, “교통비”, “교육비”,

“미용비”, 의료비“, ”기타“} : 지출 내역을 나타내는 1차원 배열

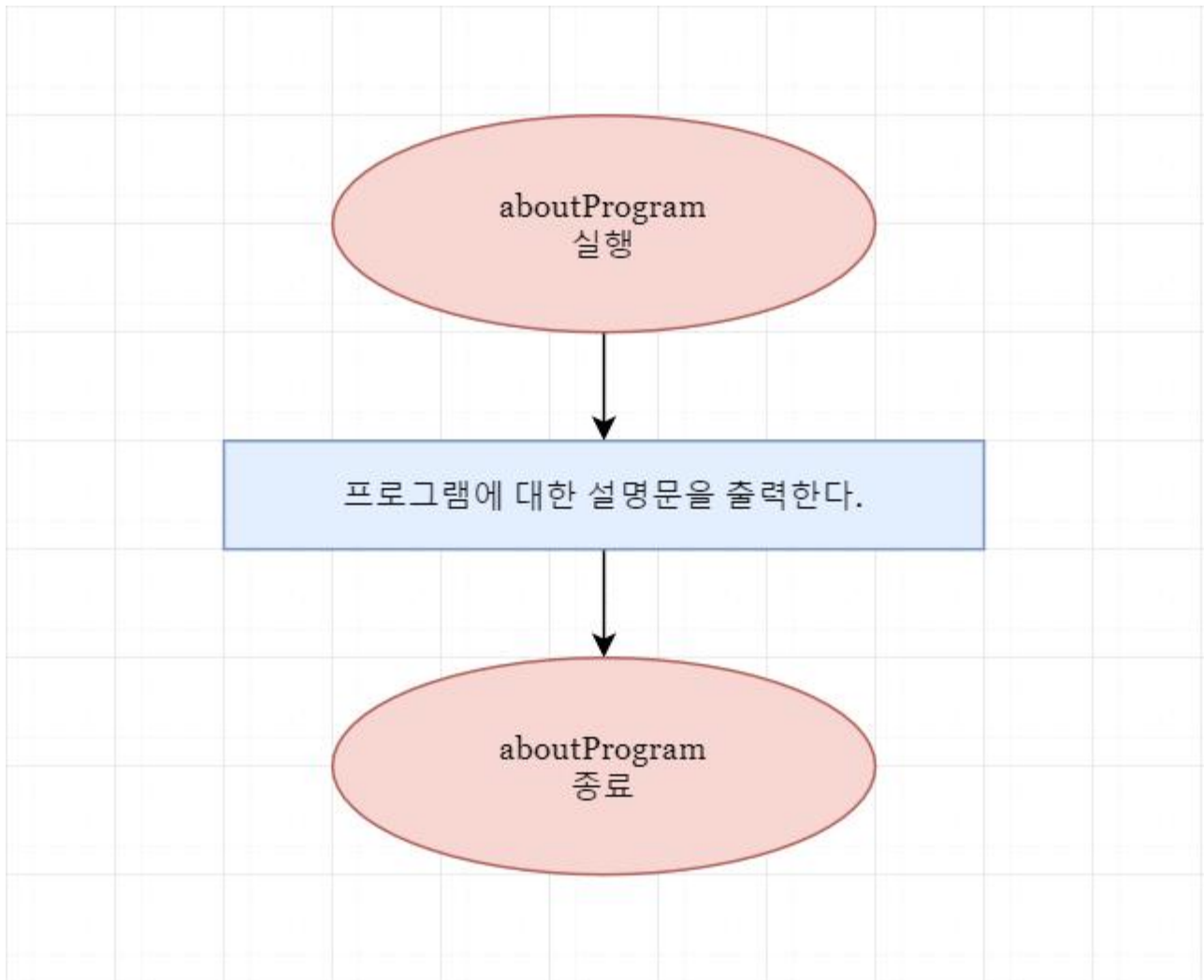
A. main함수 알고리즘

- 입력 : 프로그램의 실행 여부 / 옵션 선택
- 출력 : 실행하는 프로그램의 기능
- 일처리 순서 : 프로그램 설명 → 실행 여부 → 메뉴 화면 출력 → 메뉴 선택 → 메뉴 구현 → 종료



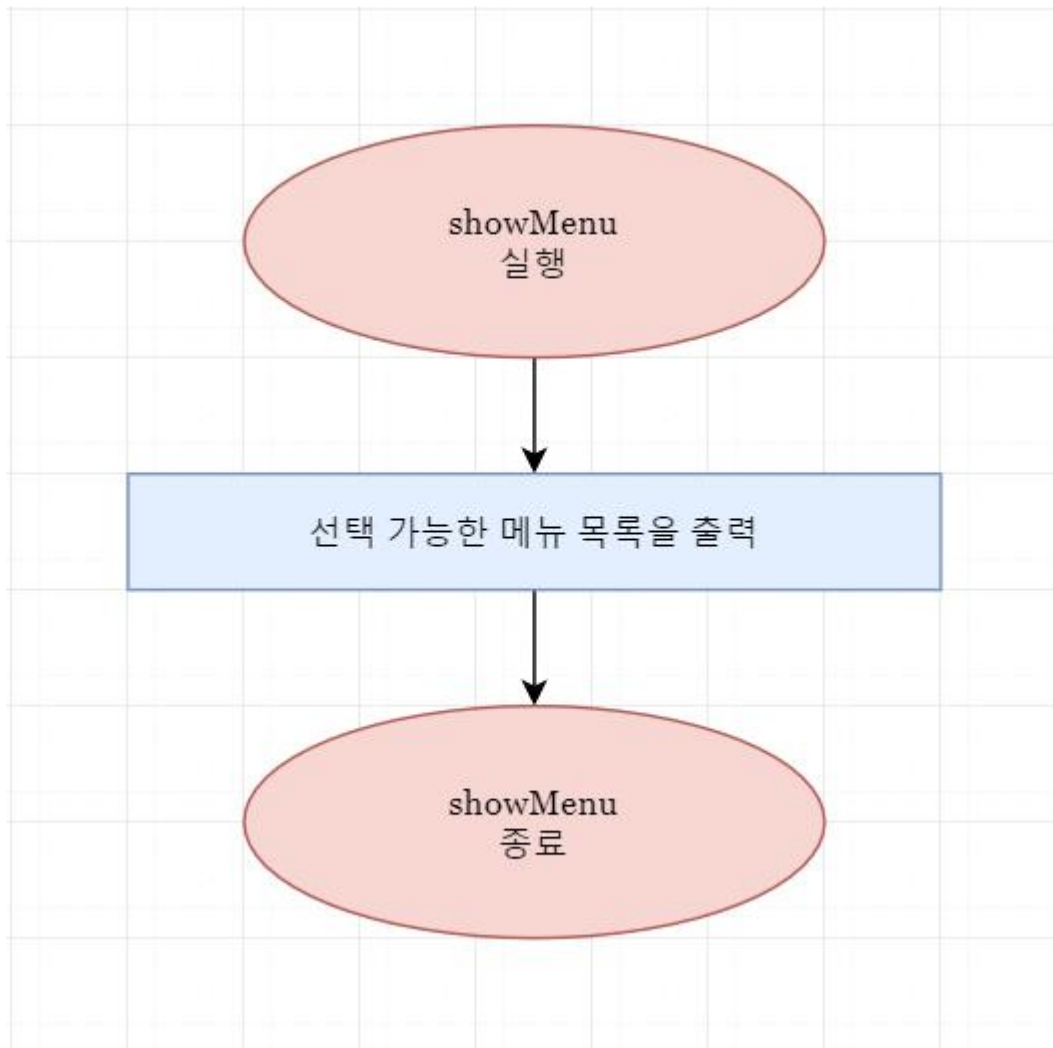
B. aboutProgram 함수 알고리즘

- 출력 : 프로그램에 대한 설명문
- 일처리 순서 : 프로그램 설명문 출력



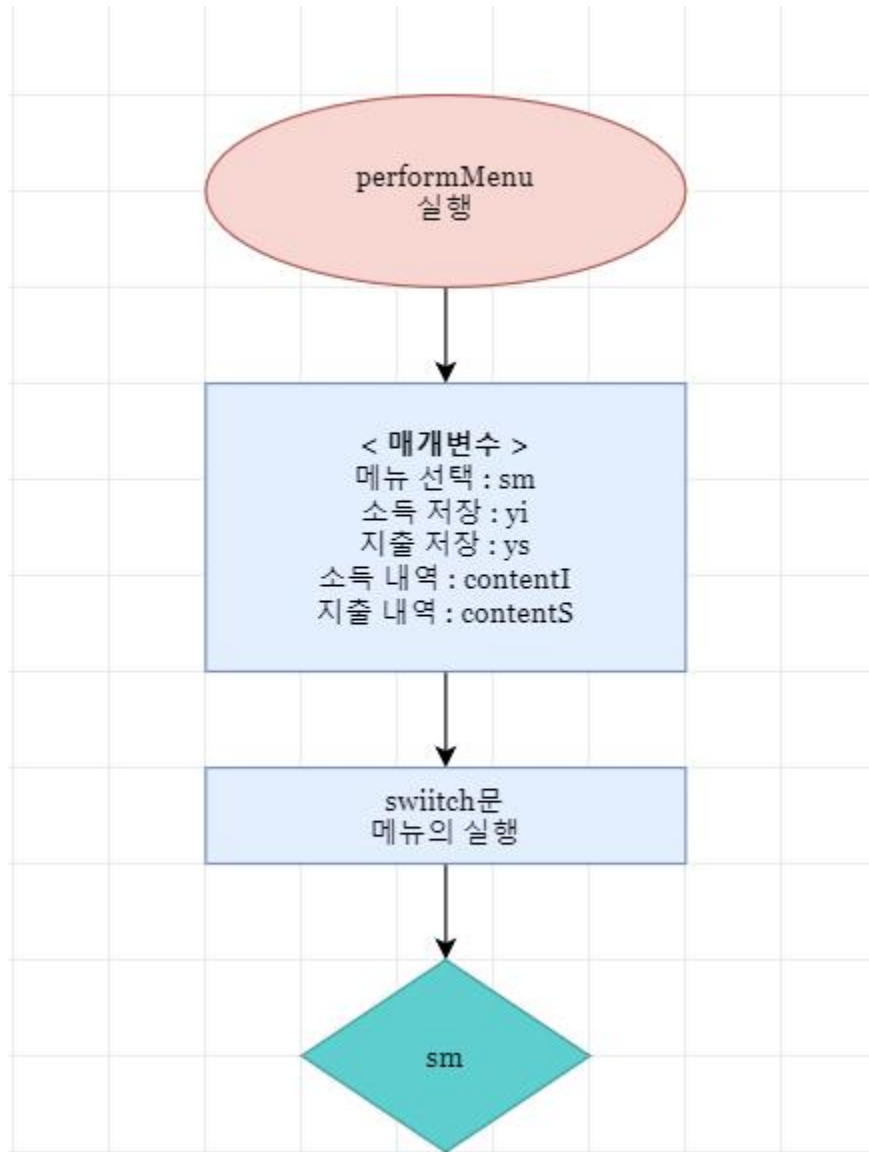
C. main함수 알고리즘

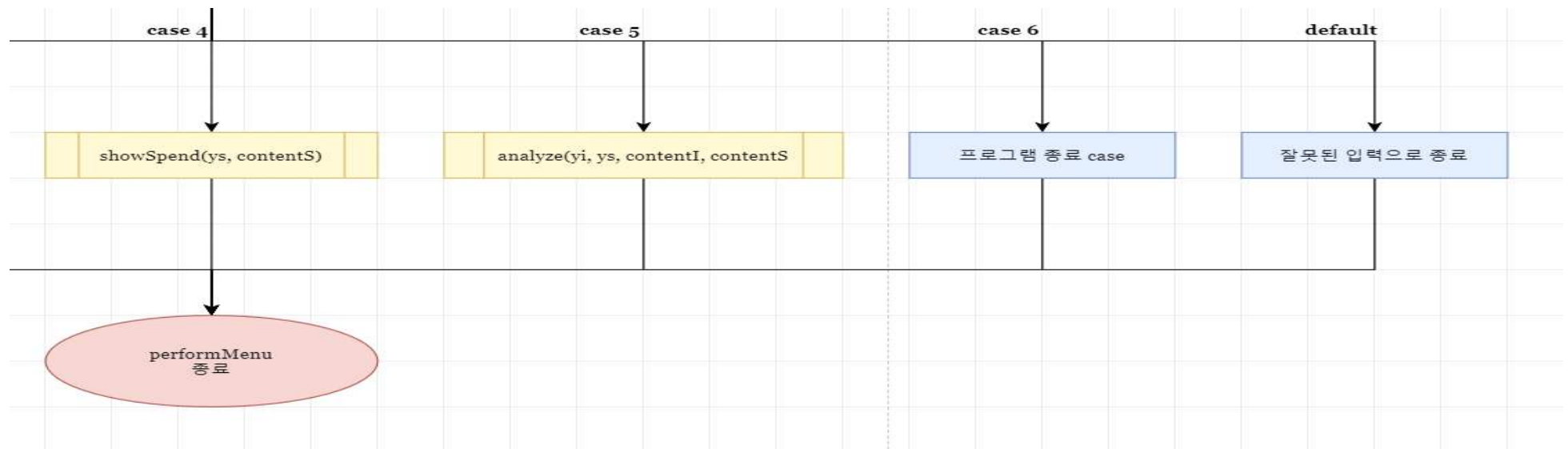
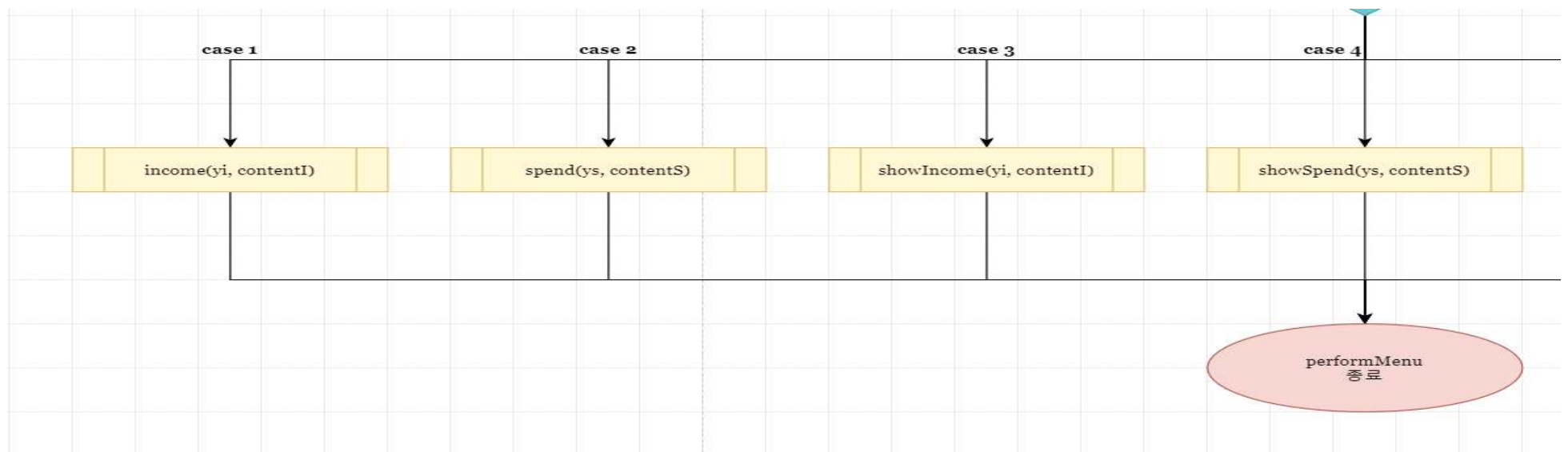
- 출력 : 메뉴 항목 출력
- 일처리 순서 : 메뉴 항목 출력



D. performMenu 함수 알고리즘

- 출력 : switch문을 통해 선택된 함수 또는 실행문을 실행
- 일처리 순서 : switch문 → switch문의 값에 따른 case 실행

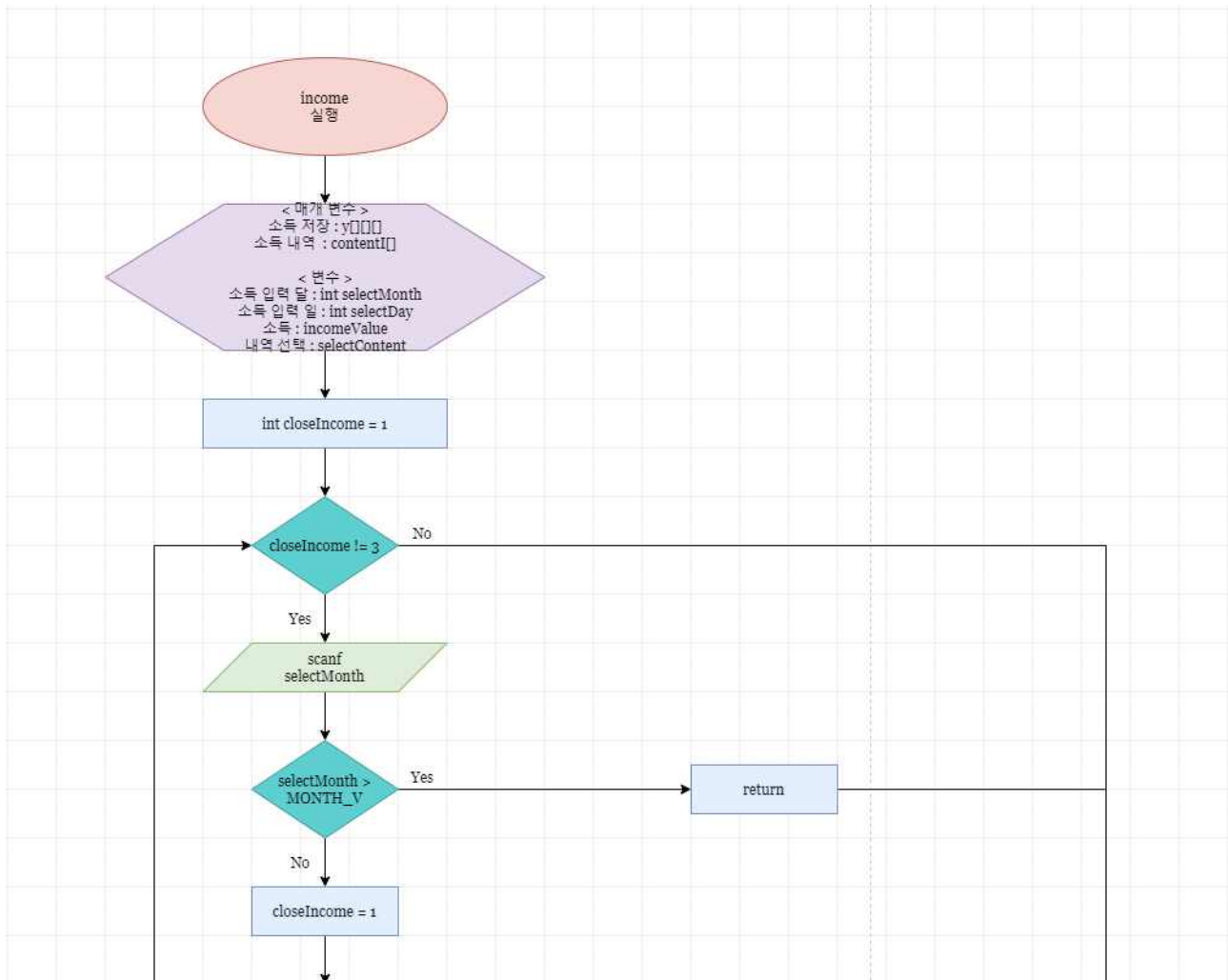


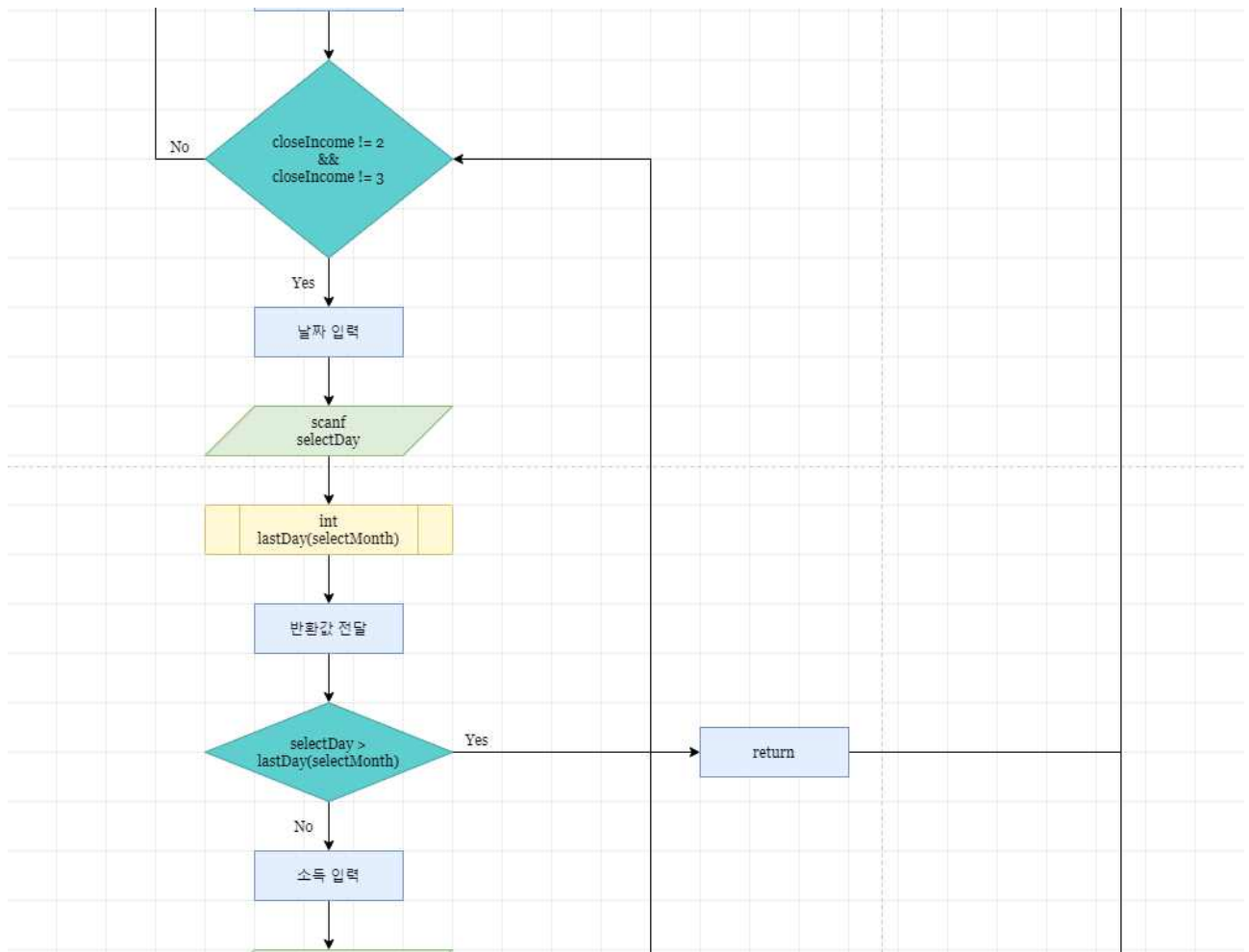


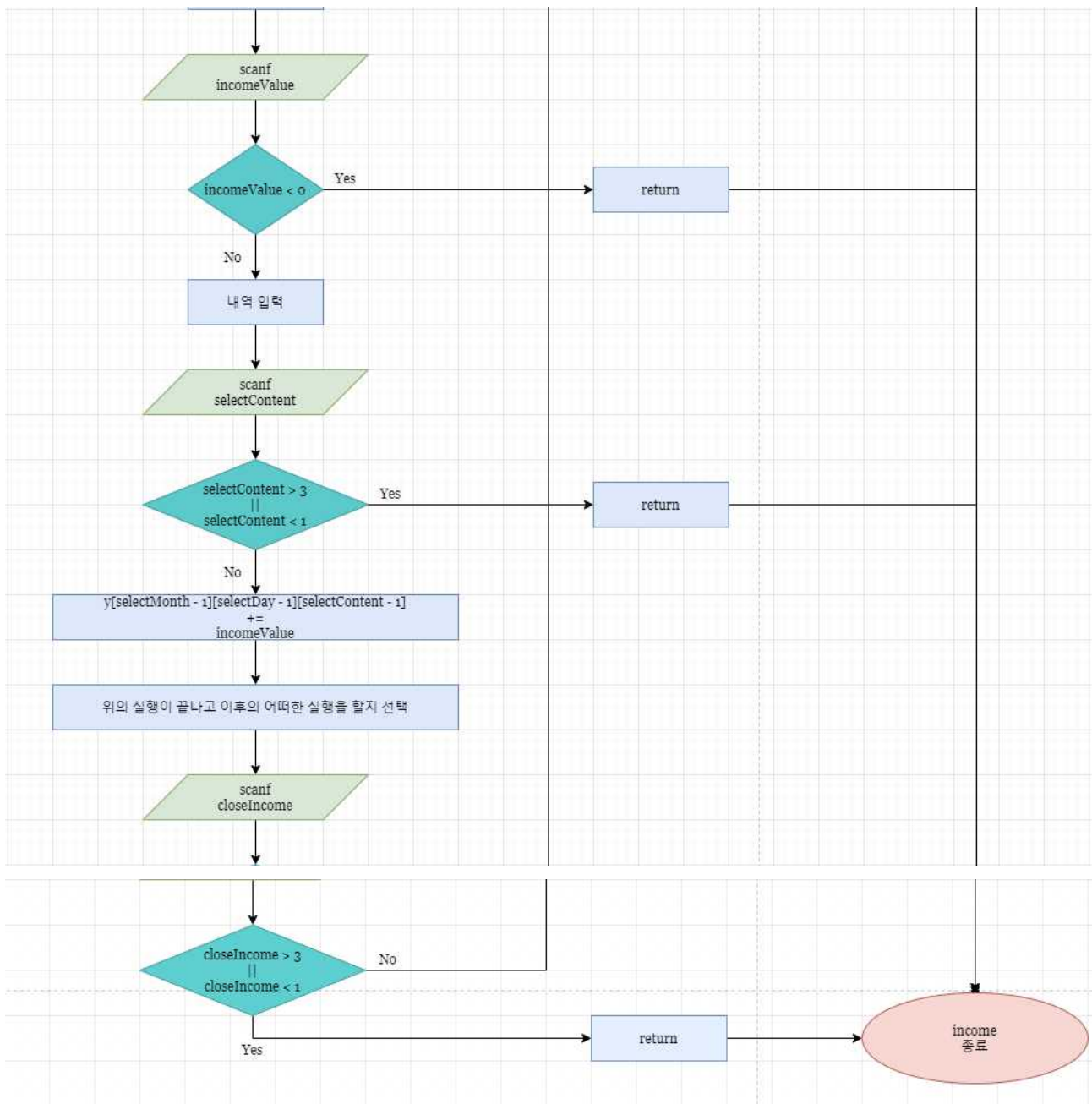
E. income 함수 알고리즘

- 입력 : 소득 저장 월 / 소득 저장 일 / 소득 저장 내역 / 소득 금액 / 차후 실행 선택
- 저장 : 사용자가 입력한 정보를 바탕으로 소득을 소득 배열에 저장한다.
- 일처리 순서 : 소득 입력 달 입력 → 날짜 입력 → 소득 입력 → 내역 입력 → 데이터 저장 → 차후 실행 선택

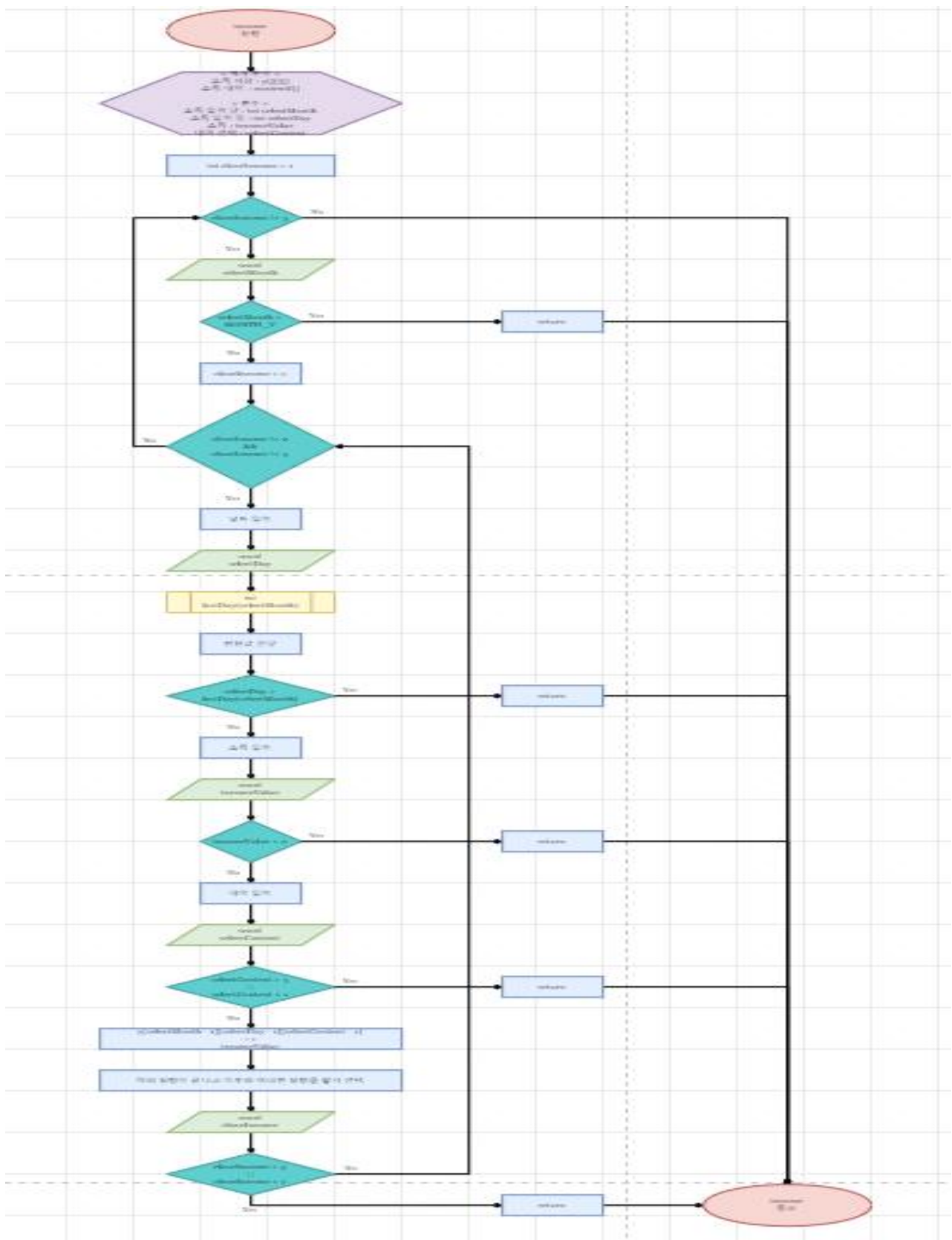
income 함수의 분할 캡처 알고리즘







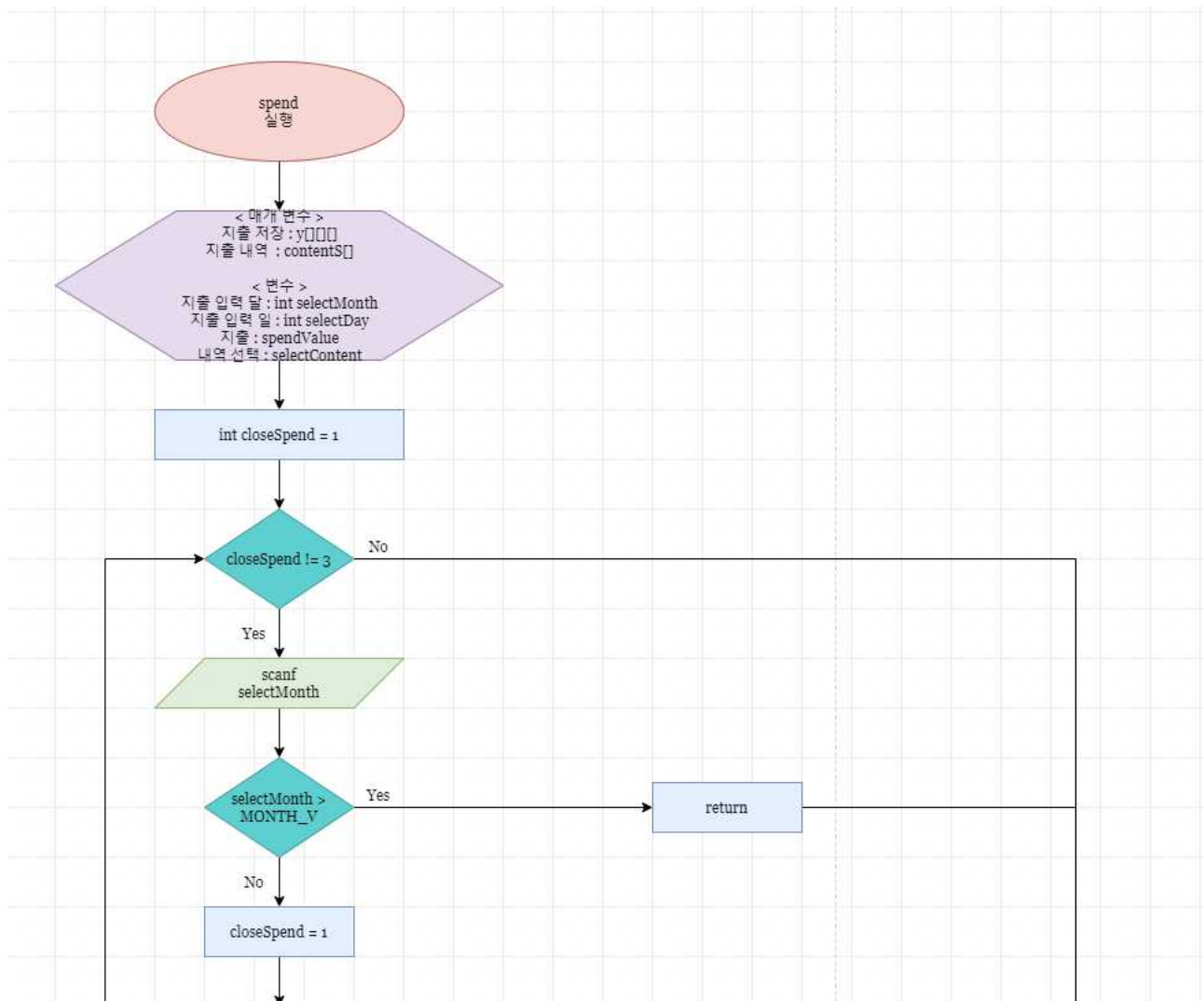
income 함수의 전체 캡처 알고리즘

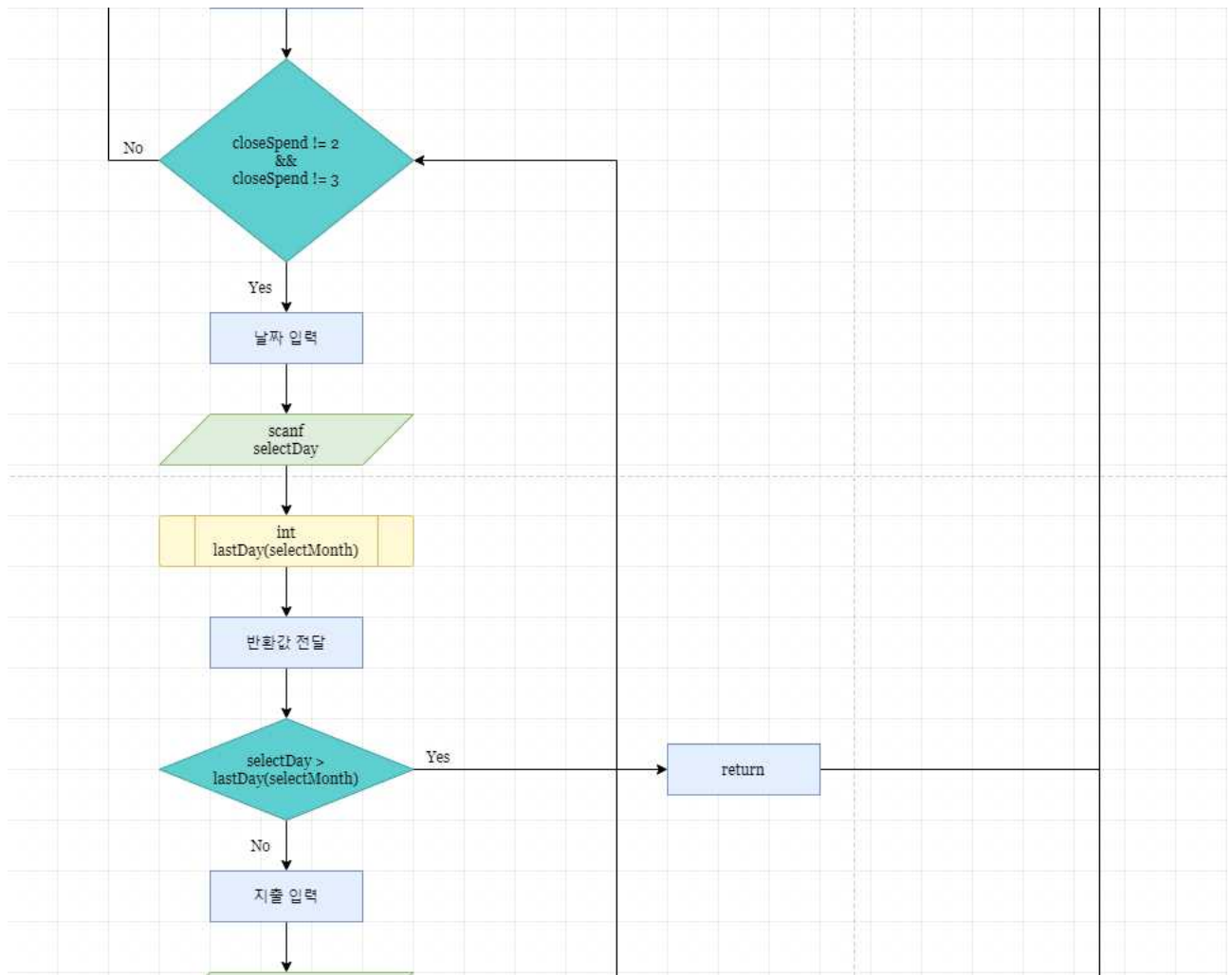


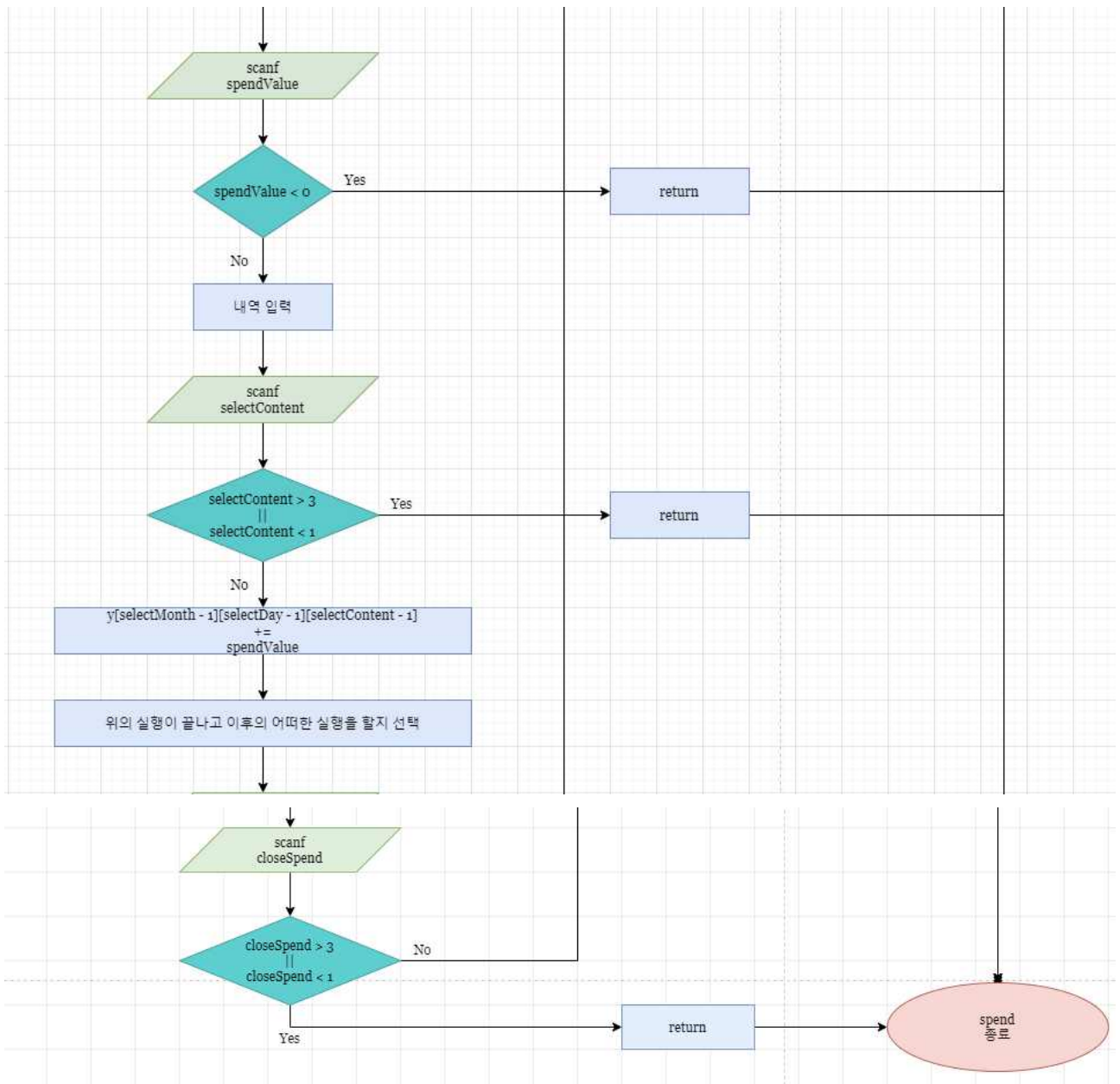
F. spend 함수 알고리즘

- 입력 : 지출 저장 월 / 지출 저장 일 / 지출 저장 내역 / 지출 금액 / 차후 실행 선택
- 저장 : 사용자가 입력한 정보를 바탕으로 지출을 지출 배열에 저장한다.
- 일처리 순서 : 지출 입력 달 입력 → 날짜 입력 → 지출 입력 → 내역 입력 → 데이터 저장 → 차후 실행 선택

spend 함수의 분할 캡처 알고리즘





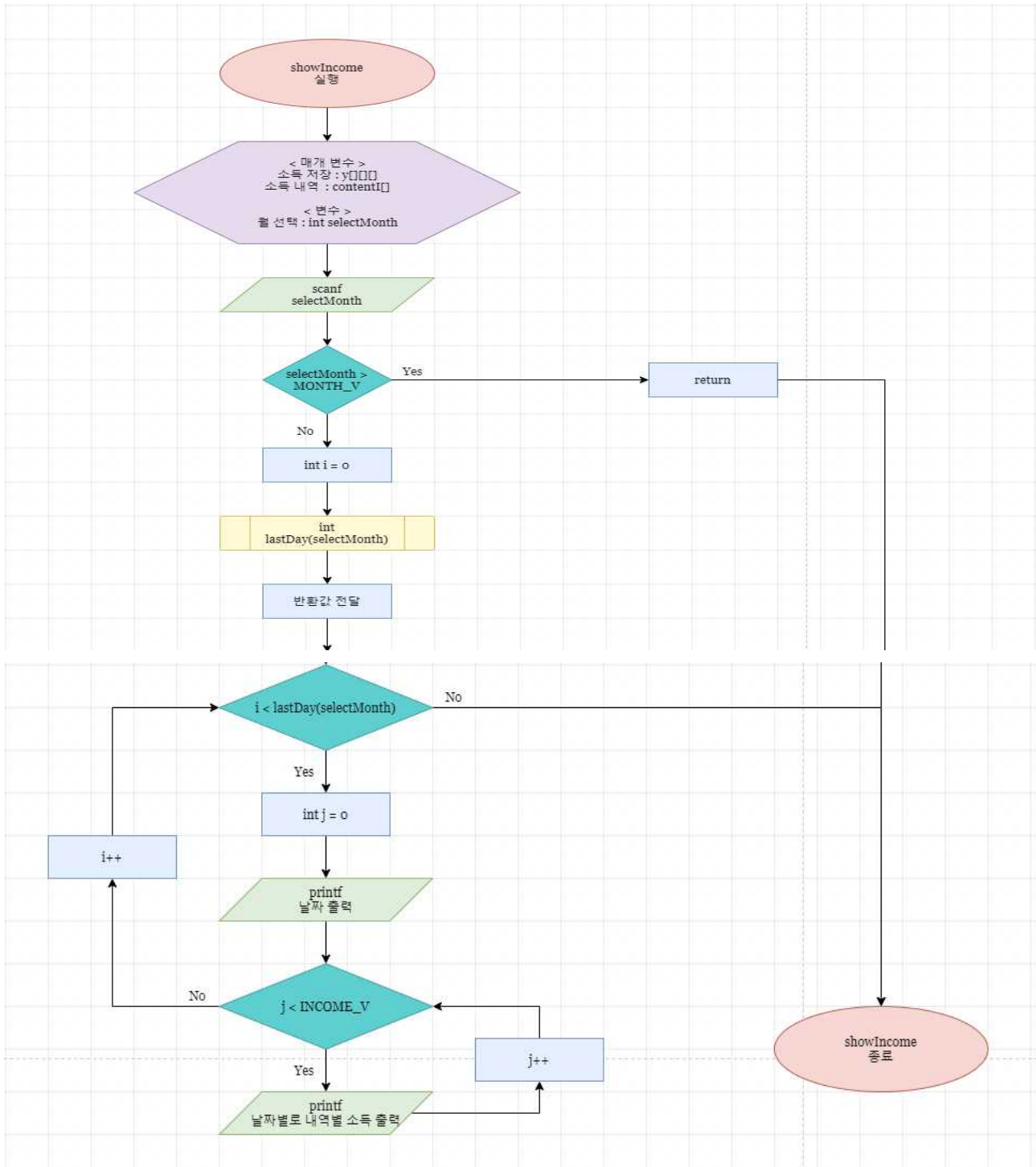


```

graph TD
    Start([array 선언]) --> Empty{array가 비어있는가?  
(array가 비어있으면, return 0)}
    Empty -- Yes --> Return0[return 0]
    Empty -- No --> SetMinValue[minValue = array[0]]
    SetMinValue --> LoopStart(( ))
    LoopStart --> LoopCond{array[i] < minValue}
    LoopCond -- Yes --> UpdateMinValue[minValue = array[i]]
    UpdateMinValue --> LoopStart
    LoopCond -- No --> LoopEnd(( ))
    LoopEnd --> ReturnMinValue[return minValue]
  
```

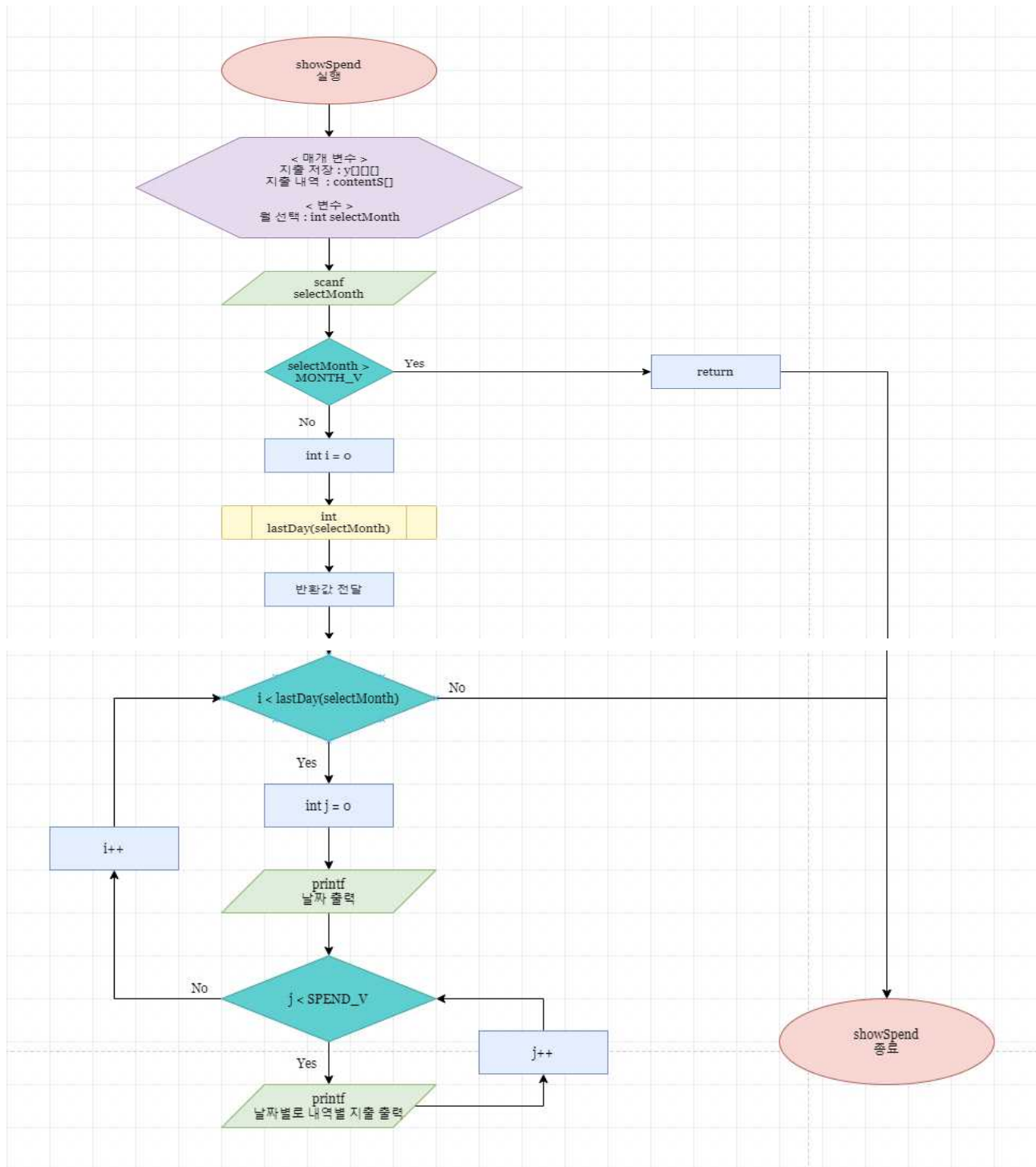
G. showIncome 함수 알고리즘

- 입력 : 소득 확인 월
- 출력 : 사용자가 선택한 월의 소득 내용을 날짜, 내역별로 출력
- 일처리 순서 : 소득 확인 월 입력 → 해당 월의 소득 현황 출력



H. showSpend 함수 알고리즘

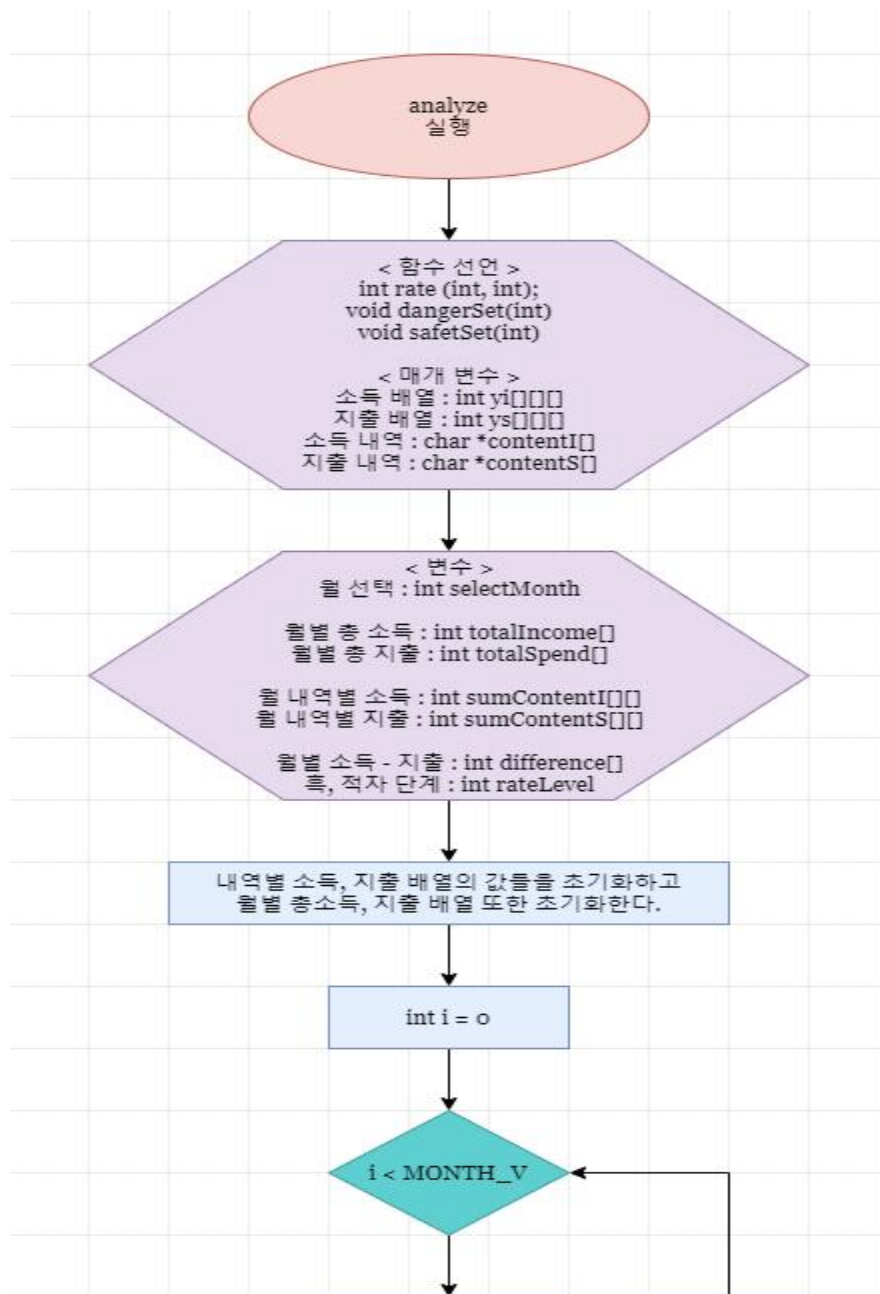
- 입력 : 지출 확인 월
- 출력 : 사용자가 선택한 월의 지출 내용을 날짜, 내역별로 출력
- 일처리 순서 : 지출 확인 월 입력 → 해당 월의 지출 현황 출력

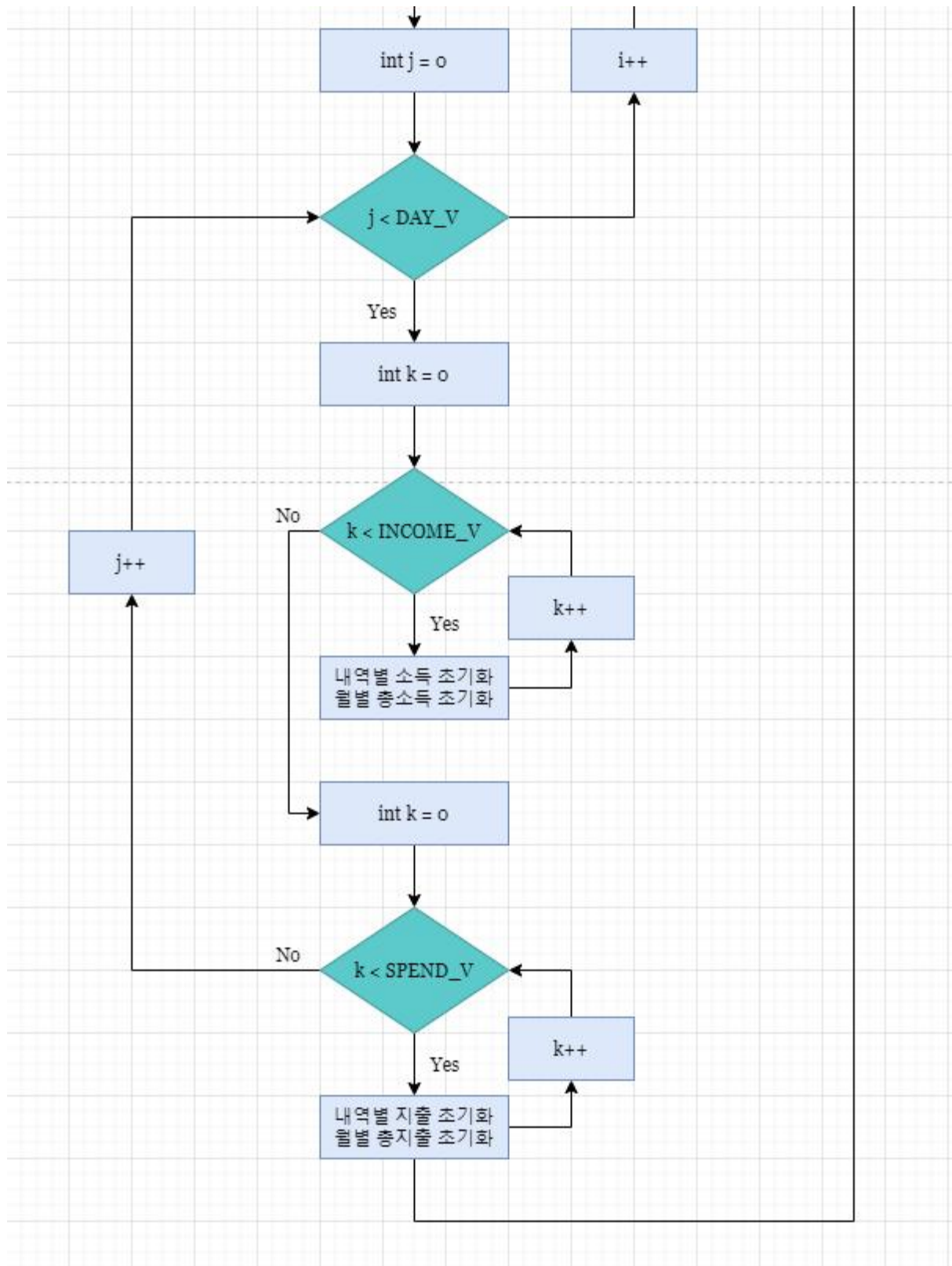


I. analyze 함수 알고리즘

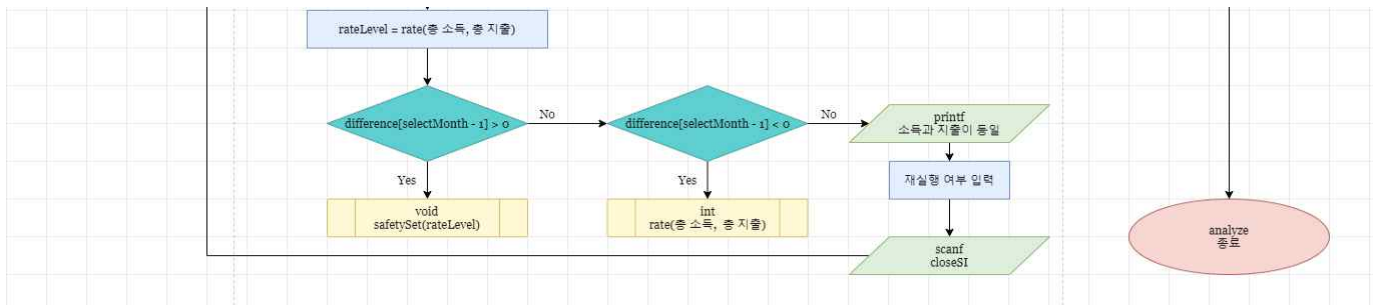
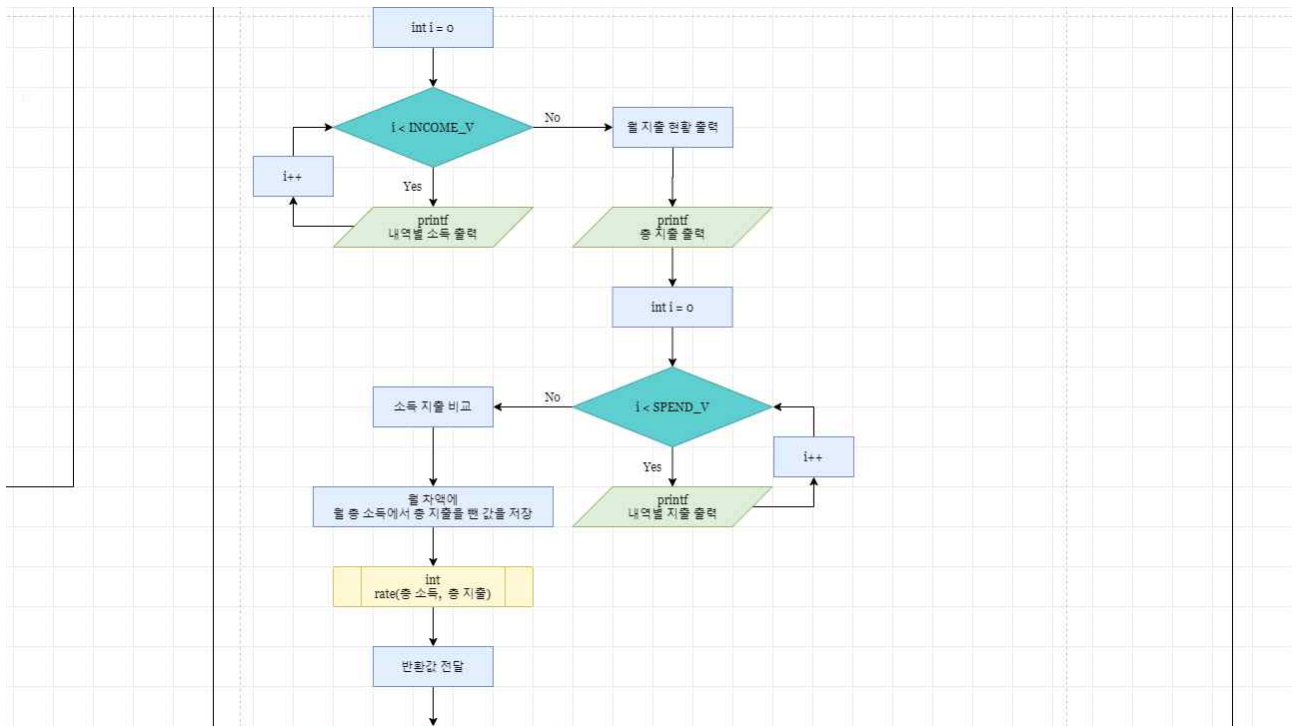
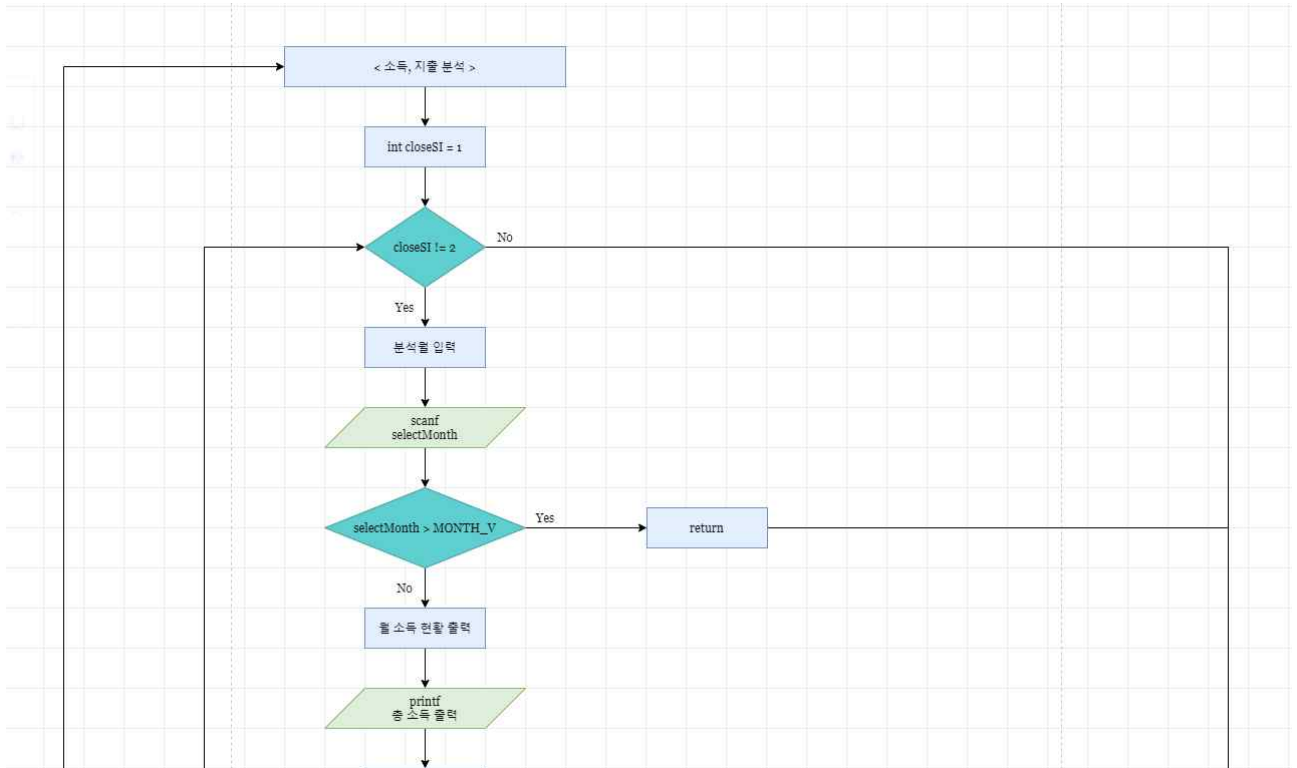
- 입력 : 분석 월 / 재실행 여부
- 출력 : 사용자가 입력한 달의 소득, 지출 현황 그리고 소득과 지출의 비교를 통한 분석
- 일처리 순서 : 분석 월 입력 → 해당월의 총 소득과 내역별 소득 출력 → 해당월의 총 지출과 내역별 지출 출력 → 해당 월의 흑자, 적자 여부를 판단하고 이에 따른 경제 행동 요령 추천
- 보조 함수 : 분석 (analyze) 함수의 기능을 보조하기 위해서 해당 소스 파일에는 3개의 보조함수가 추가되어 있다. 각각 1. 흑, 적자의 단계 2. 흑자일 때의 추천 사항 3. 적자일 때의 추천 사항을 표기하는 몸체로 구현되어 있다.

analyze 함수의 분할 캡처 알고리즘

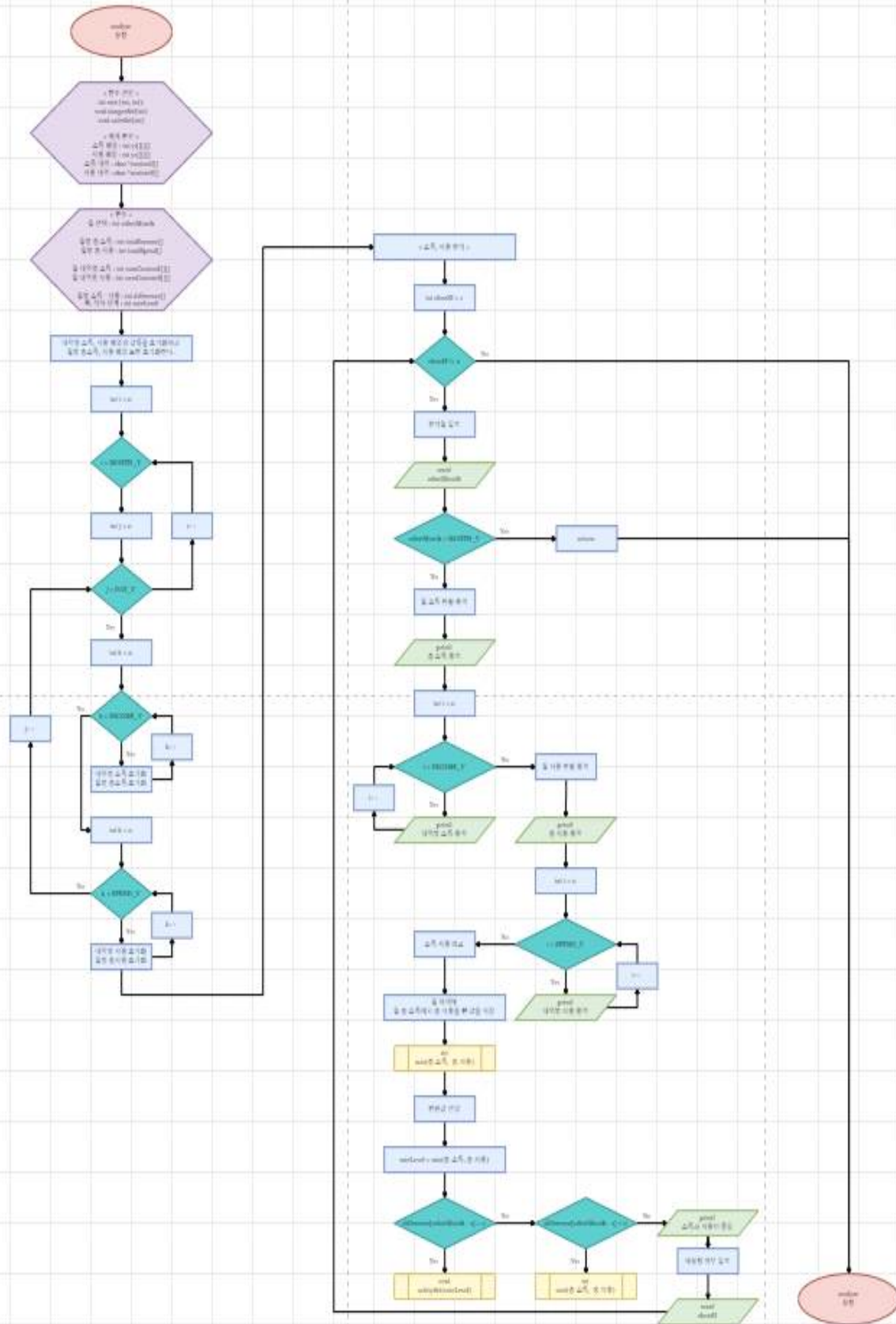




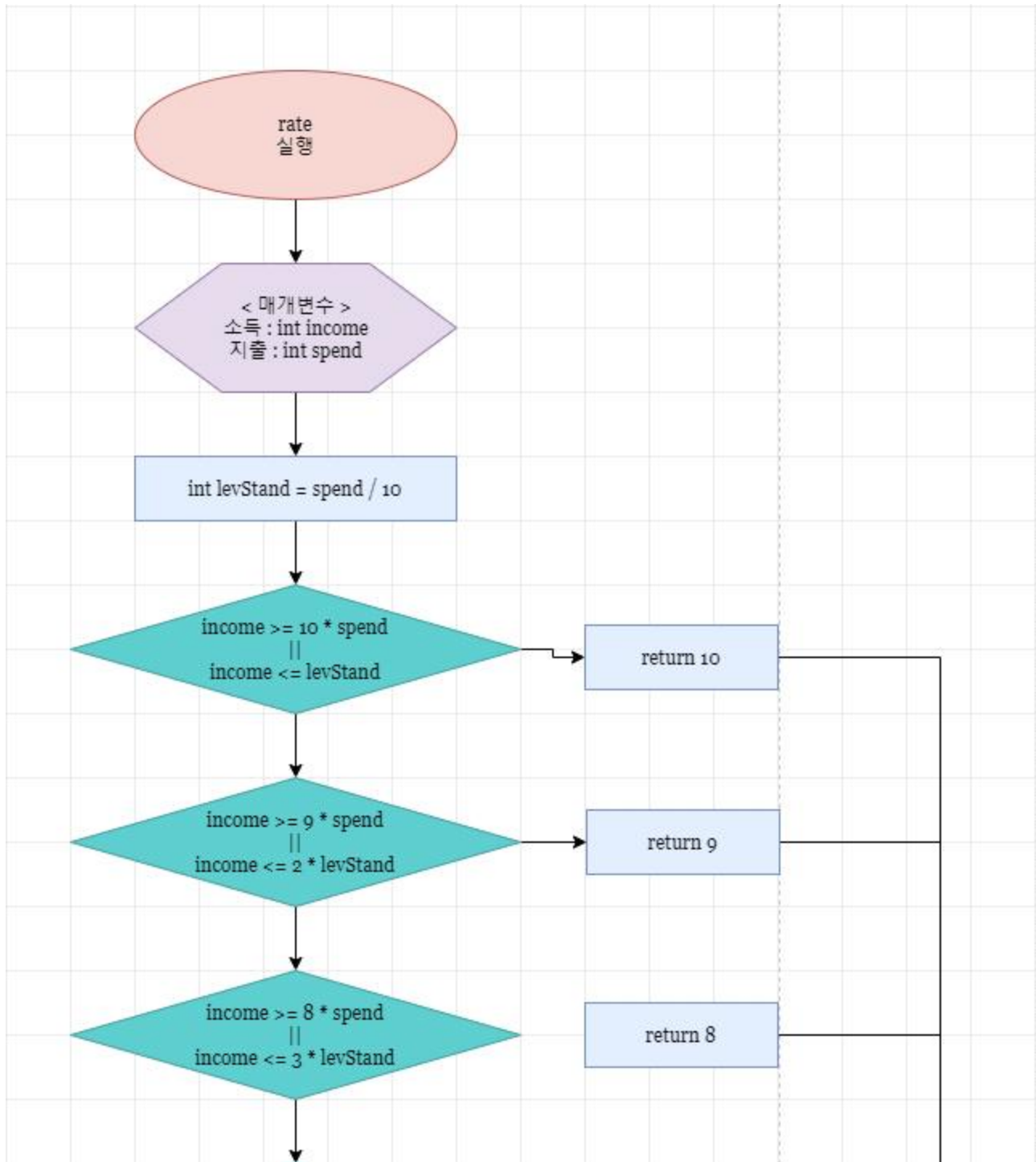
맨 오른쪽의 연결선은 밑의 알고리즘과 연결의 맨 왼쪽 연결선과 연결됩니다.

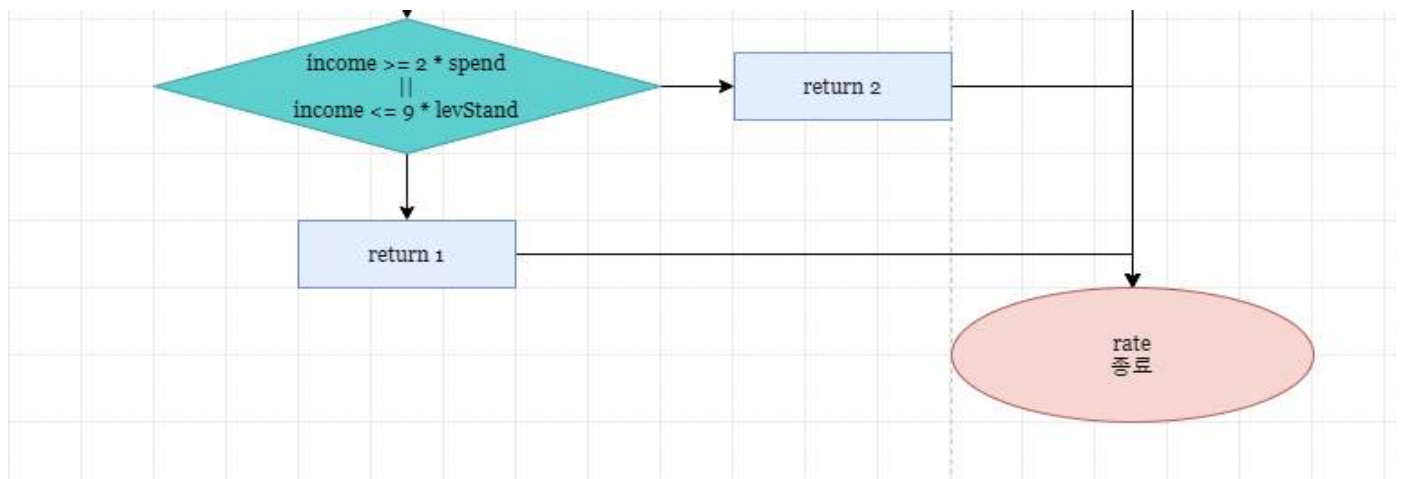
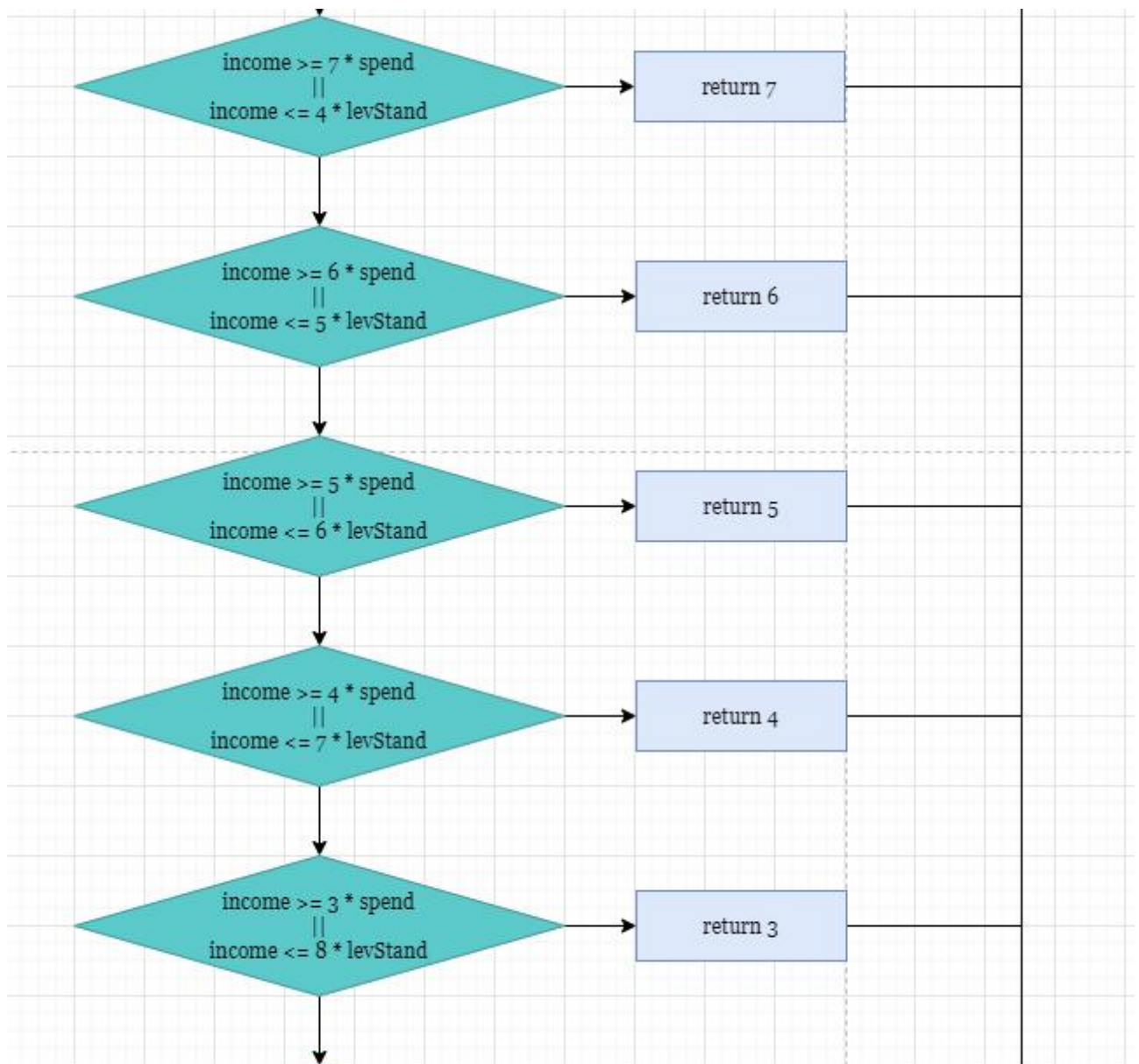


analyze 함수의 전체 캡처 알고리즘

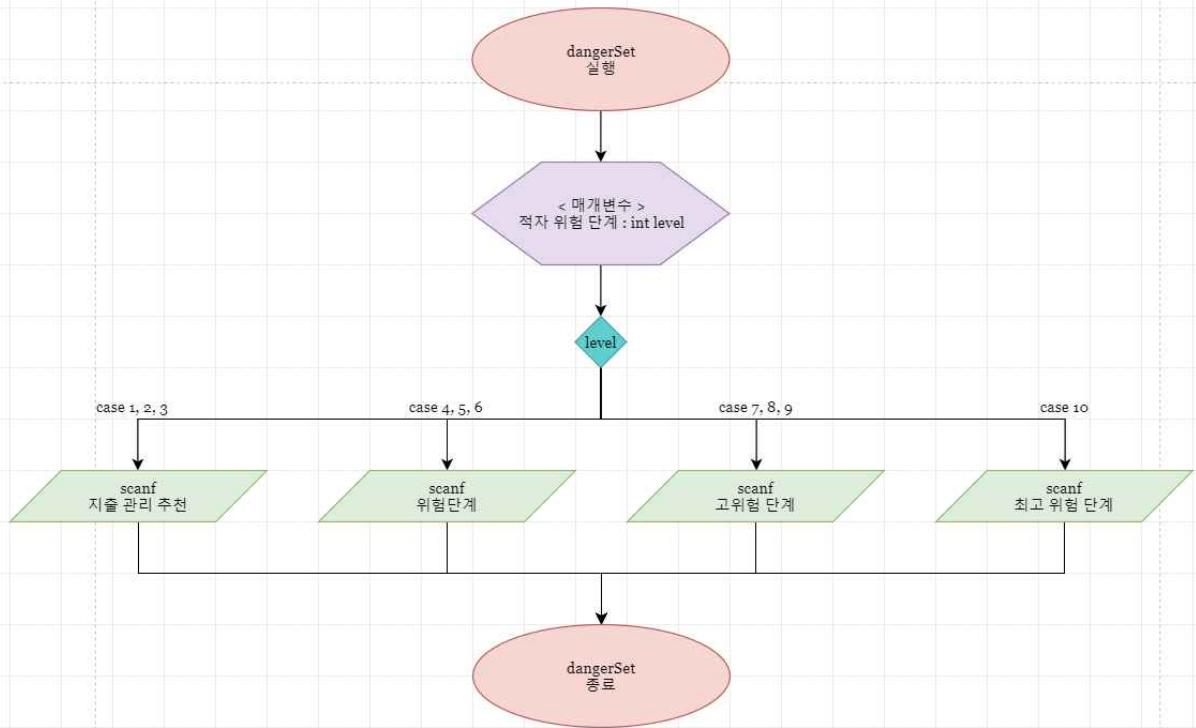


rate 함수 알고리즘

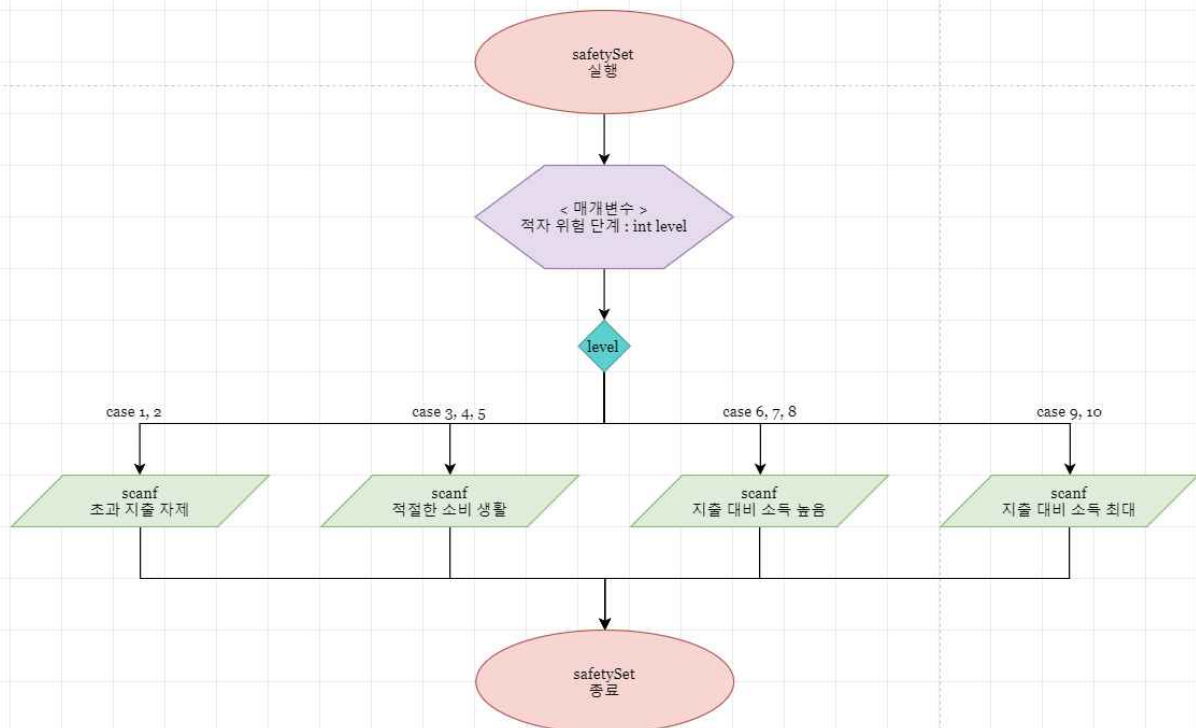




dangerSet 함수 알고리즘

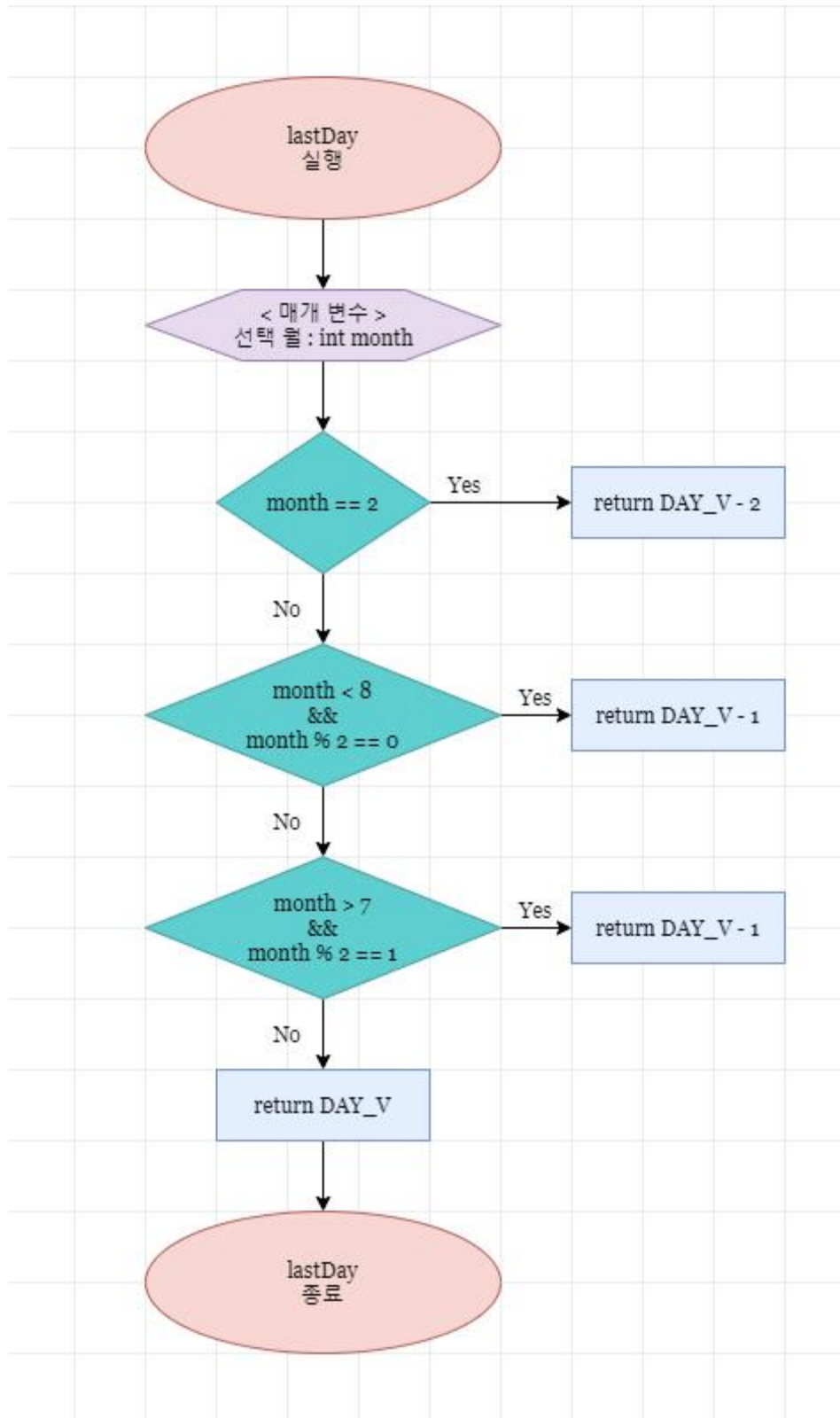


safetySet 함수 알고리즘



J. lastDay 함수 알고리즘

- 입력 : 월(int month)
- 출력 : 월별 마지막 날을 반환한다.
- 일처리 순서 : month가 2이면 (2월이면) 29를 반환한다. → month가 8미만이고 짝수이면 30을 반환한다. → month가 7 초과이고 홀수이면 30을 반환한다. → 이외의 경우에는 31을 반환한다.



4. 구현 내용(소스코드 복사해서 붙여넣기)

A. main 함수

```
#include <stdio.h>
#include <stdlib.h>
#include "aboutProgram.h"
#include "showMenu.h"
#include "performMenu.h"

#define MONTH_V    12
#define DAY_V      31
#define INCOME_V   3
#define SPEND_V    9

int main()
{
    /* 변수 선언 */
    int yearIncome[MONTH_V][DAY_V][INCOME_V] = {0};           // 소득을 저장하는 3차원 배열 [월][일][내역]
    int yearSpend[MONTH_V][DAY_V][SPEND_V] = {0};             // 지출을 저장하는 3차원 배열 [월][일][내역]
    int selectMenu;                                           // 메뉴 선택 변수
    int startProgram;                                         // 프로그램 실행 변수

    /* 소득, 지출로 표기가 가능한 내역 */
    char *contentI[INCOME_V] = {"월급", "용돈", "기타"};       // 소득 내역을 주소로 저장하는 1차원 포인터 배열
    char *contentS[SPEND_V] = {"식비", "식비외", "주거비",      // 지출 내역을 주소로 저장하는 1차원 포인터 배열
                               "의류비", "교통비", "교육비",
                               "미용비", "의료비", "기타"};

    /* 프로그램 시작 및 설명 */
    printf("< 2020년의 소득, 지출을 다루는 프로그램 >\n\n");
    aboutProgram();

    /* 프로그램 실행 조건문 */
    printf("프로그램을 실행하시겠습니까? (1. Yes 2. NO) : ");
    scanf("%d", &startProgram);
    if (startProgram == 1){
        /* 메뉴 선택 실행 */
        for (; selectMenu != 6;) {
            showMenu();                                     // 메뉴 화면 출력
            scanf("%d", &selectMenu);                     // 메뉴 옵션 선택
            printf("\n");
            performMenu(selectMenu, yearIncome, yearSpend, contentI, contentS);
        }
    }
    else
        printf("프로그램을 실행하지 않습니다.");
    return 0;
}
```

B. aboutProgram 함수

aboutProgram.h

```
#ifndef ABOUTPROGRAM_H_INCLUDED
#define ABOUTPROGRAM_H_INCLUDED

void aboutProgram();

#endif // ABOUTPROGRAM_H_INCLUDED
```

aboutProgram.c

```
#include <stdio.h>

/* 프로그램 설명 함수 */
void aboutProgram() {
    printf("=====\n");
    printf("< 기능 >\n");
    printf("이 프로그램은 사용자의 월별 소득, 지출을 저장하고 분석하는 프로그램입니다.\n\n");

    printf("< 입력 >\n");
    printf("사용자는 월/일에 대한 날짜를 입력하고 그에 대한 소득, 지출을 저장할 수 있습니다.\n");
    printf("3가지의 소득 내역과 9가지의 지출 내역을 저장할 수 있습니다.\n\n");

    printf("< 저장 >\n");
    printf("같은 날짜에서 이루어진 동일 내역은 하나의 값으로 저장됩니다.\n");
    printf("(예 : 3월 3일에 점심 1만원, 저녁 2만원 = 식비 3만원으로 저장)\n\n");

    printf("< 처리 >\n");
    printf("분석은 한 달 단위로 이루어집니다.\n\n");

    printf("< 출력 >\n");
    printf("사용자의 데이터를 바탕으로 분석한 새로운 데이터를 출력합니다.\n\n");
    printf("설명이 종료 되었으며 프로그램을 시작합니다.\n");
    printf("=====\n");
}
```

C. showMenu 함수

showMenu.h

```
#ifndef SHOWMENU_H_INCLUDED
#define SHOWMENU_H_INCLUDED

void showMenu();

#endif // SHOWMENU_H_INCLUDED
```

showMenu.c

```
#include <stdio.h>

/* 프로그램 메뉴 출력 함수 */
void showMenu(){
    printf("실행 작업 번호를 선택하세요.\n");
    printf("===== \n");
    printf("1. 소득 입력\n");
    printf("2. 지출 입력\n");
    printf("3. 소득 입력 현황 출력\n");
    printf("4. 지출 입력 현황 출력\n");
    printf("5. 월 소득, 지출 분석\n");
    printf("6. 프로그램 종료\n");
    printf("===== \n");
    printf("선택 번호 : ");
}
```

D. performMenu 함수

perfromMenu.h

```
#ifndef PERFORMMENU_H_INCLUDED
#define PERFORMMENU_H_INCLUDED

#define MONTH_V    12
#define DAY_V      31
#define INCOME_V   3
#define SPEND_V    9

void performMenu(int sm, int yi[MONTH_V][DAY_V][INCOME_V], int ys[MONTH_V][DAY_V][SPEND_V], char
*contentI[INCOME_V], char *contentS[SPEND_V]);

#endif // PERFORMMENU_H_INCLUDED
```

perfromMenu.c

```
#include <stdio.h>

#include "income.h"
#include "spend.h"
#include "showIncome.h"
#include "showSpend.h"
#include "analyze.h"

#define MONTH_V    12
#define DAY_V      31
#define INCOME_V   3
#define SPEND_V    9

/* 프로그램 메뉴 작동 함수 */
void performMenu(int sm, int yi[MONTH_V][DAY_V][INCOME_V], int ys[MONTH_V][DAY_V][SPEND_V], char
*contentI[INCOME_V], char *contentS[SPEND_V]) {
    switch(sm) {
        case 1 :
            income(yi, contentI);
            break;
        case 2 :
            spend(ys, contentS);
            break;
        case 3 :
            showIncome(yi, contentI);
            break;
        case 4 :
            showSpend(ys, contentS);
            break;
        case 5 :
            analyze(yi, ys, contentI, contentS);
            break;
        case 6 :
            printf("소득, 지출 분석 프로그램이 종료됩니다.\n");
            break;
        default :
            printf("입력이 잘못 되었습니다. 다시 입력해 주세요.\n\n ");
    }
}
```


E. income 함수

income.h

```
#ifndef INCOME_H_INCLUDED
#define INCOME_H_INCLUDED

#define MONTH_V    12
#define DAY_V      31
#define INCOME_C_V  3

void income(int y[MONTH_V][DAY_V][INCOME_C_V], char *contentI[INCOME_C_V]);

#endif // INCOME_H_INCLUDED
```

income.c

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

#define MONTH_V    12
#define DAY_V      31
#define INCOME_C_V  3

/* 소득 입력 함수 */
void income(int y[MONTH_V][DAY_V][INCOME_C_V], char *contentI[INCOME_C_V]) {

    /* 변수 */
    int selectMonth;           // 소득을 입력하려는 월을 저장하는 정수형 변수
    int selectDay;             // 소득을 입력하려는 일을 저장하는 정수형 변수
    int incomeValue;           // 소득을 입력하는 정수형 변수
    int selectContent;         // 내역을 선택하는 정수형 변수

    /* 소득 입력 반복문 */
    printf("< 소득 입력을 시작합니다 >.\n\n");
    for (int closeIncome = 1; closeIncome != 3;){           // 반복문을 1회 실행하고 closeIncome = 3이 되면 반복 종료

        /* 소득 입력을 하려는 달 입력 */
        printf("소득 입력 달 입력 : ");                     // 소득 입력 달 입력
        scanf("%d", &selectMonth);
        if (selectMonth > MONTH_V){                           // 올바른 월을 입력하지 않으면 메뉴 화면으로 초기화
            printf("올바른 월 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
            return;
        }
        closeIncome = 1;                                       // 중첩 반복문의 원활한 실행을 위해 1로 초기화

        /* 해당 월의 소득 입력 */
        for (; closeIncome != 2 && closeIncome != 3;){
            printf("\n입력하고자 하는 날, 금액, 내역을 입력\n");

            /* 날짜 입력 */
            printf("날짜 : ");
            scanf("%d", &selectDay);
            if (selectDay > lastDay(selectMonth)){             // 올바른 일을 입력하지 않으면 메뉴 화면으로 초기화
                printf("올바른 날짜 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
                return;
            }
        }
    }
}
```

```

/* 소득 입력 */
printf("금액 : ");
scanf("%d", &incomeValue);
if (incomeValue < 0){ // 올바른 형식의 금액을 입력하지 않으면 메뉴 화면으로 초기화
    printf("올바른 금액 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
    return;
}

/* 내역 입력 */
printf("내역 (1. 월급 2. 용돈 3. 기타) : ");
scanf("%d", &selectContent);
if (selectContent > 3 || selectContent < 1){ // 올바른 형식의 내역을 입력하지 않으면 메뉴 화면으로 초기화
    printf("올바른 내역 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
    return;
}

/* 데이터 저장 */
printf("%d월 %d일에 %s(으)로인한 소득 %d원을 입력합니다.\n",
    selectMonth, selectDay, content[selectContent - 1], incomeValue);
y[selectMonth - 1][selectDay - 1][selectContent - 1] += incomeValue;

/* 차후 실행 선택 */
printf("\n1. %d월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료\n", selectMonth);
printf("선택 번호 : ");
scanf("%d", &closeIncome);
if (closeIncome > 3 || closeIncome < 1) {
    printf("소득 입력을 종료합니다.");
    return;
}
}

printf("소득 입력이 종료되었습니다.\n\n");
}

```

F. spend 함수

spend.h

```
#ifndef SPEND_H_INCLUDED
#define SPEND_H_INCLUDED

#define MONTH_V    12
#define DAY_V      31
#define SPENDC_V   9

void spend(int y[MONTH_V][DAY_V][SPENDC_V], char *contentS[SPENDC_V]);

#endif // SPEND_H_INCLUDED
```

spend.c

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

#define MONTH_V    12
#define DAY_V      31
#define SPENDC_V   9

/* 지출 입력 함수 */
void spend(int y[MONTH_V][DAY_V][SPENDC_V], char *contentS[SPENDC_V]) {

    /* 변수 */
    int selectMonth;           // 지출을 입력하려는 월을 저장하는 정수형 변수
    int selectDay;             // 지출을 입력하려는 일을 저장하는 정수형 변수
    int spendValue;            // 지출을 입력하는 정수형 변수
    int selectContent;         // 내역을 선택하는 정수형 변수

    /* 지출 입력 반복문 */
    printf("< 지출 입력을 시작합니다 >.\n\n");
    for (int closeSpend = 1; closeSpend != 3;){           // 반복문을 1회 실행하고 closeSpend = 3이 되면 반복 종료

        /* 지출 입력을 하려는 달 입력 */
        printf("지출 입력 달 입력 : ");                   // 지출 입력 달 입력
        scanf("%d", &selectMonth);
        if (selectMonth > MONTH_V){                       // 올바른 월을 입력하지 않으면 메뉴 화면으로 초기화
            printf("올바른 월 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
            return;
        }
        closeSpend = 1;                                   // 중첩 반복문의 원활한 실행을 위해 1로 초기화

        /* 해당 월의 지출 입력 */
        for (; closeSpend != 2 && closeSpend != 3;){
            printf("\n입력하고자 하는 날, 금액, 내역을 입력\n");

            /* 날짜 입력 */
            printf("날짜 : ");
            scanf("%d", &selectDay);
            if (selectDay > lastDay(selectMonth)){         // 올바른 일을 입력하지 않으면 메뉴 화면으로 초기화
                printf("올바른 날짜 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
                return;
            }
        }
    }
}
```

```

/* 지출 입력 */
printf("금액 : ");
scanf("%d", &spendValue);
if (spendValue < 0){ // 올바른 형식의 금액을 입력하지 않으면 메뉴 화면으로 초기화
    printf("올바른 금액 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
    return;
}

/* 내역 입력 */
printf("내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : ");
scanf("%d", &selectContent);
if (selectContent > 9 || selectContent < 1){ // 올바른 형식의 내역을 입력하지 않으면 메뉴 화면으로 초기화
    printf("올바른 내역 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
    return;
}

/* 데이터 저장 */
printf("%d월 %d일에 %s(으)로인한 지출 %d원을 입력합니다.\n",
    selectMonth, selectDay, contentS[selectContent - 1], spendValue);
y[selectMonth - 1][selectDay - 1][selectContent - 1] += spendValue;

/* 차후 실행 선택 */
printf("\n1. %d월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료\n", selectMonth);
printf("선택 번호 : ");
scanf("%d", &closeSpend);
if (closeSpend > 3 || closeSpend < 1) {
    printf("지출 입력을 종료합니다.");
    return;
}
}
}
printf("지출 입력이 종료되었습니다.\n\n");
}

```

G. showIncome 함수

showIncome.h

```
#ifndef SHOWINCOME_H_INCLUDED
#define SHOWINCOME_H_INCLUDED

#define MONTH_V    12
#define DAY_V      31
#define INCOME_V    3

void showIncome(int y[MONTH_V][DAY_V][INCOME_V], char *contentI[INCOME_V]);

#endif // SHOWINCOME_H_INCLUDED
```

showIncome.c

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

#define MONTH_V    12
#define DAY_V      31
#define INCOME_V    3

/* 선택 월의 소득 입력 현황을 확인하는 함수 */
void showIncome(int y[MONTH_V][DAY_V][INCOME_V], char *contentI[INCOME_V]) {

    /* 변수 */
    int selectMonth; // 소득 현황을 확인하고자 하는 달을 저장하는 정수형 변수

    /* 달 선택 */
    printf("현황을 확인하고자 하는 월을 입력하세요.\n");
    printf("월 : ");
    scanf("%d", &selectMonth);
    if (selectMonth > MONTH_V) // 올바른 월을 입력하지 않으면 메뉴 화면으로 초기화
        printf("올바른 월 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
    return;
}

/* 선택 월의 소득 입력 현황 출력 반복문 */
printf("< %d월 달의 소득 현황 >\n", selectMonth);
printf("=====\\n");
for (int i = 0; i < lastDay(selectMonth); i++){
    printf("%2d일 : ( ", i + 1);
    for (int j = 0; j < INCOME_V; j++){
        printf("%s : %7d원 ", contentI[j], y[selectMonth - 1][i][j]);
    }
    printf("\\n");
}
printf("=====\\n\\n");
}
```

H. showSpend 함수

showSpend.h

```
#ifndef SHOWSPEND_H_INCLUDED
#define SHOWSPEND_H_INCLUDED

void showSpend(int y[MONTH_V][DAY_V][SPENDC_V], char *contentS[SPENDC_V]);

#endif // SHOWSPEND_H_INCLUDED
```

showSpend.c

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

#define MONTH_V    12
#define DAY_V      31
#define SPENDC_V   9

/* 선택 월의 지출 입력 현황을 확인하는 함수 */
void showSpend(int y[MONTH_V][DAY_V][SPENDC_V], char *contentS[SPENDC_V]) {

    /* 변수 */
    int selectMonth;                                // 지출 현황을 확인하고자 하는 달을 저장하는 정수형 변수

    /* 달 선택 */
    printf("현황을 확인하고자 하는 월을 입력하세요.\n");
    printf("월 : ");
    scanf("%d", &selectMonth);
    if (selectMonth > MONTH_V){                      // 올바른 월을 입력하지 않으면 메뉴 화면으로 초기화
        printf("올바른 월 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
        return;
    }

    /* 선택 월의 지출 입력 현황 출력 반복문 */
    printf("< %d월 달의 지출 현황 >\n", selectMonth);
    printf("=====\n");
    for (int i = 0; i < lastDay(selectMonth); i++){
        printf("%2d일\n", i + 1);
        for (int j = 0; j < SPENDC_V; j++){
            printf("%-s : %7d원 ", contentS[j], y[selectMonth - 1][i][j]);
            if ((j != 0) && ((j + 1) % 3 == 0))
                printf("\n");
        }
        printf("\n");
    }
    printf("=====\n\n");
}
```

I. analyze 함수 + analyze 함수의 보조 함수

analyze.h

```
#ifndef ANALYZE_H_INCLUDED
```

```
#define ANALYZE_H_INCLUDED
```

```
void analyze(int yi[MONTH_V][DAY_V][INCOMEC_V], int ys[MONTH_V][DAY_V][SPENDC_V], char  
*contentI[INCOMEC_V], char *contentS[SPENDC_V]);
```

```
#endif // ANALYZE_H_INCLUDED
```

analyze.c

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

```
#define MONTH_V 12
```

```
#define DAY_V 31
```

```
#define INCOMEC_V 3
```

```
#define SPENDC_V 9
```

```
int rate(int level, int income);
```

```
void dangerSet(int level);
```

```
void safetySet(int level);
```

```
/* 소득, 지출 분석 함수 */
```

```
void analyze(int yi[MONTH_V][DAY_V][INCOMEC_V], int ys[MONTH_V][DAY_V][SPENDC_V], char  
*contentI[INCOMEC_V], char *contentS[SPENDC_V]) {
```

```
    /* 변수 */
```

```
    int selectMonth;
```

```
    // 분석하고자 하는 월 선택
```

```
    int totalIncome[MONTH_V] = {0};
```

```
    // 월별 총 소득 저장 1차원 배열 (합계를 위해 0으로 초기화)
```

```
    int totalSpend[MONTH_V] = {0};
```

```
    // 월별 총 지출 저장 1차원 배열 (합계를 위해 0으로 초기화)
```

```
    int sumContentI[MONTH_V][INCOMEC_V] = {0};    // 월별, 내역별로 소득을 저장하는 2차원 배열 (합계를 위해 0  
으로 초기화)
```

```
    int sumContentS[MONTH_V][SPENDC_V] = {0};    // 월별, 내역별로 지출을 저장하는 2차원 배열 (합계를 위해 0  
으로 초기화)
```

```
    int difference[MONTH_V];
```

```
    // 월별 소득, 지출 차액
```

```
    int rateLevel;
```

```
    // 적자 위험 단계
```

```
/* 각 월별 소득, 지출을 내역별로 합계 내는 중첩 반복문 */
```

```
for (int i = 0; i < MONTH_V; i++) {
```

```
    for (int j = 0; j < DAY_V; j++) {
```

```
        for (int k = 0; k < INCOMEC_V; k++) {
```

```
            sumContentI[i][k] += yi[i][j][k];
```

```
            totalIncome[i] += yi[i][j][k];
```

```
        }
```

```
        for (int k = 0; k < SPENDC_V; k++) {
```

```
            sumContentS[i][k] += ys[i][j][k];
```

```
            totalSpend[i] += ys[i][j][k];
```

```
        }
```

```
    }
```

```
}
```

```
/* 소득 지출 분석 반복문 */
```

```
printf(" < 소득, 지출 분석 > ");
```

```

for (int closeSI = 1; closeSI != 2; ){
    /* 월 입력 */
    printf("소득 지출 분석을 확인하고자 하는 달을 입력하세요.\n월 : ");
    scanf("%d", &selectMonth);
    if (selectMonth > MONTH_V){
        // 올바른 월을 입력하지 않으면 메뉴 화면으로 초기화
        printf("올바른 월 입력이 아닙니다. Menu 화면으로 넘어갑니다.\n\n");
        return;
    }

    printf("\n%d월의 소득, 지출 분석 현황입니다.\n\n", selectMonth);

    /* 소득 현황 */
    printf("\n%d월의 소득 현황입니다.\n", selectMonth);
    printf("=====\n");
    printf("%d월의 총 소득 : %d원\n", selectMonth, totalIncome[selectMonth - 1]);
    for (int i = 0; i < INCOMEC_V; i++) {
        printf("%s에 의한 소득 : %d원\n", contentI[i], sumContentI[selectMonth - 1][i]);
    }
    printf("=====\n\n");

    /* 지출 현황 */
    printf("%d월의 지출 현황입니다.\n", selectMonth);
    printf("=====\n");
    printf("%d월의 총 지출 : %d원\n", selectMonth, totalSpend[selectMonth - 1]);
    for (int i = 0; i < SPENDC_V; i++) {
        printf("%s에 의한 지출 : %d원\n", contentS[i], sumContentS[selectMonth - 1][i]);
    }
    printf("=====\n\n");

    /* 소득, 지출 비교 */
    printf("%d월의 소득, 지출 비교 현황입니다.\n", selectMonth);
    printf("=====\n");
    difference[selectMonth - 1] = totalIncome[selectMonth - 1] - totalSpend[selectMonth - 1];
    rateLevel = rate(totalIncome[selectMonth - 1], totalSpend[selectMonth - 1]);

    /* 흑자일 때의 출력 */
    if (difference[selectMonth - 1] > 0){
        printf("소득이 지출보다 %d원 더 많습니다.\n", difference[selectMonth - 1]);
        printf("흑자 안정성 단계 10단계 중 %d단계 입니다.\n", rateLevel);
        safetySet(rateLevel);
    }

    /* 적자일 때의 출력 */
    else if (difference[selectMonth - 1] < 0){
        printf("지출이 소득보다 %d원 더 많습니다.\n", -difference[selectMonth - 1]);
        printf("적자 위험 단계 10단계 중 %d단계 입니다.\n", rateLevel);
        dangerSet(rateLevel);
    }

    /* 소득 = 지출일 때의 출력 */
    else{
        printf("소득과 지출이 동일합니다.\n");
        printf("=====\n");
    }
}

```



```

    /* 이후 실행 선택 */
    printf("분석에 대한 추가 확인이 있으면 1을 없으면 2를 입력하세요. : ");
    scanf("%d", &closeSI);
    printf("\n\n");
}
}

```

rate.h

```

#ifndef RATE_H_INCLUDED
#define RATE_H_INCLUDED

int rate(int income, int spend);

#endif // RATE_H_INCLUDED

```

rate.c

```

/* 적자 위험 단계 함수 */
/* 지출에 대한 소득의 비율을 확인하여 위험 단계 반환 */
int rate(int income, int spend){
    int levStand = spend / 10;

    if (income >= 10 * spend || income <= levStand)
        return 10;
    else if (income >= 9 * spend || income <= 2 * levStand)
        return 9;
    else if (income >= 8 * spend || income <= 3 * levStand)
        return 8;
    else if (income >= 7 * spend || income <= 4 * levStand)
        return 7;
    else if (income >= 6 * spend || income <= 5 * levStand)
        return 6;
    else if (income >= 5 * spend || income <= 6 * levStand)
        return 5;
    else if (income >= 4 * spend || income <= 7 * levStand)
        return 4;
    else if (income >= 3 * spend || income <= 8 * levStand)
        return 3;
    else if (income >= 2 * spend || income <= 9 * levStand)
        return 2;
    else
        return 1;
}

```

dangerSet.h

```

#ifndef DANGERSET_H_INCLUDED
#define DANGERSET_H_INCLUDED

void dangerSet(int level);

#endif // DANGERSET_H_INCLUDED

```

dangerSet.c

```

/* 위험 단계에 따른 추천 행동 요령 */
void dangerSet(int level){
    switch(level) {
        // 위험 단계를 1,2,3 단계 4, 5, 6 단계 7, 8, 9 단계 10 단계로 구별하여 그에 맞는 행동 요령을 추천한다.
        case 1 :
        case 2 :

```

```

        case 3 : printf("위험 단계는 아니지만 지출 관리를 하는 것을 추천합니다.\n"); break;
        case 4 :
        case 5 :
        case 6 : printf("고위험까지는 아니지만 불필요한 지출은 최대한 줄이는 것을 추천합니다.\n"); break;
        case 7 :
        case 8 :
        case 9 : printf("고위험 단계입니다. 필요한 지출을 제외하고는 지출을 줄이는 것을 추천합니다.\n"); break;
        case 10 : printf("최고 위험 단계입니다. 지출을 최소화하는 것을 권고드립니다.\n");
    }
}

```

safetySet.h

```

#ifndef SAFETYSET_H_INCLUDED
#define SAFETYSET_H_INCLUDED

void safetySet(int level);

#endif // SAFETYSET_H_INCLUDED

```

safetySet.c

```

/* 안정성 단계에 따른 추천 행동 요령 */
void safetySet(int level){
    // 안정성 단계를 1, 2 단계 3, 4, 5단계 6, 7, 8단계 9, 10단계로 구별하여 그에 맞는 행동 요령을 추천한다.
    switch(level) {
        case 1 :
        case 2 : printf("소득이 지출보다 많지만 이 이상의 지출은 자제하시길 바랍니다.");
        case 3 :
        case 4 :
        case 5 : printf("적절한 소비 생활입니다. 대출금 조기 상환을 하시길 추천드립니다.");
        case 6 :
        case 7 :
        case 8 : printf("지출 대비 소득이 높습니다. 적금을 통해 돈을 모으시는 것을 추천드립니다.");
        case 9 :
        case 10 : printf("지출 대비 소득이 상당히 높습니다. 삶의 질을 높이기 위한 지출을 하는 것을 추천드립니다.");
    }
}

```

J. lastDay 함수

lastDay.h

```
#ifndef LASTDAY_H_INCLUDED
#define LASTDAY_H_INCLUDED

int lastDay(int month);

#endif // LASTDAY_H_INCLUDED
```

lastDay.c

```
#include <stdio.h>

#define DAY_V      31

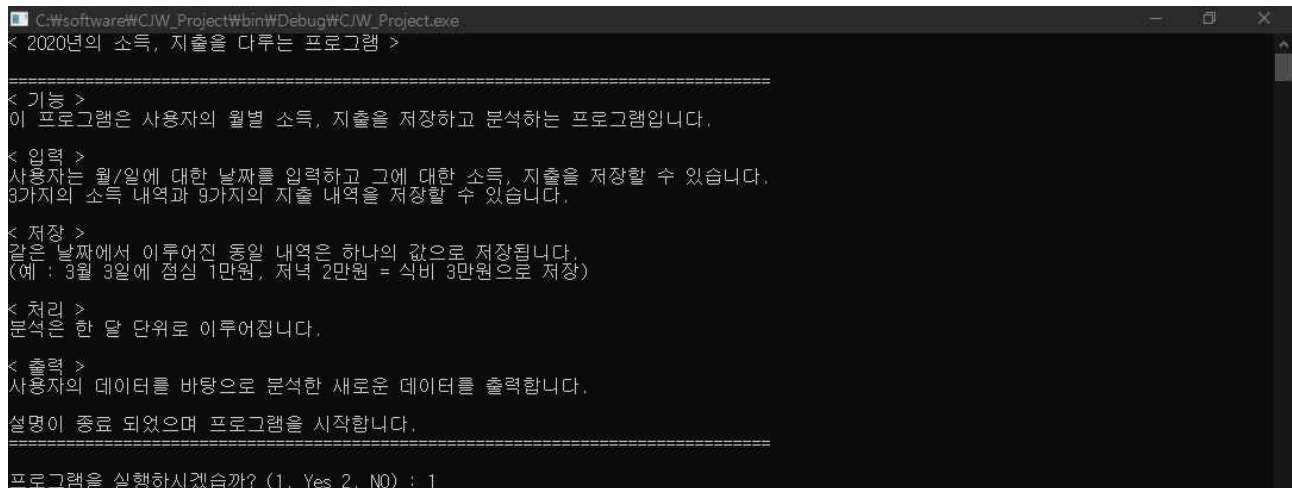
int lastDay(int month) {
    if (month == 2)
        return DAY_V - 2;
    else if ((month < 8) && (month % 2 == 0))
        return DAY_V - 1;
    else if ((month > 7) && (month % 2 == 1))
        return DAY_V - 1;
    else
        return DAY_V;
}
```

5. 시현 결과

(실제 구동되는 장면 복사해서 넣음...)

2. 실제로 저의 1월달 가계부를 바탕으로 프로그램을 실행하였습니다.

A. 시작 화면 및 프로그램 실행 선택



```
C:\software\CJW_Project\Win\Debug\CJW_Project.exe
> 2020년의 소득, 지출을 다루는 프로그램 >

=====
< 기능 >
이 프로그램은 사용자의 월별 소득, 지출을 저장하고 분석하는 프로그램입니다.

< 입력 >
사용자는 월/일에 대한 날짜를 입력하고 그에 대한 소득, 지출을 저장할 수 있습니다.
3가지의 소득 내역과 9가지의 지출 내역을 저장할 수 있습니다.

< 저장 >
같은 날짜에서 이루어진 동일 내역은 하나의 값으로 저장됩니다.
(예 : 3월 3일에 점심 1만원, 저녁 2만원 = 식비 3만원으로 저장)

< 처리 >
분석은 한 달 단위로 이루어집니다.

< 출력 >
사용자의 데이터를 바탕으로 분석한 새로운 데이터를 출력합니다.

설명이 종료 되었으며 프로그램을 시작합니다.
=====

프로그램을 실행하시겠습니까? (1. Yes 2. NO) : 1
```

B. 메뉴 옵션 1을 선택하여 1월 소득 입력

```
C:\software\CJW_Project\Wbin\Debug\CJW_Project.exe
실행 작업 번호를 선택하세요.
=====
1. 소득 입력
2. 지출 입력
3. 소득 입력 현황 출력
4. 지출 입력 현황 출력
5. 월 소득 지출 분석
6. 프로그램 종료
=====
선택 번호 : 1
< 소득 입력을 시작합니다 >.
소득 입력 달 입력 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 2
금액 : 41600
내역 (1. 월급 2. 용돈 3. 기타) : 3
1월 2일에 기타(으)로인한 소득 41600원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 9
금액 : 1
내역 (1. 월급 2. 용돈 3. 기타) : 3
1월 9일에 기타(으)로인한 소득 1원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 10
금액 : 19000
내역 (1. 월급 2. 용돈 3. 기타) : 3
1월 10일에 기타(으)로인한 소득 19000원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 12
금액 : 765000
내역 (1. 월급 2. 용돈 3. 기타) : 1
1월 12일에 월급(으)로인한 소득 765000원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 16
금액 : 24000
내역 (1. 월급 2. 용돈 3. 기타) : 3
1월 16일에 기타(으)로인한 소득 24000원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 19
금액 : 200000
내역 (1. 월급 2. 용돈 3. 기타) : 2
1월 19일에 용돈(으)로인한 소득 200000원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 22
금액 : 200000
내역 (1. 월급 2. 용돈 3. 기타) : 2
1월 22일에 용돈(으)로인한 소득 200000원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 25
금액 : 200000
내역 (1. 월급 2. 용돈 3. 기타) : 2
1월 25일에 용돈(으)로인한 소득 200000원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 26
금액 : 34500
내역 (1. 월급 2. 용돈 3. 기타) : 3
1월 26일에 기타(으)로인한 소득 34500원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 1
입력하고자 하는 날, 금액, 내역을 입력
날짜 : 28
금액 : 7000
내역 (1. 월급 2. 용돈 3. 기타) : 3
1월 28일에 기타(으)로인한 소득 7000원을 입력합니다.
1. 1월의 추가 소득 입력 2. 다른 달의 소득 입력 3. 소득 입력 종료
선택 번호 : 3
소득 입력이 종료되었습니다.
```

C. 메뉴 옵션 2를 선택하여 1월 지출 입력

```
C:\software\CJW_Project\Win\Debug\CJW_Project.exe

실행 작업 번호를 선택하세요.
=====
1. 소득 입력
2. 지출 입력
3. 소득입력의 현황 출력
4. 지출입력의 현황 출력
5. 월 소득, 지출 분석
6. 프로그램 종료
=====
선택 번호 : 2

< 지출 입력을 시작합니다 >.

지출 입력 달 입력 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 2
금액 : 115000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 7
1월 2일에 미용비(으)로인한 지출 115000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 2
금액 : 8690
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9
1월 2일에 기타(으)로인한 지출 8690원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 2
금액 : 86000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 4
1월 2일에 의류비(으)로인한 지출 86000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 2
금액 : 7700
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5
1월 2일에 교통비(으)로인한 지출 7700원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 2
금액 : 5500
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5
1월 2일에 교통비(으)로인한 지출 5500원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 3
금액 : 3000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 7
1월 3일에 미용비(으)로인한 지출 3000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 3
금액 : 15000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1
1월 3일에 식비(으)로인한 지출 15000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 3
금액 : 10300
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5
1월 3일에 교통비(으)로인한 지출 10300원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 3
금액 : 4800
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5
1월 3일에 교통비(으)로인한 지출 4800원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 4
금액 : 18000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9
1월 4일에 기타(으)로인한 지출 18000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 4
금액 : 10300
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5
1월 4일에 교통비(으)로인한 지출 10300원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1
```

C:\software\CJW_Project\bin\Debug\CJW_Project.exe

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 4

금액 : 26000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 4일에 교통비(으)로인한 지출 26000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 4

금액 : 7300

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1

1월 4일에 식비(으)로인한 지출 7300원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 6

금액 : 21700

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1

1월 6일에 식비(으)로인한 지출 21700원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 6

금액 : 23200

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 6일에 기타(으)로인한 지출 23200원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 7

금액 : 14000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1

1월 7일에 식비(으)로인한 지출 14000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 7

금액 : 5500

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2

1월 7일에 식비외(으)로인한 지출 5500원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 8

금액 : 164600

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 8일에 기타(으)로인한 지출 164600원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 9

금액 : 20000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 3

1월 9일에 주거비(으)로인한 지출 20000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 9

금액 : 100000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 9일에 기타(으)로인한 지출 100000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 10

금액 : 27250

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 10일에 교통비(으)로인한 지출 27250원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 10

금액 : 26000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 10일에 교통비(으)로인한 지출 26000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 11

금액 : 8300

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 11일에 교통비(으)로인한 지출 8300원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료

선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 11

금액 : 1700

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2

1월 11일에 식비외(으)로인한 지출 1700원을 입력합니다.

C:\software\CJW_Project\bin\Debug\CJW_Project.exe

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 11

금액 : 5600

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2

1월 11일에 식비외(으)로인한 지출 5600원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 12

금액 : 4920

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2

1월 12일에 식비외(으)로인한 지출 4920원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 12

금액 : 14500

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1

1월 12일에 식비(으)로인한 지출 14500원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 13

금액 : 4750

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2

1월 13일에 식비외(으)로인한 지출 4750원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 16

금액 : 17500

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1

1월 16일에 식비(으)로인한 지출 17500원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 16

금액 : 5800

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2

1월 16일에 식비외(으)로인한 지출 5800원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 16

금액 : 6700

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 16일에 교통비(으)로인한 지출 6700원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 16

금액 : 20300

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 16일에 교통비(으)로인한 지출 20300원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 17

금액 : 25000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 4

1월 17일에 의류비(으)로인한 지출 25000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 17

금액 : 13000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2

1월 17일에 식비외(으)로인한 지출 13000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 17

금액 : 20300

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 17일에 교통비(으)로인한 지출 20300원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 18

금액 : 70000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 18일에 기타(으)로인한 지출 70000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 19

금액 : 50000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

C:\software\CJW_Project\bin\Debug\CJW_Project.exe

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 19

금액 : 1000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 19일에 기타(으)로인한 지출 1000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 19

금액 : 1100

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 19일에 교통비(으)로인한 지출 1100원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 19

금액 : 8500

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1

1월 19일에 식비(으)로인한 지출 8500원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 20

금액 : 12000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 20일에 기타(으)로인한 지출 12000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 20

금액 : 16360

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 20일에 기타(으)로인한 지출 16360원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 21

금액 : 2000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 21일에 기타(으)로인한 지출 2000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 21

금액 : 9500

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2

1월 21일에 식비외(으)로인한 지출 9500원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 21

금액 : 5300

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 21일에 기타(으)로인한 지출 5300원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 21

금액 : 15000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9

1월 21일에 기타(으)로인한 지출 15000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 22

금액 : 13800

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 22일에 교통비(으)로인한 지출 13800원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 22

금액 : 5000

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 5

1월 22일에 교통비(으)로인한 지출 5000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력

날짜 : 23

금액 : 10500

내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1

```
C:\software\CJW_Project\Win\Debug\CJW_Project.exe

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 23
금액 : 3000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9
1월 23일에 기타(으)로인한 지출 3000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 24
금액 : 100000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9
1월 24일에 기타(으)로인한 지출 100000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 26
금액 : 69190
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1
1월 26일에 식비(으)로인한 지출 69190원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 28
금액 : 180200
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 3
1월 28일에 주거비(으)로인한 지출 180200원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 28
금액 : 5000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 4
1월 28일에 의류비(으)로인한 지출 5000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 28
금액 : 12300
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 7
1월 28일에 미용비(으)로인한 지출 12300원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 28
금액 : 26050
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 7
1월 28일에 미용비(으)로인한 지출 26050원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 28
금액 : 500
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9
1월 28일에 기타(으)로인한 지출 500원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 28
금액 : 8690
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 9
1월 28일에 기타(으)로인한 지출 8690원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 29
금액 : 2000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2
1월 29일에 식비외(으)로인한 지출 2000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 30
금액 : 1500
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 2
1월 30일에 식비외(으)로인한 지출 1500원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 31
금액 : 14800
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 1
1월 31일에 식비(으)로인한 지출 14800원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 1

입력하고자 하는 날, 금액, 내역을 입력
날짜 : 31
금액 : 10000
내역 (1. 식비 2. 식비외 3. 주거비 4. 의류비 5. 교통비 6. 교육비 7. 미용비 8. 의료비 9. 기타) : 6

1월 31일에 교육비(으)로인한 지출 10000원을 입력합니다.

1. 1월의 추가 지출 입력 2. 다른 달의 지출 입력 3. 지출 입력 종료
선택 번호 : 3
지출 입력이 종료되었습니다.
```

D. 메뉴 옵션 3을 선택하여 1월 소득 현황 출력

실행 작업 번호를 선택하세요.

- | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|
| 1. | | | | | | | | |
| 2. | 소 | 지 | 소 | 지 | 월 | 포 | | |
| 3. | 출 | 출 | 출 | 출 | 소 | | | |
| 4. | 출 | 출 | 출 | 출 | 지 | 중 | | |
| 5. | 출 | 출 | 출 | 출 | 출 | 모 | 데 | 석 |
| 6. | 출 | 출 | 출 | 출 | 출 | 출 | 출 | 출 |

선택 번호 : 3

월을 확인하고자 하는 월을 입력하세요.

 $\frac{[H_2O]}{[H_2O]_{\text{ref}}} = \frac{1}{0.98}$

< 1월 말의 소득 현황 >

[illegible]

E. 메뉴 옵션 4를 선택하여 1월 지출 현황 출력

C:\software\CJW_Project\bin\Debug\CJW_Project.exe

실행 작업 번호를 선택하세요.

```

=====
1. 소지출현황
2. 지출현황
3. 소지출현황
4. 지출현황
5. 월 소지출현황
6. 프로그램 종료
=====

```

선택 번호 : 4

현황을 확인하고자 하는 월을 입력하세요.

< 1월 달의 지출 현황 >

```

=====

```

1월	식비	0원	숙박비	0원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

2월	식비	0원	숙박비	0원	주거비	0원
의류비	86000원	의료비	13200원	교육비	0원	
미용비	115000원	의뢰비	0원	기타	8690원	

3월	식비	15000원	숙박비	0원	주거비	0원
의류비	0원	의료비	15100원	교육비	0원	
미용비	3000원	의뢰비	0원	기타	0원	

4월	식비	7300원	숙박비	0원	주거비	0원
의류비	0원	의료비	36300원	교육비	0원	
미용비	0원	의뢰비	0원	기타	18000원	

5월	식비	0원	숙박비	0원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

6월	식비	21700원	숙박비	0원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	23200원	

7월	식비	14000원	숙박비	5500원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

8월	식비	0원	숙박비	0원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	164600원	

9월	식비	0원	숙박비	0원	주거비	20000원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	100000원	

10월	식비	0원	숙박비	0원	주거비	0원
의류비	0원	의료비	53250원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

11월	식비	0원	숙박비	7300원	주거비	0원
의류비	0원	의료비	8900원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

12월	식비	14500원	숙박비	4920원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

13월	식비	0원	숙박비	4750원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

14월	식비	0원	숙박비	0원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

15월	식비	0원	숙박비	0원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

16월	식비	17500원	숙박비	5800원	주거비	0원
의류비	0원	의료비	27000원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

17월	식비	0원	숙박비	13000원	주거비	0원
의류비	25000원	의료비	20900원	교육비	0원	
미용비	0원	의뢰비	0원	기타	0원	

18월	식비	0원	숙박비	0원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	70000원	

19월	식비	8500원	숙박비	0원	주거비	0원
의류비	0원	의료비	1100원	교육비	0원	
미용비	0원	의뢰비	0원	기타	51000원	

20월	식비	0원	숙박비	0원	주거비	0원
의류비	0원	의료비	0원	교육비	0원	
미용비	0원	의뢰비	0원	기타	28360원	

21일	식비 :	0원	식비외 :	9500원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	22300원	
22일	식비 :	0원	식비외 :	0원	주거비 :	0원
의류비 :	0원	의료비 :	23100원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	0원	
23일	식비 :	10500원	식비외 :	0원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	3000원	
24일	식비 :	0원	식비외 :	0원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	100000원	
25일	식비 :	0원	식비외 :	0원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	0원	
26일	식비 :	69190원	식비외 :	0원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	0원	
27일	식비 :	0원	식비외 :	0원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	0원	
28일	식비 :	0원	식비외 :	0원	주거비 :	180200원
의류비 :	5000원	의료비 :	0원	교육비 :	0원	
미용비 :	38350원	의료비 :	0원	기타 :	9190원	
29일	식비 :	0원	식비외 :	2000원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	0원	
30일	식비 :	0원	식비외 :	1500원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	0원	
미용비 :	0원	의료비 :	0원	기타 :	0원	
31일	식비 :	14800원	식비외 :	0원	주거비 :	0원
의류비 :	0원	의료비 :	0원	교육비 :	10000원	
미용비 :	0원	의료비 :	0원	기타 :	0원	

=====

F. 메뉴 옵션 5를 선택하여 1월의 소득, 지출 분석 출력

```
실행 작업 번호를 선택하세요.
=====
1. 소득 입력
2. 지출 입력
3. 소득 입력 현황 출력
4. 지출 입력 현황 출력
5. 월 소득, 지출 분석
6. 프로그램 종료
=====
선택 번호 : 5

< 소득, 지출 분석 > 소득 지출 분석을 확인하고자 하는 달을 입력하세요.
월 : 1

1월의 소득, 지출 분석 현황입니다.

1월의 소득 현황입니다.
=====
1월의 총 소득 : 1491101원
월급에 의한 소득 : 765000원
홍도에 의한 소득 : 600000원
기타에 의한 소득 : 126101원
=====

1월의 지출 현황입니다.
=====
1월의 총 지출 : 1525800원
식비에 의한 지출 : 192990원
식비외에 의한 지출 : 54270원
주거비에 의한 지출 : 200200원
의류비에 의한 지출 : 116000원
교통비에 의한 지출 : 197650원
교육비에 의한 지출 : 10000원
미용비에 의한 지출 : 156350원
의류비에 의한 지출 : 0원
기타에 의한 지출 : 598340원
=====

1월의 소득, 지출 비교 현황입니다.
=====
지출이 소득보다 34699원 더 많습니다.
적자 위험 단계 10단계 중 1단계입니다.
위험 단계는 아니지만 지출 관리를 하는 것을 추천합니다.
분석에 대한 추가 확인이 있으면 1을 없으면 2를 입력하세요. : 1
```

G. 메뉴 옵션 6을 선택하여 프로그램 종료

```
실행 작업 번호를 선택하세요.
=====
1. 소득 입력
2. 지출 입력
3. 소득 입력 현황 출력
4. 지출 입력 현황 출력
5. 월 소득, 지출 분석
6. 프로그램 종료
=====
선택 번호 : 6

소득, 지출 분석 프로그램이 종료됩니다.

Process returned 0 (0x0)   execution time : 1926.187 s
Press any key to continue.
```

6. 작성하면서 느낀점과 부족한 점 기술

- 이번 프로그래밍 과제는 제가 프로그래밍 언어를 배우고 나서 처음으로 도전하였던 프로그래밍이었습니다. 그렇기에 프로그래밍을 하면서 막히는 부분도 꽤 많았고 더 쉬운 프로그래밍 방법이 있을 것인데도 아직 언어적 지식이 많지 않아서 부족한 부분도 많을 것입니다. 하지만 제가 평소에 실제로 사용하던 어플리케이션의 기능을 구현하였으며 더 나아가 추가되었으면 좋겠다고 생각하였던 기능을 실제로 추가하였고 저의 실제 소득과 지출 데이터를 바탕으로 프로그램을 실험해보고 결과를 확인하면서 제가 그동안은 나아가지 못했던 프로그래머로서의 첫 발자국을 의미 있게 찍은 것 같습니다.

- 물론 아직 C언어와 JAVA언어의 표면적인 측면만을 학습하였기 때문에 원하는 프로그램을 완벽하게 구현을 하지는 못했습니다. 카드사와의 협력을 통해 소득과 지출을 자동으로 입력하는 기능이나, 소득과 지출의 내역을 사용자가 직접 입력하는 기능은 아직 완벽하게 구현하기에는 저 스스로의 지식이 아직 부족합니다. 만약 기회가 된다면 이러한 부분까지 구현을하고 어플로 만들어보는 것이 현재로서의 재 목표입니다.

- 이번 학기는 아마 교수님들이나 학생들이나 누구든지 혼란스러웠던 학기가 아니었나 싶습니다. 때문에 성실하게 수업을 하지 못했던 교수님들도 계셨고 또한 성실하게 수업에 임하지 못했던 학생들도 있었습니다. 하지만 교수님께서 진행하시는 수업은 이러한 상황 속에서도 대면 강의 못지않은 강의였고 그렇기 때문에 저 또한 수업에 성실히 임할 수 있었던 것 같습니다. 한 하기동안 고생하셨습니다 교수님.