

REPORT

보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목 : 프로그래밍 기초와 실습

과 제 명 : 실습 9 : Structures

담당교수 : 민 형 복 교수

학 과 : 자연과학계열

학 년 : 1학년

학 번 : 2014310407

이 름 : 이 준 혁

제 출 일 : 2014년 12월 7일

1. Introduction

이번 실습에서는 structure에 대해 공부한다. 민형복 교수님의 ch12.pdf에 나온 말을 인용하자면, 일전에 같은 type의 변수들을 한 번에 선언하기 위해 array라는 도구를 이용했다. structure 역시 비슷하게 한 변수에 여러 type을 동시에 선언하기 위해 사용하는 도구이다.^[1] 사실 structure를 사용하지 않아도, 프로그램을 하는 데에는 문제가 없다. 그러나 structure는 한 변수가 여러 type의 값을 가지게 해 줌으로써, 프로그램을 아주 간단하게 만들어주기도 하고, 프로그래밍 하기도 쉽게 만들어 준다.

이런 이유 때문에 structure에 대해 잘 알고 있어야 한다. 따라서 이번 실습에서는 코일의 인덕턴스와 길이를 구하는 프로그램을 짜 보며, structure를 어떻게 사용하는 지에 대해 익힐 것이다.

2. Problem Statement

(A)Describe what the problem is

최종 목표는 코일의 전체 길이와 전체 인덕턴스 값을 구하는 것이다. 이를 위해서, 네 개의 함수를 짜보는 것이 이번 실습의 목표이다.

1. build_a_coil() : 각각의 코일의 radius, number of turns, inductance 값을 코일을 나타내는 변수의 structure member에 저장하는 함수이다.
2. print_coil_info() : 각 코일의 member에 저장된 정보를 출력하는 함수이다.
3. length_of_wire() : 코일의 전체 길이를 구하는 함수이다.
4. inductance_in_series() : 코일의 전체 인덕턴스 값을 구하는 함수이다.

이 과정을 다음과 같은 member가 있는 COIL이라는 structure를 이용해 프로그래밍 해야 한다.

double radius; (반지름)

int number_of_turns; (감은 횟수)

double inductance; (인덕턴스)

또, 한 코일의 길이를 구하는 법은 다음과 같다.^[2]

$$L = \frac{R(N^2R - 9H)}{10H}$$

L은 길이, R은 반지름, N은 감은 횟수, H는 인덕턴스이다.

그리고 총 길이를 구하려면 모든 코일의 길이를 다 더하면 되고, 총 인덕턴스를 구하려면 역시 모든 인덕턴스를 다 더하면 된다.

(B) Describe how you solve the problem.

우선 민형복 교수님의 program template p9.c를 살펴보았다. 동적 메모리 할당을

제외한 프로그램은 다 짜져 있었고, 위 네 개의 함수만 작성하면 된다는 사실을 알게 되었다. 동적 메모리 할당은 별 문제가 되지 않으므로 함수 네 개를 어떻게 짜야 할 지 생각해 보았다.

1. build_a_coil 함수 : 각각의 코일의 radius, number of turns, inductance 값을 코일을 나타내는 변수의 structure member에 저장하는 함수이다. function 내부에서 둘 이상의 값을 변경시켜야 하니, 당연히 pointer로 입력을 받을 것이라고 보았다. 그러기 위해서는 COIL변수 하나의 주소를 pointer로 입력받고, 또 그 COIL변수의 member에 저장하고 싶은 값을 입력 받으면 될 것이라 생각했다. 그 다음에 함수 내에서 operator -> 를 이용해 입력 받은 pointer가 가리키는, main 함수에 있는 COIL 변수의 member값에 접근 할 수 있을 것이라 보았다. 이 것은 모두 pointer를 이용하기 때문에 함수 내에서 main함수의 변수의 값을 바꿀 수 있다는 데에서 기인했다.

2. print_coil_info 함수 : COIL structure의 정보를 출력하는 함수이다. COIL type의 변수를 하나 입력으로 받아서 member operator "."를 이용해 그 변수의 member값을 차례대로 출력하면 될 것 같았다. 값을 변경시키는 것이 아니므로 굳이 pointer를 이용해 입력 받을 이유는 없다고 생각했다.

3. length_of_wire 함수 : coil의 전체 wire 길이를 구하는 함수이다. 각 coil들을 입력 받아서 그 coil들의 길이를 구한 후 출력하고, 그 길이들을 모두 더해 반환하면 될 것이라 보았다. 그런데 coil을 입력 받을 때 하나하나 입력 받기에는

번거로워 보였으므로, program template에도 나와 있듯이 coil들이 들어있는 array를 pointer의 형태로 입력 받는 게 나을 것이라 보았다. 또, 이렇게 되면 몇 개의 array를 더해야 하는 지 모르므로, 몇 개를 더해야 하는 지를 알려주는 array의 크기를 입력 받기로 했다.

4. inductance_in_series 함수 : coil의 전체 인덕턴스 값을 구하는 함수이다. 이 함수는 입력 받은 array의 각 inductance라는 member들을 다 더하면 될 것이라 보았다. length_of_wire 함수와 마찬가지로 역시 pointer형태의 array와 array의 크기를 입력 받기로 했다.

3. Implementation

헤더 파일 삽입이나 COIL이라는 type의 선언은 모두 민형복 교수님의 program template p9.c에 있는 것을 참고했다.^[3] 또 main함수의 내용이나 function의 이름들 역시 template를 참고했다.

3. 1. 동적 메모리 할당

우선 main함수에 coils라는 COIL type의 pointer 변수가 주어졌는데, 이것을 array처럼 여러 data를 담게 하기 위해서는 메모리를 할당하는 과정이 필요했다. 그렇기 때문에 실습 7에서 했던 것처럼 ($4 * \text{sizeof}(\text{COIL})$)만큼의 메모리를 할당 받았다. 4를 곱한 이유는 총 코일의 개수, 즉 array의 크기가 4였기 때문이다.

3. 2. build_a_coil 함수 작성

이 함수를 포함해 작성한 네 개의 함수의 이름은 주어진 template의 것을 참고했다. 구상한 대로 이 함수의 parameter로 값을 변경시키고자 하는 COIL 변수의 주소를 입력 받고, 그 pointer가 가리키는 main의 COIL 변수의 member에 할당할 값을 입력으로 받았다.

이 때 parameter를 어떻게 작성 해야 할 지를 정했다. 우선 COIL type의 주소를 담는 pointer이기 때문에 coil이라는 포인터 변수를 이용해 주소를 입력 받도록 했다. 또 COIL이라는 변수의 member의 이름과 거의 같게 member에 들어갈 값들의 parameter를 명명했다. 반지름을 의미하는 parameter를 실수형 변수 radius로, 감은 횟수를 의미하는 parameter를 정수형 변수 turns로, 인덕턴스를 의미하는 parameter를 실수형 변수 inductance로 이름 붙였다. 이렇게 설정하면 member의 이름과 같아서 헷갈릴 여지가 줄어들고 더 편리해 질 것이라 생각했기 때문이다.

그 다음에는 coil이라는 변수가 pointer라는 점을 이용해 -> operator를 이용해 그 pointer가 가리키는 값의 member변수를 입력 받은 값을 통해 저장했다. 다음은 그 예시이다.

```
coil -> radius = radius;
```

값의 할당 이외에 이 함수가 수행하는 일은 없으므로, return을 통한 반환 없이 함수를 끝내기로 했다.

3. 3. print_coil_info 함수 작성

이 함수는 단순히 coil의 정보를 출력하는 함수이므로, 위 함수와 달리 굳이 pointer로 입력 받을 필요가 없었다. 함수 내부에서 값을 변경시키지 않기 때문이다. 물론 값을 변경시키지 않아도 메모리의 관점에서는 pointer를 이용하는 것이 낫지만, 일단 이번 실습에서는 그냥 COIL type의 변수 그 자체를 입력으로 받아들이기로 했다.

parameter를 COIL type의 coil으로 명명했다. 이렇게 되면 coil이라는 변수에 담겨 있는 member들의 값이 전달되므로, member operator를 이용해 coil의 member들(반지름, 감은 횟수, 인덕턴스)에 접근해 출력할 수 있었다. 그리고 printf 함수를 이용해 coil의 각 값들을 출력하는데, 이 때 콘솔 창에서의 가독성을 높이기 위해 'wt'를 사용했다.

이 함수 역시 출력(print)만이 목적이므로, return을 통한 반환이 필요없었기 때문에 어떤 값도 return하지 않았다.

3. 4. length_of_wire 함수 작성

이 함수는 두 개의 역할을 수행해야 했다. 우선 각 코일의 길이를 구하고 그 길이를 출력하는 것과 각 코일의 길이들을 다 더해서 그 값을 반환하는 것 두 가지가 필요했다.

가장 먼저 코일들을 담고 있는 main에 있는 coils라는 array(pointer로 선언되어있긴 하지만)를 입력 받기로 했다. COIL type의 pointer이므로 그 이름을 coil이라 두었다. 또, 몇 개의 coil이 있는 지 알아야 그 개수만큼의 array를 다 더할 수 있으므로, coil이 몇 개인지 역시 입력 받기로 했다. 이것을 수를 뜻하는 정수형 변수 number로 명명했다.

그 다음에 cnt라는 셈을 위한 정수형 변수를 선언했다. 세기 위한 변수이므로 cnt는 0으로 초기화 했다. 또, 각 array의 coil의 길이를 저장하고 또 출력하기 위해 길이를 출력한다는 의미로 print_length라는 실수형 변수를 선언했고, 함수의 끝에서 반환할 총 길이를 계산하기 위한 총 길이라는 의미의 total_length라는 실수형 변수를 선언했다. 이 변수에 각 coil의 길이를 더할 것이므로 0으로 초기화 했다. 게다가 이 길이들은 모두 실수값을 가지므로, 실수형 변수로 선언했다.(왜냐하면 inductance, radius도 모두 실수인 데다 길이를 계산하는 과정에 분수가 들어 있어 실수가 나올 것이었기 때문)

그리고 while문과 변수 cnt를 이용해 number개의 array를 모두 더해야 했다. 그러기 위해서 array에 어떻게 접근하느냐가 문제였는데, coil이라는 parameter 뒤에 array표현식을 이용해 접근하기로 했다. 즉, coil[cnt]와 같은 형식으로

접근했다. 이것이 가능한 이유는 coil이라는 parameter가 입력 받을 때 main에 있는 coils라는 저장할 메모리가 있는 포인터 변수의 값을 복사해 왔기 때문에, 위와 같은 array의 식으로 접근할 수 있는 것이다.

이 점을 이용해 cnt가 0부터 number - 1일 때까지 coil[cnt]의 각각의 길이를 구했다. 이 때 다음과 같은 식을 이용했는데, 이 때 각 coil의 memebre에 접근하기 위해 member operator를 이용했다.

$$L = \frac{R(N^2R - 9H)}{10H}$$

그리고 이렇게 해서 나온 길이를 print_length에 저장시켰고, 그 값을 출력한 후 total_length에 더했다. while문을 통해 coil[number - 1]까지 이 과정을 반복하고 끝나면 total_length에는 number개 길이의 총 합이, 즉 총 길이가 들어 있을 것이다. 그래서 바로 total_length를 반환하고 함수를 종료했다. total_length가 실수이므로, 이 함수의 return type은 double으로 선언했다.

3. 5. inductance_in_series 함수 작성

이 함수 역시 length_of_wire 함수와 비슷하게 생각하기로 했다. 다만 length_of_wire 함수에서는 coil의 길이를 계산하고 모두 더해야 했지만, 이 함수는 주어진 inductance값을 그냥 다 더하기만 하면 됐다.

그래서 우선 length_of_wire함수와 같이 COIL pointer type의 coil과 (pointer이자

array) 그 array의 크기를 나타내는 number를 입력으로 받았다. 또, 함수 내부에서 셈을 하기 위한 변수인 cnt라는 정수형 변수를 0으로 초기화 시키고, 총 인덕턴스를 계산할 총 인덕턴스라는 의미를 가진 실수형 변수 total_inductance를 0으로 초기화 시켰다.

그리고 length_of_wire함수와 마찬가지로 while문을 이용해 coil[cnt].inductance의 값을 cnt가 0부터 number - 1일 때 까지 다 더해 total_inductance값에 저장했다. 이렇게 되면 total_inductance 값에 총 inductance값이 들어가게 되므로, total_inductance를 반환했다. 이 변수는 double type이므로, return type 역시 double로 설정했다.

3. 6. 메모리 반환

inductance_in_series라는 함수가 끝나면, main함수 내의 coils라는 변수의 사용이 종료되었으므로 메모리를 deallocate하도록 free함수를 이용해 작성했다.

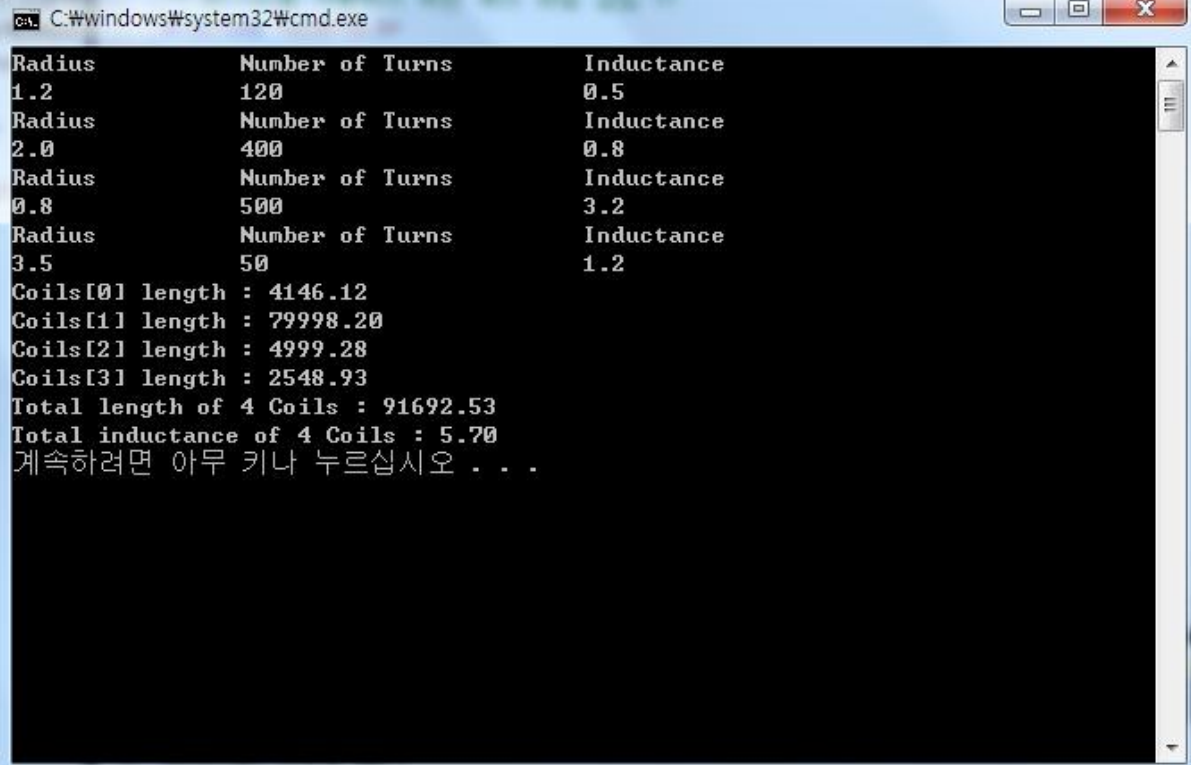
4. Result

우선 다음 내용들이 잘 출력되는 지 확인하려 했다.

- 1) 반지름, 감은 횟수, 인덕턴스가 main함수 내에서 주어진 대로 잘 출력되는 지 여부
- 2) 각 coil의 length가 잘 출력되는지 여부

3) 총 length와 총 inductance가 잘 출력되는지 여부

결과 화면은 다음과 같다.



```
C:\windows\system32\cmd.exe
Radius      Number of Turns      Inductance
1.2         120                  0.5
Radius      Number of Turns      Inductance
2.0         400                  0.8
Radius      Number of Turns      Inductance
0.8         500                  3.2
Radius      Number of Turns      Inductance
3.5         50                   1.2
Coils[0] length : 4146.12
Coils[1] length : 79998.20
Coils[2] length : 4999.28
Coils[3] length : 2548.93
Total length of 4 Coils : 91692.53
Total inductance of 4 Coils : 5.70
계속하려면 아무 키나 누르십시오 . . .
```

[그림 1] 프로그램 실행 후 결과 화면

1) main에서 각 코일의 반지름, 감은 횟수, 인덕턴스가 다음과 같이 주어졌다.

(1.2, 120, 0.5), (2.0, 400, 0.8), (0.8, 500, 3.2), (3.5, 50, 1.2)

이는 위 결과와 일치한다.

2) 코일의 길이를 구하는 식을 이용해 위 값들을 직접 계산기를 통해 구했을 때 나오는 값은 다음과 같다.

4146.12, 79998.2, 4999.28, 2548.933...

즉, 위 결과는 실제로 계산한 값과 일치한다. (게다가, 민형복 교수님의 practice9.pdf에 나온 결과와도 일치한다.)

3) 네 코일의 길이와 인덕턴스를 각각 다 더해보았다.

네 코일의 길이의 합 : 91692.53

네 인덕턴스의 합 : 5.7

이는 역시 위의 결과와 일치한다

따라서, 모든 결과가 예측한 것과 같이 나왔다.

5. Conclusion & Evaluation

structure는 앞서 배운 개념들보다 어렵지 않았기 때문에, 이번 실습에서 structure를 직접 다뤄보며 프로그래밍 해 봄으로 인해 structure에 대해 쉽게 익숙해 졌다.

structure라는 도구는 프로그램의 logic 자체에 큰 영향을 주지는 못하지만, 이것을 잘 이용하면 프로그램을 정말 쉽게 만들 수 있을 거라는 느낌을 이번 실습을 통해 얻을 수 있었다. 만약 이번 실습에서 각 코일마다 그에 해당하는 반지름, 감은 횟수, 인덕턴스 변수를 선언했다고 생각하면, 프로그램이 이해하기도 어려워 졌을 테고, 작성할 때도 훨씬 더 힘들었을 것이다.

그런 의미에서 이번 실습에서는 프로그램을 더 간단히 만들 수 있는 법을 배웠다.
또, 이것을 pointer나 array와 같은 개념들과 연계해 사용함으로 인해 pointer나
array에 대해서도 더 생각해 보고, 심도 깊게 이해할 수 있었다.

이제 다음 실습이 마지막 실습이다. 이전에는 프로그래밍에 대해서 낯설기만
했는데 이제는 이렇게 structure, pointer, array등등 여러 개념을 이용해
프로그래밍 할 수 있게 되었다는 것을 새삼 깨닫게 되니 뭔가 뿌듯하다. 다음
실습도 무사히 끝내서 프로그래밍 기초와 실습이라는 과목을 잘 마무리 할 수
있도록 해야겠다.

6. 참고 문헌

- [1] 민형복, 강의 자료, ch12.pdf, page.2
- [2] 민형복, practice9.pdf, page. 7
- [3] 민형복, program template, p9.c.
- [4] Al Kelly, Ira Pohl, *C by Dissection: The Essentials of C Programming, Fourth Edition*, Pearson, p.391 ~ p.400.
- [5] 프로그래밍 기초와 실습 사이트, <http://class.icc.skku.ac.kr/~min/C/>, 보고서 작성
요령, 2014 년 12 월.