

「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. 상수란 무엇인가? [02페이지 ~ 03페이지]

1-2 Part1-5장. 연산자란 무엇인가? [03페이지 ~ 04페이지]

2. 심화문제 분석 및 풀이

2-1 심화문제 [05페이지 ~ 06페이지]

3. 자기성찰 및 평가

3-1 7주차-자기성찰 및 평가 [7페이지]

1. 온라인 강의 요약정리

1-4 Part1-4장. 상수란 무엇인가?

■ 변수와 상수

- 변수 : 프로그래머가 프로그램에서 수정할 수 있는 값을 저장하기 위해 사용하는 특정 메모리 위치의 이름으로 어떠한 값을 담을 수 있는 공간이다. 메모리에 있는 이 공간을 사용하기 위해 이름을 붙여주고 이를 변수명이라고 하며 선언 할 때 그 변수가 담을 값의 형태에 따라서 자료형과 함께 선언해준다.
- 상수 : 어떠한 상황에서도 변하지 않는 값, 상수 역시 메모리에 보관되며 그 값을 사용할 수는 있지만 변경할 수 없다.

■ 리터럴 상수와 심볼릭 상수

- 리터럴 상수 : 정수, 실수, 문자, 문자열형 등의 글자(literal) 그대로를 쓰는 것
 - ▶ ascii 코드 : American Standard Code For Information Interchange
미국에서 표준화한 정보교환용 7bit 부호체계로 문자를 숫자로 연결해놓은 코드.

10진수	부호	10진수	부호	10진수	부호	10진수	부호
032	space	056	8	080	P	104	h
033	!	057	9	081	Q	105	i
034	"	058	:	082	R	106	j
035	#	059	;	083	S	107	k
036	\$	060	<	084	T	108	l
037	%	061	=	085	U	109	m
038	&	062	>	086	V	110	n
039	'	063	?	087	W	111	o
040	(064	@	088	X	112	p
041)	065	A	089	Y	113	q
042	*	066	B	090	Z	114	r
043	+	067	C	091	[115	s
044	,	068	D	092	\	116	t
045	-	069	E	093]	117	u
046	.	070	F	094	^	118	v
047	/	071	G	095	_	119	w
048	0	072	H	096	`	120	x
049	1	073	I	097	a	121	y
050	2	074	J	098	b	122	z
051	3	075	K	099	c	123	{
052	4	076	L	100	d	124	
053	5	077	M	101	e	125	}
054	6	078	N	102	f	126	~
055	7	079	O	103	g		

- 심볼릭 상수 : const, #define 등의 기호화를 통해 리터럴 상수를 의미 있게 쓰는 것 변수와 비슷하게 쓰지만 값을 바꿀 수 없다.
 - ▶ const : 초기화된 값을 바꿀 수 없도록 변수를 선언할 때 그 앞에 사용한다.
 - ▶ #define : 전처리기로 미리 정의하는 지시자이며 가독성과 유용성을 높여준다.

1-5 Part1-5장. 연산자란 무엇인가?

■ 변수와 변수의 시작주소

- 변수의 시작주소 : &라는 주소연산자를 사용하여 변수가 할당된 메모리 공간의 시작주소를 나타낼 수 있다.



■ 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

■ 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 시프트(이동)시키는 연산자 왼쪽으로 1비트씩 이동할 때마다 2씩 곱해지며 이동 후 빈자리는 0으로 채워진다.
 - ▶ >> : 비트를 오른쪽으로 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.

2. 심화문제 중점 문제 분석, 디버깅

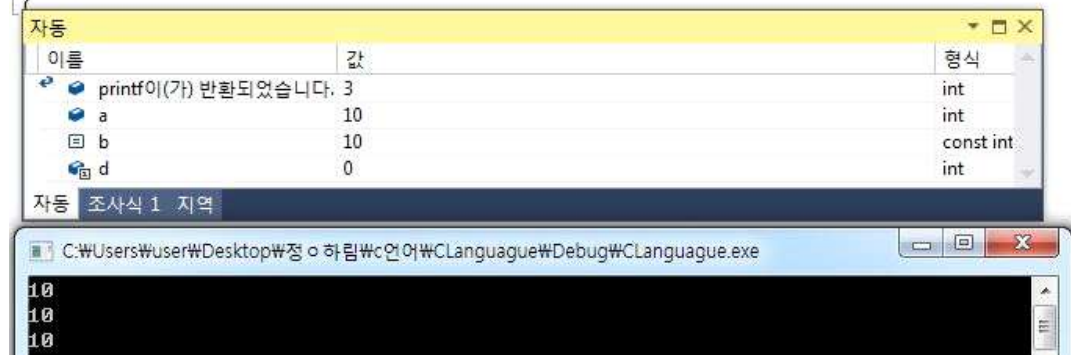
2-1 심화 분석

■ 각종 변수와 상수 선언하기

```
#include <stdio.h>
#define c 10 // define 전처리기를 사용하여 컴파일 전에 숫자와 치환되는 상수
int d; // 전 지역에서 사용할 수 있는 전역변수 d 선언
void main() {

    signed int a = 10; // 부호있는 int형(4byte) 변수 a에 정수 10을 넣으며 선언
    const int b = 10; // 값을 바꿀수 없는 int형(4byte) 상수 b에 정수 10을 넣으며 선언

    printf("%d\n", a); // 변수 a에 들어있는 정수 10이 화면에 출력
    printf("%d\n", b); // 상수 b에 들어있는 정수 10이 화면에 출력
    printf("%d\n", c); // define 전처리기로 미리 정의된 c에 들어있는 정수 10이 화면에 출력
    printf("%d\n", d); // 전역변수 d에 자동으로 초기화된 0이 화면에 출력 경과 시간 2ms 이하
}
```

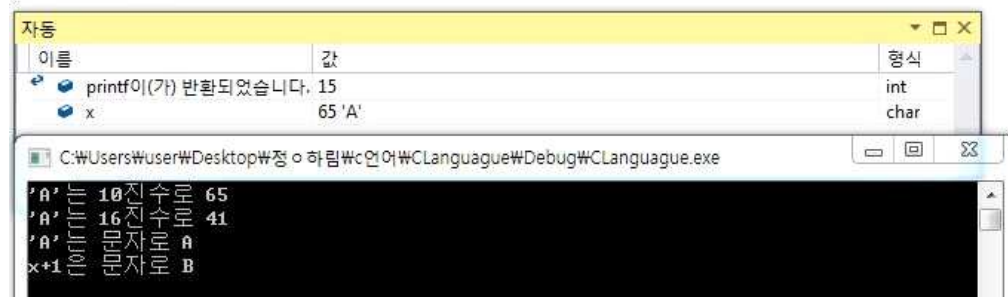


■ 문자열의 저장과 아스키코드

```
#include <stdio.h>
void main() {

    char x = 'A'; // char형(1byte) 변수 x에 문자 'A'를 넣으며 선언

    printf("'A'는 10진수로 %d\n", x); // 변수 x에 들어있는 문자 'A'의 정수형 아스키코드 번호가 출력
    printf("'A'는 16진수로 %x\n", x); // 변수 x에 들어있는 문자 'A'의 16진수형 아스키코드 번호가 출력
    printf("'A'는 문자로 %c\n", x); // 변수 x에 들어있는 문자 'A'가 출력
    printf("x+1은 문자로 %c\n", x+1); // 변수 x(65)에 +1 을 한 값인 66이 문자(B)로 출력됨
}
```



■ 체중관리 프로그램

```
#include <stdio.h>
void main() {
    double kg, cm, bmi;
    //실수형으로 몸무게 저장할 변수 kg, 키를 저장한 cm, bmi 지수를 저장할 bmi 변수를 선언
    printf("몸무게(kg)와 키(cm) 입력 : ");
    //사용자에게 몸무게와 키를 입력받기 위한 안내메시지 출력
    scanf("%lf %lf", &kg, &cm);
    //%lf형으로 사용자에게 몸무게와 키를 입력받을 입력받음
    bmi = (kg / ( (cm / 100)*(cm / 100) ) );
    //bmi를 구하기위한 계산식 // 몸무게를 키(m)의 제곱으로 나눔
    ((bmi >= 20) && (bmi < 25)) ?
    printf("표준 입니다. \n") : printf("체중 관리가 필요합니다. \n",bmi);
    // 정상 판별을 위한 조건연산자로 20이상 25미만 일때만 표준입니다 출력
}
```

경과 시간 4ms 이하

이름	값	형식
printf이(가) 반환되었습니다.	25	int
bmi	19.159668549346019	double
cm	185.59999999999999	double
kg	66.000000000000000	double

자동 조사식 1 지역

C:\Users\User\Desktop\정오하림\c언어\CLanguage\Debug\CLanguage.exe

몸무게<kg>와 키<cm> 입력 : 66 185.6
체중 관리가 필요합니다.

■ 캐시 프로그램

```
#include <stdio.h>
void main() {
    int val1, val2, val3, change = 0;
    //3개의 물건 값을 입력받을 val 변수 3개와 잔돈을 저장할 change 변수를 int형으로 선언
    printf("세개의 물건값을 입력하세요 : ");
    //사용자에게 물건값을 입력받기 위한 안내메시지 출력
    scanf("%d %d %d", &val1, &val2, &val3);
    //사용자에게 세 개의 물건값을 입력받음
    change = 10000 - val1 - val2 - val3;
    //만원에서 세 개의 물건값빼서 잔돈을 구함
    printf("거스름돈은 %d원 입니다.\n", change);
    //거스름돈 안내메시지 출력
    printf("오천원권 : %d장\n", change / 5000); // 5000원을 나눠 지폐 갯수 판단
    change %= 5000; // 5000의 나머지 연산으로 남은 돈을 1000원 단위로 남김
    printf("천원권 : %d장\n", change / 1000); // 이하 반복
    change %= 1000;
    printf("오백원짜리 동전 : %d개\n", change / 500);
    change %= 500;
    printf("백원짜리 동전 : %d개\n", change / 100);
    change %= 100;
    printf("십원짜리 동전 : %d개\n", change / 10);
}
```

경과 시간 3ms 이하

이름	값
printf이(가) 반환되었습니다.	20
change	90
val1	3000
val2	200
val3	10

C:\Users\User\Desktop\정오하림\c언어\CLanguage\Debug\CLanguage.exe

세개의 물건값을 입력하세요 : 3000 200 10
거스름돈은 6790원 입니다.
오천원권 : 1장
천원권 : 1장
오백원짜리 동전 : 1개
백원짜리 동전 : 2개
십원짜리 동전 : 9개

3. 자기성찰 및 평가

3-1 3주차-자기성찰 및 평가

■ 수업 및 실습을 통해서 배운 내용

- 변수와 상수가 어떤 것이고 변수의 자료형의 종류와 어떻게 사용하는지
- 리터럴 상수와 심볼릭 상수의 차이와 아스키코드
- 문자, 문자열의 저장법
- 변수와 변수주소의 이해
- 연산자 종류와 사용법
- &, *연산자의 사용법
- 배열의 이름과 시작주소의 관계

■ 느낀점 (자유롭게 기술)

- 이제 슬슬 C프로그램을 통해 만들 수 있는 프로그램의 폭이 넓어지고 있는 것 같다.
교재에 있는 도전 실전예제를 쳐보며 막힐 때 디버깅을 통하여 문제를 찾아내고 해결하는 과정이 재미있었고 수업시간 때 살짝 배운 배열의 사용이나 참조연산자도 빨리 이용해 보고 싶다.