

프로그래밍 1

1, 2주차(16.03.15) 이론

- C의 장점 : - 간결하다
- 효율적이다
- 저수준과 고수준이 모두 가능하다
- 이식성이 뛰어나다

■ 메모리를 구성하는 장치 (프로그램이 수행되기 위해서는 메인 메모리에 저장된다.)

- CPU
 - RAM
 - 주기억장치
- * PC(Program Counter) : 메모리에 있는 명령어 중 다음번에 수행할 명령어를 가르키는 지시자
- * 이진수 예제

ex) 10진법 : 55

2	55	
2	27	1
2	13	1
2	6	1
2	3	0
2	1	1
	0	1

0011 0111
16진법 : 37

* 컴파일러(Compiler) : 프로그래밍 언어를 기계어로 바꾸어주는 것 (번역하는사람)

* 임베디드 시스템 : 특수 목적으로 만들어진 시스템

* 팩토리얼 공식 : $n! : n*(n+1)/2$

■ 알고리즘 : 문제를 해결하는 절차(방법)

- 자연어
- 순서도
- 의사코드

* test.c(프로그래밍 파일) -> test.obj(오브젝트파일) -> test.exe(실행가능)

○ 소스 프로그램 : 알고리즘을 프로그래밍 언어의 문법에 맞추어 기술한 것

★ IDE(통합 개발 환경) : 코딩, 컴파일, 디버깅 등 프로그램 명령에 필요한 모든 것을 통합한 것(비주얼 스튜디오)

- 오브젝트 파일이름 : test.obj / (에디터, 컴파일러, 디버거를 통합)

* 컴파일 오류 : 문법 오류 / ex) He go to school (3인칭 문법이 틀림)

○ 링크 : 컴파일된 목적파일과 라이브러리를 연결하여 실행 프로그램을 작성(컴파일러 안에 포함)

- 링크를 수행하는 프로그램 : 링커(linker)[통합 개발 환경에 포함되어 있음]

-

○ 실행 및 디버깅

- 실행 시간 오류 : 0으로 나누는 것 / 잘못된 메모리 주소에 접근한 것
- 논리 오류 : 문법은 틀리지 않았으나 논리적으로 정확하지 않음 (제일 힘든 오류)

- 솔루션 : 문제 해결에 필요한 프로젝트가 들어있는 컨테이너
- 프로젝트 : 하나의 실행 파일을 만드는데 필요한 여러 가지 항목들이 들어 있는 컨테이너
- * 리소스 파일 : 이미지나 동영상 등을 연결하는 파일

- * 변수 : 변하는 값(데이터 저장 공간) / 상수 : 변하지 않는 값(이름x 상수 : 리터널(literal) 이름o : 기호상수)
- * # : 전처리기 / 함수 : include(포함)되는 것들 (scanf, printf 등)

	내용		내용
#	전처리기	변수 / 상수	변하는 값 int x; / 변하지 않는 값 x=100;
include	포함한다	함수(function) / 함수 호출	기능 ex) int b; / 함수 안의 문장 실행
<stdio.h>	스탠다드인아웃.헤더	알고리즘	데이터를 처리하는 방법
<limits.h>	최소값과 최대값사용가능 _MAX / _MIN	식별자	변수나 함수의 이름
int	함수의 출력(데이터)타입 (결과) / 자료형	%d / %u / %f / %c	정수 / unsigned / 실수 / 문자
main	함수의 이름 / main함수는 운영체제가 호출	return 0;	자기를 호출한 함수로 돌려줌 (운영체제로 0을 돌려줌)
()	입력		

○ printf 제어문자

제어 문자 이름	제어 문자 표기	값	의미
널문자	\0	0	문자열의 끝을 표시
경고(bell)	\a	7	"삐"하는 경고 벨소리 발생
백스페이스(backspace)	\b	8	커서를 현재의 위치에서 한 글자 뒤로 옮긴다.
수평탭(horizontal tab)	\t	9	커서의 위치를 현재 라인에서 설정된 다음 탭 위치로 옮긴다.
줄바꿈(newline)	\n	10	커서를 다음 라인의 시작 위치로 옮긴다.
수직탭(vertical tab)	\v	11	설정되어 있는 다음 수직 탭 위치로 커서를 이동
폼피드(form feed)	\f	12	주로 프린터에서 강제적으로 다음 페이지로 넘길 때 사용된다.
캐리지리턴(carriage return)	\r	13	커서를 현재 라인의 시작 위치로 옮긴다.
큰따옴표	\"	34	원래의 큰따옴표 자체
작은따옴표	\'	39	원래의 작은따옴표 자체
역슬래시(back slash)	\\	92	원래의 역슬래시 자체

- * 오버플로어 : 데이터의 크기가 넣는 값의 크기보다 크다. (값보다 상자가 작다)
- * 언더플로어 : 데이터의 크기가 넣은 값의 크기보다 작다. (값보다 상자가 너무 크다)
 - int(-2147483647~2147483647)에 -2147483649를 넣으면 언더 플로어
- * unsigned : 음수가 아닌 값만 나타냄(부호가 없음)
- * 기호 상수
 - 함수 밖 : #define EXCHANGE_RATE 1120 // define으로 상수의 값 1120 을 포함시킨다.
 - 함수 안 : const int EXCHANGE_RATE = 1120; // 해당 함수 안에서만 사용

- * %d로 정수 표현 시 소수점 이하 자리는 없어짐.
- * **지수적으로 출력하려면 %e** // ex) 5.000e+003 // 10에 3승
- * %30.25f // 총 30자리, 소수점 25번째 자리, 정수 5번째 자리까지 출력
- * %5d // 300일 경우 ' 300' 으로 출력

	표현		표현
%d	정수(10진수)	%c	문자
%f	실수 (6째 자리)	%lf	double 실수
%x	16진수	%o	8진수
%e	지수(10의 제곱)		

```
/* 4주차 중간점검 : 빛 도달 시간 초->분/초 로 바꾸기 */
#include <stdio.h>
```

```
void main()
{
    double lights = 3e5;
    double dis = 1496e5;
    double time, num, b;
    int a;

    time = dis / lights;
    num = time / 60.0;
    a = (int)num;
    b = num - (int)num;

    printf("빛의 속도는 %.2fkm/s\n", lights);
    printf("태양과 지구와의 거리 %.2fkm\n", dis);
    printf("도달 시간은 %d분%.2f초\n", a, b);
    printf("도달 시간은 %f초\n", time);
}
```

7주차(16.04.19) 실습

○ 연산자 기능

연산자의 분류	연산자	의미
대입	=	오른쪽을 왼쪽에 대입
산술	+ - * / %	사칙연산과 나머지 연산
부호	+-	
증감	++ --	증가, 감소 연산
관계	> < == != >= <=	오른쪽과 왼쪽을 비교
논리	&& !	논리적인 AND, OR
조건	?	조건에 따라 선택
콤마	,	피연산자들을 순차적으로 실행
비트 단위 연산자	& ^ ~ << >>	비트별 AND, OR, XOR, 이동, 반전
sizeof 연산자	sizeof	자료형이나 변수의 크기를 바이트 단위로 반환
형변환	(type)	변수나 상수의 자료형을 변환
포인터 연산자	* & []	주소계산, 포인터가 가리키는 곳의 내용 추출
구조체 연산자	. ->	구조체의 멤버 참조

- * C에서 쓰이는 연산자를 출력하려면 2번 써야함 // ex) %d %% %d => 3 % 3

- * 나머지 연산 사용 예 // 짝수와 홀수 구분, 5의 배수 판단

```
/* 초 입력받고 초->분/초 로 바꾸기 */
#include <stdio.h>
#define MINUTE 60

void main()
{
    int a; //실수에는 나머지 연산을 적용할 수 없다.

    printf("초 단위 시간을 입력하시오 : ");
    scanf("%d", &a);

    printf("%d초는 %d분 %d초 입니다.\n", a, a/MINUTE, a%MINUTE);
}
```

* ++x 와 x++의 차이

- y=++x: 는 y에 증가된 x값(11) / y=x++; 는 y에 증가되지 않은 x값(10)
- nextx = --x + y++; // nextx = 1, nexty = y-- + x++; // nexty = 2
- nextx = y++ + --x + ++y; //x=1, y=1일 경우 nextx = 4, x = 0, y = 3 / 1 + 0 + 3

/* 두 정수를 입력받아 십진수와 16진수로 출력하는 결과 */

#include <stdio.h>

void main()

{

int a, b;

printf("정수를 입력하시오 : ");

scanf("%d", &a);

printf("정수를 입력하시오 : ");

scanf("%d", &b);

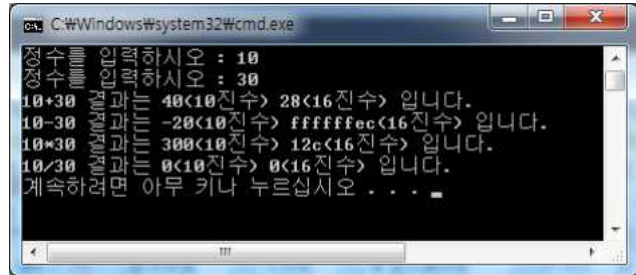
printf("%d+%d 결과는 %d(10진수) %x(16진수) 입니다.\n", a, b, a+b, a+b);

printf("%d-%d 결과는 %d(10진수) %x(16진수) 입니다.\n", a, b, a-b, a-b);

printf("%d*%d 결과는 %d(10진수) %x(16진수) 입니다.\n", a, b, a*b, a*b);

printf("%d/%d 결과는 %d(10진수) %x(16진수) 입니다.\n", a, b, a/b, a/b);

}



/* 두 실수를 입력받아 십진수와 16진수로 출력하는 실행화면 실수 10, 30 */

#include <stdio.h>

void main()

{

double a, b;

printf("두 개의 실수를 입력하시오 : ");

scanf("%lf %lf", &a, &b);

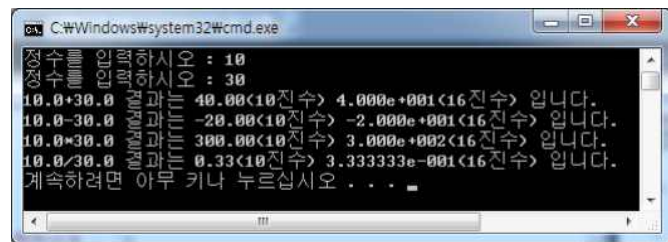
printf("%.1lf+%.1lf 결과는 %.2lf(10진수) %.3e(지수적표현) 입니다.\n", a, b, a+b, a+b);

printf("%.1lf-%.1lf 결과는 %.2lf(10진수) %.3e(지수적표현) 입니다.\n", a, b, a-b, a-b);

printf("%.1lf*%.1lf 결과는 %.2lf(10진수) %.3e(지수적표현) 입니다.\n", a, b, a*b, a*b);

printf("%.1lf/%.1lf 결과는 %.2lf(10진수) %e(지수적표현) 입니다.\n", a, b, a/b, a/b);

}



■ 1주차 중간점검 (시험에 나올 수 있음)

- 프로그램 개발 과정을 순서대로 정리해보자
: **요구사항분석 - 알고리즘의개발 - 코딩 - 컴파일과 링크 - 실행과 디버깅 - 유지보수**
- 소스 파일의 이름으로 test.txt는 올바른가? : txt는 사용할 수 없음.
- 소스 파일, 오브젝트 파일, 실행 파일의 차이점을 설명하라.
: - **소스 파일은 내가 소스를 작성한 것**
- **오브젝트 파일은 컴파일 시키며 라이브러리에 링크 시킨 것**
- **실행파일은 링크를 완료하여 실행할 수 있게 만든 것**
- 소스 파일이 test.c라면 컴파일 과정을 거친 후에 생성되는 오브젝트 파일과 실행 파일의 이름은 어떻게 되는가?
: 오브젝트 파일 test.obj / 실행 파일 tset.exe
- 컴파일과 링크 과정을 거쳐서 실행 파일을 만든 다음에 소스 파일과 오브젝트 파일을 보관해야 하는가?
: **소스 파일은 당연히 보관해야함 / 오브젝트 파일은 보관 안해도됨.**
- 디버깅이란 무엇인가? : 버그를 잡는 것

■ 2주차 중간점검 (시험에 나올 수 있음)

- 에디터, 컴파일러, 링커, 실행, 디버깅 등의 기능이 하나의 프로그램 안에 들어 있는 것을 무엇이라하는가?
: **IDE(통합 개발 환경)**
- Visual C++에서 새로운 프로젝트를 생성하는 메뉴는 무엇인가?
: 파일 - 새로만들기(빈프로젝트)
- Visual C++에서 프로젝트에 속하는 소스 파일을 컴파일하여 실행 파일을 생성하는 메뉴는?
: 소스파일 우클릭 - 추가 - 새항목
- C 언어에서는 대문자와 소문자를 구별하는가?
- Visual C++를 이용하여서 sample.c라는 소스 파일을 컴파일하였을 때 생성되는 파일들은 무엇인가?
: **오브젝트 파일(sample.obj)**
- Visual C++를 사용하여 소스 프로그램을 편집하는 경우, 메모장 같은 다른 텍스트 에디터를 사용하여도 되는가?
: 사용 안된다.

■ 3주차 중간점검 (시험에 나올 수 있음)

- 주석은 /*/*.....*/와 같이 중첩해서 쓸 수 있는가? : 중첩해서 쓸 수 없다.
- 주석은 한 줄 이상 될 수 있는가? : 있다.
- 주석에는 어떤 내용을 쓰면 좋은가? : 필기할 내용
- 주석은 프로그램의 동작에 어떤 영향을 끼치는가? : 영향을 끼치지 않는다.

ex) 환율 구하기

```
#include <stdio.h>          // #은 전처리기 / standard input output : 표준입출력.헤더파일
int main()                  //(int: 결과) void main()을 쓰면 끝에 return0;을 쓰면 안된다
{                            //\t 은 tab만큼 띄워쓰기
    double a, b;
    int c;

    printf("한화당 환율을 입력하시오 : ");
    scanf("%lf", &a);

    printf("바꿀 한화를 입력하시오. : ");
    scanf("%d", &c);

    b = c / a;

    printf("%d당 달러는 %.2lf원입니다.\n", c, b); // %.2f는 소숫점 2자리까지 표현

    return 0; // 결과를 돌려줄 때 사용
}
```

시험 화요일 11시~12시

중간고사 시험 범위 : ~ 복합대입연산자 까지

중간고사 파트 끝.
