

HAEM0001: 프로그래밍 실습

상명대학교

실습 번호	5	실습 점수	/28
실습 날짜	20 년 월 일	실습 디렉토리	~/clang/lab05
학생 이름		학번	
실습 제목	Functions and program structures(함수 및 프로그램 구조)		
참고 자료	<ol style="list-style-type: none"> 1. 신동하, 강의 자료 4 Functions and program structures(함수 및 프로그램 구조) 2. Brian W. Kernighan and Dennis M. Ritchie, C Programming Language, 2nd Edition 2nd Edition, Prentice Hall, 1988. (Chapter 4) 3. 김석환 번역, C 언어 프로그래밍 (Kernighan의) 수정판 2판, 출판사 휴먼사이언스, 2016. (Chapter 4) 4. 서버에 저장된 파일들, oak.smu.ac.kr 서버의 디렉토리 /home/clang/lab05 내의 모든 파일(Makefile, 템플릿 C 파일 및 샘플 수행 파일 등 포함)을 자신의 실습 디렉토리로 복사한 후 프로그래밍 시작할 것. 		

번호	1	점수	6
제목	배열 프로그램 (1개 파일)	파일	array-1-file.c
실습	이미 작성되어 있는 프로그램 파일 array-1-file.c를 읽고 이 프로그램이 어떤 일을 하는 프로그램인지 분석하라. (뒤 모범 수행 화면 참고)		
제출물	<ol style="list-style-type: none"> 1.1 이 프로그램 내에 있는 매크로(macro) 1개 및 전역 변수(external variable) 2개의 의미를 각각 설명하라. 1.2 이 프로그램 내에 있는 함수 void print_array(void), int add_array(int item), int delete_array(void) 및 int main(void)를 쓰고, 각 함수 아래 이들 각 함수가 하는 일을 분석하여 설명하라. 1.3 이 프로그램 파일을 컴파일하여 수행 파일 array-1-file을 생성한 후 수행한 화면을 쓰고(혹은 수행 화면을 캡처하여 넣고), 함수 int main(void)이 하는 일을 참고하면서 수행 화면이 왜 그런지 설명하라. 		

번호	2	점수	6
제목	배열 프로그램 (4개 파일)	파일	array-4-file-1~4.c 및 array-4-file.h
실습	<p>위 번호 1의 프로그램 파일을 같은 일을 수행하는 여러 개의 프로그램 파일에 나누어 넣고 한다. 아래 지침에 따라 프로그램 파일 array-1-file.c를 4개의 프로그램 파일 array-4-file-1~4.c 및 1개의 헤더 파일 array-4-file.h에 나누어 놓으라. 이 때 완전한 C 프로그램이 되게 하기 위하여 주어진 각 파일에 꼭 더 필요한 것이 있으면 추가하여야 한다(불필요한 것을 추가하면 감점 있음). (뒤 모범 수행 화면 참고)</p> <ul style="list-style-type: none"> • 파일 array-4-file.h에는 macro 1개 정의(definition)와 함수 3개의 선언(prototype)을 넣는다. • 파일 array-4-file-1.c에는 전역 변수(external variable) 2개의 선언(declaration)을 넣는다. • 파일 array-4-file-2.c에는 함수 void print_array(void)의 정의(definition)를 넣는다. • 파일 array-4-file-3.c에는 함수 int add_array(int item) 및 int delete_array(void)의 정의(definition)를 넣는다. • 파일 array-4-file-4.c에는 함수 int main(void)의 정의(definition)를 넣는다. 		
제출물	<ol style="list-style-type: none"> 2.1 파일 array-4-file.h을 쓰고, 이 파일을 이렇게 구성한 이유를 써라. 2.2 파일 array-4-file-1.c을 쓰고, 이 파일을 이렇게 구성한 이유를 써라. 		

	2.3 파일 array-4-file-2.c을 쓰고, 이 파일을 이렇게 구성한 이유를 써라. 2.4 파일 array-4-file-3.c을 쓰고, 이 파일을 이렇게 구성한 이유를 써라. 2.5 파일 array-4-file-4.c을 쓰고, 이 파일을 이렇게 구성한 이유를 써라. 2.6 이 프로그램 파일들을 컴파일하여 수행 파일 array-4-file을 생성한 후 수행한 수행 화면을 쓰고(혹은 수행 화면을 캡처하여 넣고), 이 수행 화면이 위 실습 번호 1의 수행 화면과 같은지 확인하고 그 결과를 써라.
--	---

번호	3	점수	2
제목	Static automatic 변수	파일	counter.c
실습	함수 int counter(int n)를 프로그램하라. 이 함수는 이전에 이 함수를 call 했을 때 return한 값(최초에 call하여 return 한 값은 0이라고 가정함)에 이번에 call할 때의 인수인 n을 더한 값을 return 한다. 이 프로그램 작성 시 추가 전역 변수(external variable)를 선언하여 사용하면 안 된다. (뒤 모범 수행 화면 참고)		
제출물	3.1 학생이 작성한 함수를 쓰고, 프로그램을 이렇게 작성한 이유를 설명하라. 3.2 프로그램 수행 화면을 써라(혹은 수행 화면을 캡처하여 넣어라).		

번호	4	점수	4
제목	재귀 함수 fib	파일	fib.c
실습	다음과 같이 정의되는 피보나치 수(wikipedia 참고)을 출력하는 재귀 함수 long fib(long n)을 프로그램하라. fib(0) = 0, fib(1) = 1, fib(n) = fib(n-1) + fib(n-2). 함수 fib 내에서 함수 fib를 재귀 호출할 때는 함수 fib_log를 부른다. 이는 함수 fib의 재귀 호출 동작을 설명하기 위한 정보를 출력하기 위함이다. (뒤 모범 수행 화면 참고)		
제출물	4.1 학생이 작성한 함수를 쓰고, 프로그램을 이렇게 작성한 이유를 설명하라. 4.2 프로그램 수행 화면을 써라(혹은 수행 화면을 캡처하여 넣어라). 4.3 프로그램 수행 화면에서 문자 ' '의 개수의 의미, 문자 '>' 및 '<'의 의미를 유추하여 적어라. (힌트: 함수는 call 및 return과 관계됨) 4.4 프로그램 수행 화면을 보고 fib(4) 호출 시 fib(2)가 재귀적으로 몇 번 호출되었는지 확인하여 쓰고, 왜 그런지 그 이유를 설명하라.		

번호	5	점수	5
제목	매크로 SUM 정의 작성	파일	sum-macro.c
실습	매크로 SUM(from, to)을 프로그램하라. 이 매크로는 int 형의 from에서 int 형의 to까지의 정수를 합한 int 형의 결과 값을 출력한다. 예를 들어 SUM(1, 10)를 수행하면 55를 출력한다. (뒤 모범 수행 화면 참고)		
제출물	5.1 학생이 작성한 매크로를 쓰고, 프로그램을 이렇게 작성한 이유를 설명하라. 5.2 학생이 작성한 매크로를 사용하면, SUM(1, 10)이 전처리(preprocess) 시 대체(substitute)되어 어떤 프로그램으로 변하는지 적어라. (주의: SUM(1, 10)의 수행 화면을 쓰는 것이 아님) 5.3 프로그램 수행 화면을 써라(혹은 수행 화면을 캡처하여 넣어라).		

번호	6	점수	5
제목	매크로 CAL 정의 작성	파일	cal-macro.c
실습	매크로 CAL(from, to, init, op, str)을 프로그램하라. 이 매크로는 int 형의 from에서 int 형의		

	to까지의 정수를 산술 연산자인 op로 계산한 float 형의 결과 값을 출력한다. 이때 init은 계산 시 첫 operand의 값이고, str은 해당 산술 연산자를 출력하기 위한 스트링이다. 예를 들어 CAL(10, 20, 10, +, "+")를 수행하면 10+10+11+12+13+14+15+16+17+18+19+20=175.000000를 출력한다. (뒤 모범 수행 화면 참고)
제출물	6.1 학생이 작성한 매크로를 쓰고, 프로그램을 이렇게 작성한 이유를 설명하라. 6.2 학생이 작성한 매크로를 사용하면, CAL(10, 20, 10, +, "+")가 전처리(preprocess) 시 대체(substitute)되어 어떤 프로그램으로 변하는지 적어라. 6.3 프로그램 수행 화면을 써라(혹은 수행 화면을 캡처하여 넣어라).

모범 수행 화면	
<pre> \$./array-1-file add_array(0)=0: (0) add_array(1)=1: (0,1) add_array(2)=2: (0,1,2) add_array(3)=3: (0,1,2,3) add_array(4)=4: (0,1,2,3,4) add_array(5)=-1: (0,1,2,3,4) delete_array(4)=4: (0,1,2,3) delete_array(3)=3: (0,1,2) delete_array(2)=2: (0,1) delete_array(1)=1: (0) delete_array(0)=0: () delete_array()=-1: () \$./array-4-file add_array(0)=0: (0) add_array(1)=1: (0,1) add_array(2)=2: (0,1,2) add_array(3)=3: (0,1,2,3) add_array(4)=4: (0,1,2,3,4) add_array(5)=-1: (0,1,2,3,4) delete_array(4)=4: (0,1,2,3) delete_array(3)=3: (0,1,2) delete_array(2)=2: (0,1) delete_array(1)=1: (0) delete_array(0)=0: () delete_array()=-1: () \$./counter counter(0)=0 counter(1)=1 counter(2)=3 counter(3)=6 counter(4)=10 counter(0)=10 counter(-1)=9 counter(-2)=7 counter(-3)=4 counter(-4)=0 \$./fib > fib(4) > fib(3) > fib(2) > fib(1) < fib(1)=1 > fib(0) < fib(0)=0 < fib(2)=1 > fib(1) < fib(1)=1 < fib(3)=2 > fib(2) > fib(1) </pre>	

```
| | < fib(1)=1
| | > fib(0)
| | < fib(0)=0
| < fib(2)=1
< fib(4)=3
$ ./sum-macro
55
5050
0
-5050
$ ./cal-macro
0+1+2+3+4+5+6+7+8+9+10=55.000000
10+10+11+12+13+14+15+16+17+18+19+20=175.000000
0-1-2-3-4-5-6-7-8-9-10=-55.000000
100-10-11-12-13-14-15-16-17-18-19-20=-65.000000
1*1*2*3*4*5*6*7*8*9*10=3628800.000000
100/1/2/3/4/5/6/7/8/9/10=0.000028
$
```

주의: eCampus의 "참고자료" 폴더에 있는 "실습 제출물 작성 지침"에 따라 작성되지 않은 경우 감점 있음.

끝.