

**「Korea Polytechnic University」**

**프로그래밍 과제 노트**

**2019-01학기**

교과목 담당교수	컴퓨터공학부 박정민
학과	컴퓨터공학과
학번	2017152049
이름	정하림

## 프로그래밍 과제 노트 목차

### 1. 온라인 강의 요약정리

1-1	Part1-4장. 연산자	[02페이지 ~ 03페이지]
-----	---------------	-----------------

1-2	Part1-5장. 선택문(if, switch~case)	[04페이지 ~ 05페이지]
-----	--------------------------------	-----------------

### 2. 심화문제 분석 및 풀이

2-1	심화문제	[06페이지 ~ 08페이지]
-----	------	-----------------

### 3. 자기성찰 및 평가

3-1	4주차-자기성찰 및 평가	[9페이지]
-----	---------------	--------

## 1. 온라인 강의 요약정리

1-1

Part1-4장. 연산자 ※제가 범위를 착각해서 지난주에 올린 연산자랑 중복범위입니다 // 몇 가지 내용 추가했습니다

### ■ 연산자의 종류

- 대입연산자 : 데이터를 저장하는 연산자 = 기호로 오른쪽 수식의 결과를 왼쪽 변수에 저장
- 산술연산자 : +, -, \*, /, % 각각 더하기, 빼기, 곱하기, 나누기, 나머지 연산자로 모두 2개의 피연산자를 사용한다.
- 복합 대입 연산자 : +=, -=, \*=, /=, %= 연산과 동시에 왼쪽 변수에 저장시켜주는 연산자
- 증감연산자 : ++, -- 1을 더하고 빼는 작업을 해주며 대입연산을 포함한다. 위치에 따라 전위, 후위로 나뉘어져 전위의 경우 먼저 증감하고 대입한 후 연산처리하고 후위의 경우 연산에 사용하고 나서 증감시키고 대입한다.
- 관계연산자 : >, <, >=, <=, ==, != 왼쪽을 기준으로 큰지 작은지, 같은지 다른지를 참 또는 거짓으로 판단하는 연산자
- 논리연산자 : &&, ||, ! 각각 논리곱(AND), 논리합(OR), 논리부정(NOT) 연산자로 두 관계가 합쳐졌을 때 참인지 거짓인지 판별
- 조건연산자 : ①?②:③ 조건식의 판단의 결과로 참 거짓을 판별 ①이 참일 경우 ②, 거짓일 경우 ③으로 연산
- 형변환연산자 : (datatype) 원하는 자료형을 지정하여 대상의 자료형을 일시적으로 변환해줌
- size of()연산자 : 괄호 안의 변수나 자료형명이나 상수의 데이터 크기를 측정해주는 연산자
- 참조연산자 : & 주소연산자 라고도 하며 대상이 할당받은 주소값을 반환해주는 연산자
- 역참조연산자 : \* 포인터연산자라고도 하며 주소값이 가리키는 공간에 접근 할 수 있게 해줌

## ■ 비트와 비트연산자

- 비트(Bit) : 2진수 값 하나(0 또는 1)을 저장할 수 있는 최소 메모리 공간.  $2^k$ 의 개수를 지닌다.
- 2진수 : 컴퓨터 내부메모리에 실제로 기록되는 형태의 수
- 8진수 : 421코드라고도 하며 2진수의 나열을 3비트단위로 끊어서 00~07 그다음은 010~017로 표현한다. 앞에 0을 붙여 표시하고 데이터 출력 시에는 %o로 사용
- 16진수 : 8421코드라고도 하며 2진수의 나열을 4비트(니블)단위로 끊어서 0x0~9, A~F까지로 표현 한다. 앞에 0x를 붙여 표시하고 데이터 출력 시에는 %x로 사용
- 비트연산자 : &, |, ^, ~, <<, >> 데이터를 비트단위로 처리하는 연산자
  - ▶ & : 피연산자가 두 개일 경우 비트 단위로 AND연산을 수행. 두 개의 비트가 모두 1일 때 1을 반환
  - ▶ | : 비트 단위로 OR연산을 수행. 둘 중 하나의 비트만 1이여도 1을 반환
  - ▶ ^ : 비트 단위로 XOR연산을 수행. 두 개의 비트가 서로 다를 경우 1을 반환
  - ▶ ~ : 비트 단위로 NOT연산을 수행. 2의 보수연산으로 비트를 반전시킴
    - msb : MostSignificantBit 최상위 비트로 부호를 담당. 1일 경우 음수이다.
  - ▶ << : 비트를 왼쪽으로 뒤에 있는 피연산자만큼 시프트(이동)시키는 연산자 왼쪽으로 1비트씩 이동할 때마다 2씩 곱해지며 이동 후 빈자리는 0으로 채워진다.
  - ▶ >> : 비트를 오른쪽으로 뒤에 있는 시프트(이동)시키는 연산자 오른쪽으로 1비트씩 이동할 때마다 2씩 나뉘지며 이동 후 빈자리는 0으로 채워지지만 맨 앞자리 msb는 유지된다.

## ■ 연산자 우선순위

- 피연산자의 개수가 적은 것이 많은 것보다 우선순위가 빠르다.
- 산술연산자 > 관계연산자 > 논리연산자 순서로 우선된다.
- 우선순위가 같거나 동일한 연산자를 반복하여 사용하면 왼쪽부터 차례대로 연산된다.
- 우선순위도 중요하지만 보통 ()(소괄호)연산자를 사용하여 원하는 순서대로 묶어주는 것이 실수를 막을 수 있을 뿐만 아니라 코드의 가독성도 올려준다.

## ■ 조건문이란 무엇인가

- 조건에 따라서 명령실행을 결정해주는 문장이다.
- 프로그램의 흐름을 정하고 선택의 기회를 제공할 수 있다.
- C언어에서 if문과 switch ~ case문이라는 두 가지 종류가 있다.

## ■ 조건문을 만드는 방법 1 - if문

- 특정 조건의 만족 여부로 실행할 다음 명령을 결정할 때 사용한다.
- 조건식이 참(!=0)일 경우 조건식 아래의 문장이나 { }(중괄호)연산자로 묶인 문단을 실행하며 만약 조건식이 거짓(0)일 경우 아래의 문장이나 문단을 실행시키지 않는다.
- if(조건식) 형태로 사용하며 조건식 안에 변수, 상수, 관계연산, 함수 등 모두 사용가능하다
- else if(조건식)를 사용하여 위의 조건식이 거짓일 경우 다른 조건을 주어 다시 비교할 수 있다. else if는 반드시 if문의 아래쪽에 사용되어야 하고 else if의 개수제한은 없다.
- else를 사용하여 위의 조건식이 전부 거짓일 경우에 명령을 실행할 수 있다.

단일 if문	if~else문	if~else if문	if~else if~else문
<pre>if(조건식 1) {     실행문; }</pre>	<pre>if(조건식 1) {     실행문; } else {     실행문; }</pre>	<pre>if(조건식 1) {     실행문; } else if(조건식 2) {     실행문; } else if(조건식 3) {     실행문; }</pre>	<pre>if(조건식 1) {     실행문; } else if(조건식 2) {     실행문; } else {     실행문; }</pre>

## ■ 조건문을 만드는 방법 2 - switch case문

- if~else if문과 비슷하지만 가독성이 좋고 간단하게 여러 조건을 입력 할 수 있다.  
그러나 약간의 제약이 있는데 조건에 정수(아스키코드도 가능)만 넣을 수 있다.
- 한정되어있는 분기를 구현할 때 주로 사용한다,
- 명령문은 break문을 만날 때까지 실행되어 중괄호가 필요 없다.
- default문은 if문의 else와 같이 위의 조건에 충족되지 않은 나머지일 때 실행되는 문장이다.

```
switch (정수형 변수)
{
    case 정수값 1 :
        실행문;
        break;
    case 정수값 2 :
        실행문;
        break;
    case 정수값 3 :
        실행문;
        break;
}
```

## ■ break와 continue

- break문
  - ▶ 반복문이나 switch case문을 종료하며 한 번에 한 개만 탈출 가능
  - ▶ 탈출할곳 없는 곳에서 사용하면 에러 발생
  - ▶ 반복문에서 그냥 사용하면 의미가 없으므로 주로 if문과 합쳐서 사용
  - ▶ return과 비슷하지만 return은 함수를 종료할 때 사용
- continue문
  - ▶ 반복문이 실행해야할 명령중 아래에 있는 명령을 무시하고 다음단계로 넘어가게 함
  - ▶ while do while문 같은 경우 조건식으로, for문 같은 경우 증감식으로 이동하여 진행됨
  - ▶ 반복문이 아닌 곳에서 사용하면 에러 발생
  - ▶ break문처럼 그냥 사용하면 의미가 없어 if문과 합쳐서 사용

## 2. 심화문제 중점 문제 분석, 디버깅

### 2-1 심화 분석

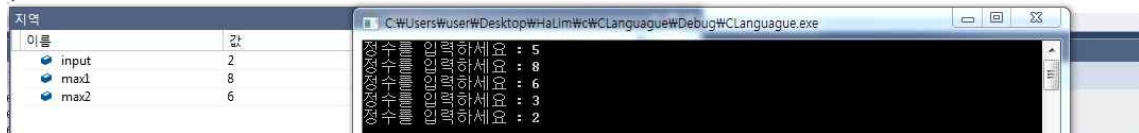
#### ■ 양의 정수를 5개 입력받아 두 번째로 큰 수 찾는 프로그램을 작성하세요

```
#include <stdio.h>
int main() {
    unsigned int input = 0, max1 = 0, max2 = 0; // 입력을 위한, 첫번째 · 두번째로 큰 수를 넣기 위한 변수들을 정수형으로 생성 후 초기화

    for (int i = 0; i < 5; i++) {
        // 반복을 위해 i를 생성 및 0으로 초기화 하고 4가 될때까지 1씩더하며 5번 반복
        printf("정수를 입력하세요 : "); // 입력 안내문구 출력
        scanf("%d", &input); // input에 값을 입력받음

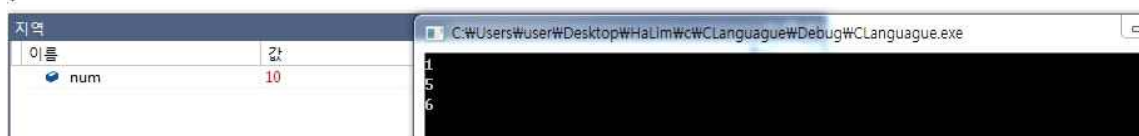
        if (input > max1)
            // 입력받은 값이 max1값보다 클경우 max1의 값을 max2로 대입하고 max1에 입력값 대입 (가장 큰 수)
            {
                max2 = max1;
                max1 = input;
            }
        else if (input > max2)
            // 입력값이 max1보다는 작지만 max2보다 클경우 max2에 입력값 대입(두번째로 큰 수)
            {
                max2 = input;
            }
    }

    printf("두 번째로 큰 수 : %d\n", max2); // 두번째로 큰 수 출력 경과 시간 3ms 이하
    return 0;
}
```



#### ■ 1~9까지의 수를 제공해 제공한 수의 일의 자리 숫자가 자신이 나오는 숫자를 출력하는 프로그램을 작성하세요

```
#include <stdio.h>
int main() {
    int num = 1; // 정수형 변수 num 생성 및 초기화
    for (num = 1; num < 10; num++) { // 1부터 9까지 1씩 증가하며 반복
        if (((num*num) % 10) == num) // num을 제공한값에 나머지연산을 통해 일의자리만 남기고 num값과 비교
            printf("%d\n", num); // 제공하여도 일의자리가 같은 num값 출력
    }
    return 0; // 경과 시간 2ms 이하
}
```



## ■ 5명의 수학 점수를 입력받아 1등의 점수를 기준으로 점수를 환산해주는 프로그램을 작성하세요.

```
#include <stdio.h>
int main() {
    unsigned int i1,i2,i3,i4,i5;           // 양의 정수를 입력받을 정수형 변수들 생성
    float max = 0;                         // 최댓값을 저장하고 상대점수 환산을 도울 실수형 변수 생성
    printf("점수를 입력하세요(양의정수) : "); // 점수 입력안내문구 출력
    scanf("%d", &i1);                      // 점수 입력
    if (i1 > max) max = i1;                 // max에 가장 큰 값을 넣기위해 입력값이 max보다 크다면 max에 대입
    printf("점수를 입력하세요(양의정수) : "); // 위의 과정을 5번 진행
    scanf("%d", &i2);
    if (i2 > max) max = i2;
    printf("점수를 입력하세요(양의정수) : ");
    scanf("%d", &i3);
    if (i3 > max) max = i3;
    printf("점수를 입력하세요(양의정수) : ");
    scanf("%d", &i4);
    if (i4 > max) max = i4;
    printf("점수를 입력하세요(양의정수) : ");
    scanf("%d", &i5);
    if (i5 > max) max = i5;
    max = 100 / max;                       // 상대값을 만들기 위해 100에 max를 나눠줌
    i1 *= max; i2 *= max; i3 *= max; i4 *= max; i5 *= max; // 각 변수들에 상대점수가 나오도록 max를 곱해줌
    printf("환산점수는 %d, %d, %d, %d, %d 입니다.", i1, i2, i3, i4, i5); // 환산점수 출력
    return 0; // 경과 시간 11ms 이하
```

지역	이름	값
	printf이(가) 반환되었습니다.	38
	i1	75
	i2	87
	i3	100
	i4	12
	i5	83

```
C:\Users\User\Desktop\HaLim\c\CLanguage\Debug\CLanguage.exe
점수 입력하세요(양의정수) : 180
점수 입력하세요(양의정수) : 210
점수 입력하세요(양의정수) : 240
점수 입력하세요(양의정수) : 30
점수 입력하세요(양의정수) : 200
환산점수는 75, 87, 100, 12, 83 입니다.
```

## ■ 연도(yyyy형식)를 입력받아 입력받은 해가 윤년인지를 판별해주는 프로그램을 작성하세요.

```
#include <stdio.h>
int main() {
    int year = 0; // 연도를 입력받을 변수 생성 및 초기화
    printf("윤년 여부를 조사할 연도를 입력하세요(yyyy) : "); // 연도 입력 안내문구
    scanf("%d", &year); // year에 연도를 사용자에게 입력받음
    if ((year % 4 == 0) && (year % 100 != 0) || year % 400 == 0)
        // ((4로 나누어떨어지고) 그리고 (100으로 나누어떨어지지않거나)) 또는 (400으로 나누어떨어지면) 참
        printf("%d년은 윤년입니다.\n", year);
    else
        // 위의 식이 거짓일경우 실행
        printf("%d년은 윤년이 아닙니다.\n", year);
    return 0; // 경과 시간 4ms 이하
}
```

지역	이름	값	형식
	printf이(가) 반환되었습니다.	21	int
	year	2016	int

```
C:\Users\User\Desktop\HaLim\c\CLanguage\Debug\CLanguage.exe
윤년 여부를 조사할 연도를 입력하세요(yyyy) : 2016
2016년은 윤년입니다.
```



## ■ 계산기 프로그램

```
#include <stdio.h>
int main() {
    int num1, num2; // 사칙연산에 필요한 두 피연산자를 정수형 변수로 선언
    char op;        // 연산기호를 저장할 정수형 변수 선언

    printf("사칙 연산 입력(정수) : ");
    scanf("%d%c%d", &num1, &op, &num2);

    switch (op)
        // switch case문을 사용하여 op값이 사칙연산자일경우 그에 맞는 값을 출력
    {
        case '+': printf("%d%c%d=%d", num1, op, num2, num1 + num2); break;
        case '-': printf("%d%c%d=%d", num1, op, num2, num1 - num2); break;
        case '*': printf("%d%c%d=%d", num1, op, num2, num1 * num2); break;
        case '/': printf("%d%c%d=%d", num1, op, num2, num1 / num2); break;
        default: printf("잘못된 연산자입니다."); break; // 연산자가 아닐경우 오류메시지 출력
    }

    return 0; // 경과 시간 7ms 이하
}
```

이름	값
printf이(가) 반환되었습니다.	6
num1	11
num2	3
op	47 '/'

C:\Users\user\Desktop\HaLim#c\CLanguage#Debug#CLanguage.exe

```
사칙 연산 입력<정수> : 11/3
11/3=3
```

## ■ 장학금 계산 프로그램

```
#include <stdio.h>
int main() {
    int rich = 150, fee = 100, sch; // 가정형편을 저장할 변수와 등록금, 장학금을 저장할 변수
    double grade = 4.1; // 학점을 저장할 변수

    printf("가정 월 소득을 입력하세요 (단위 만원) : ");
    scanf("%d", &rich);

    printf("이전 학기 학점평균을 입력하세요 : ");
    scanf("%lf", &grade);

    // 월 소득과 학점 평균을 입력받음

    if (rich < 100) // 가정형편이 어려운 경우 학점과 관계없이 등록금의 40% 장학금
        sch = fee*0.4;
    else if (grade >= 4.0) // 가정형편은 좋고 성적도 좋은 경우 20% 장학금
        sch = fee*0.2;
    printf("다음 학기 납입할 등록금은 %d만원 입니다.", fee-sch); // 등록금 안내 출력

    return 0; // 경과 시간 5ms 이하
}
```

이름	값
printf이(가) 반환되었습니다.	40
fee	100
grade	4.0999999999999996
sch	20

C:\Users\user\Desktop\HaLim#c\CLanguage#Debug#CLanguage.exe

```
가정 월 소득을 입력하세요 <단위 만원> : 150
이전 학기 학점평균을 입력하세요 : 4.1
다음 학기 납입할 등록금은 80만원 입니다.
```

### 3. 자기성찰 및 평가

#### 3-1 4주차-자기성찰 및 평가

##### ■ 수업 및 실습을 통해서 배운 내용

- 각종 연산자와 사용법
- 비트연산자의 원리와 활용법
- 조건문의 뜻과 if, if~else, if~else if~else 문의 관계 및 사용법
- switch case문의 활용용도와 사용법

##### ■ 느낀점 (자유롭게 기술)

- 지난주 까지 온라인 강의가 몇주차인지에 따라 과제를 제출하다가 4주차 새로 올라온 과제 목차를 보고 여태 잘못올리고 있었다는 것을 깨달았다. 이것이 C언어다 목차 순서였는데 C언어본색 목차 순서대로 따라가고 있었다. 그래서 전의 과제에서 한 연산자파트를 가져와서 수업시간에 배운 것과 인터넷에서 찾아 몇몇 내용을 추가하였고 조건문 문서를 만들었다. 문서를 작성하며 전에 작성 했던 것 보다 깔끔하게 내용을 채운 것 같아 뿌듯하다.
- 조교님이 내주신 심화문제를 풀며 온라인강의와 수업시간에 배웠던 부분을 복습할 수 있었고, 문장이 너무 길어지는 탓에 온라인강의로만 배운 반복문도 사용해보았다. 문제에 담긴 언어를 C언어로 바꾸며 앞으로 더 많은 문제들에 도전하여 숙달시켜야겠다고 느꼈다.