**Report 4 – 설명**

**20182931 신석경**

Report 4-1. 사원의 정보를 출력하고, 인센티브를 수정할 수 있는 프로그램

실행파일: employee.c

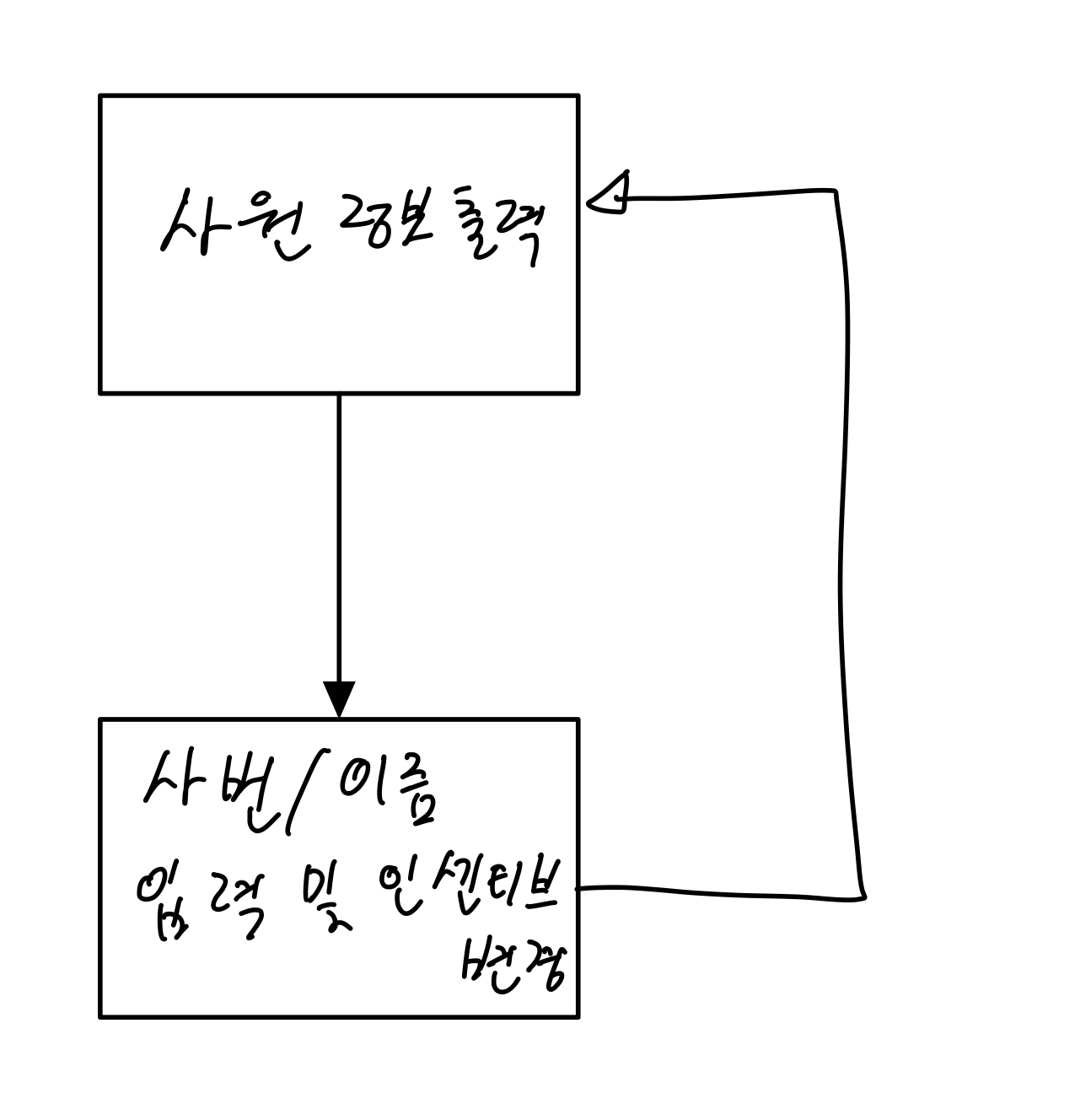
이 프로그램은 반복적으로 사원의 사번 혹은 이름을 입력 받고 인센티브 조정 수치를 입력 받아 다시 각 사원의 연봉을 계산하고 그 결과를 출력하는 프로그램이다. 이 프로그램의 메인 흐름은

1번. 회사 사원의 정보를 출력한다.

2번. 사번 혹은 이름을 입력 받고 인센티브 변경 수치도 입력 받는다.

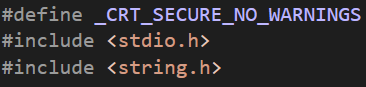
3번. 변경된 수치를 반영한 회사 사원의 정보를 출력하고 다시 2번으로 돌아간다.

프로그램의 동작 흐름을 살펴보면 아래의 그림과 같다.

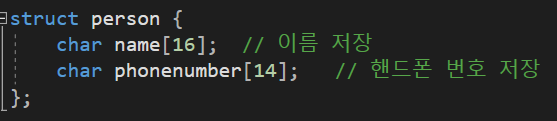


프로그램에서 사용할 구조체 2개 person과 employee를 선언하였다. 그리고 사원 정보를 출력하는 print() 함수, 연봉을 계산하는 함수 calYearMoney() 함수를 선언하였다.

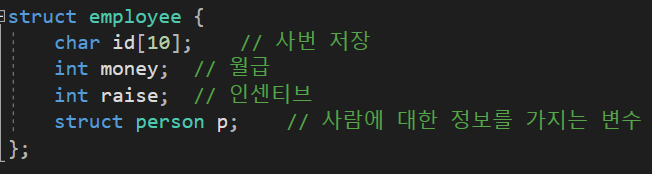
이 프로그램은 반복적 루프이기 때문에 while문을 사용하였다. while() 문 속에서 정보 출력 -> 사번 혹은 이름 및 새로운 인센티브 입력 -> 정보 바꿈 -> 다시 정보 출력 순으로 실행된다.



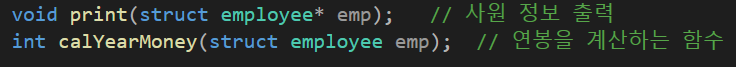
먼저, 필요한 헤더 파일을 include 하였다. string.h를 include 한 이유는 strcmp를 사용하여 사번 혹은 이름을 입력된 값과 비교하기 위함이다.



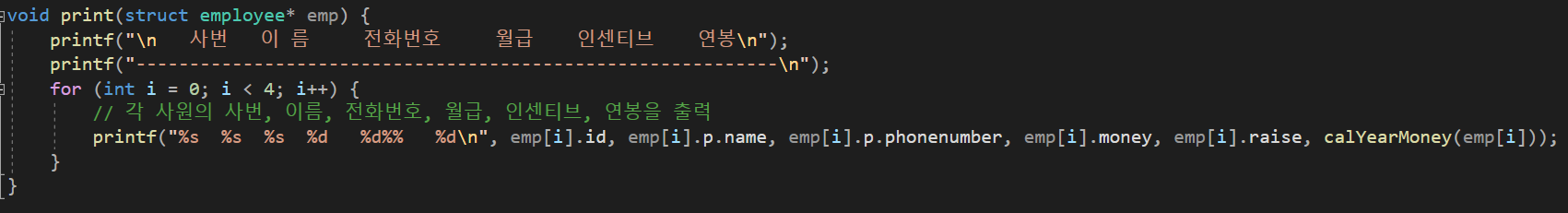
사람을 나타내는 person 구조체이다. 이름과 전화번호를 각각 char 배열인 name과 phonenumber로 선언한다.



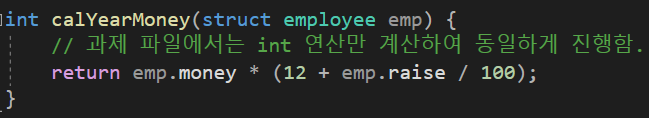
사원을 나타내는 employee 구조체이다. 여기에는 먼저 사번을 저장하는 id변수가 있고,사람의 정보가 들어가니 struct person 변수 p를 가지게 하고, 월급을 저장하는 money와 인센티브를 저장하는 raise 변수를 선언한다.



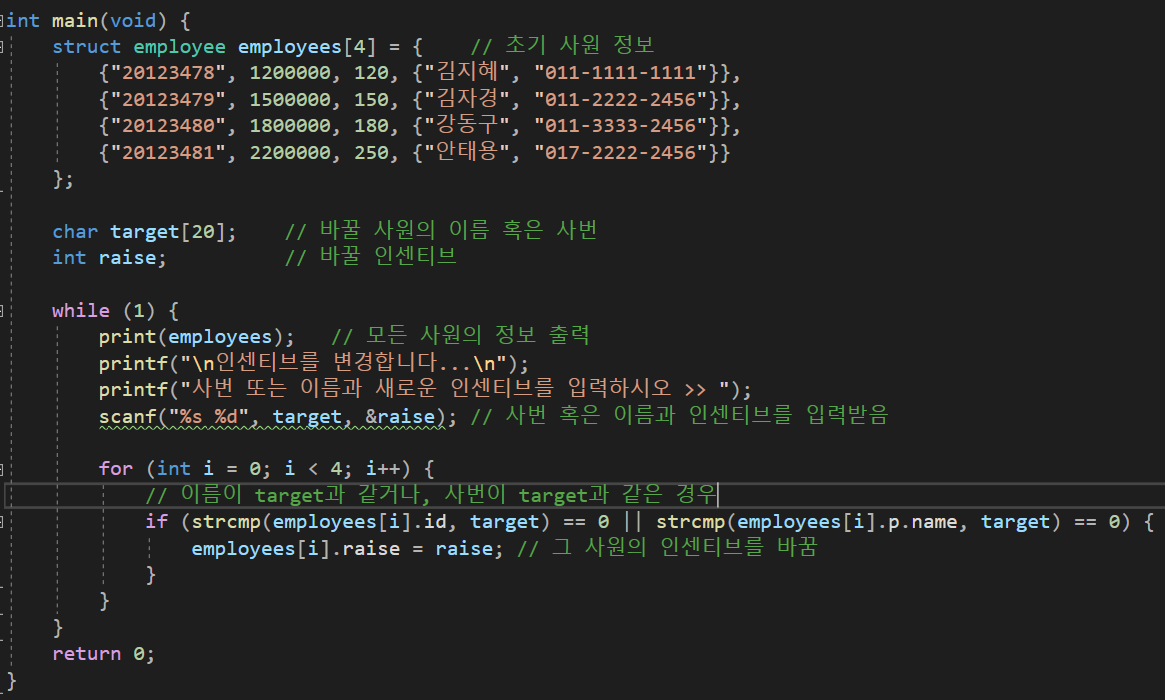
각 함수의 정의이다. print함수는 employee 배열을 받아 출력하고, calYearMoney는employee 인스턴스를 받아 계산한다.



print함수이다. 여기서는 각 출력 형식에 맞춰서 4명의 사원의 정보를 출력한다. 연봉의 경우 calYearMoney함수에 해당하는 사원을 보내 계산하여 출력한다.

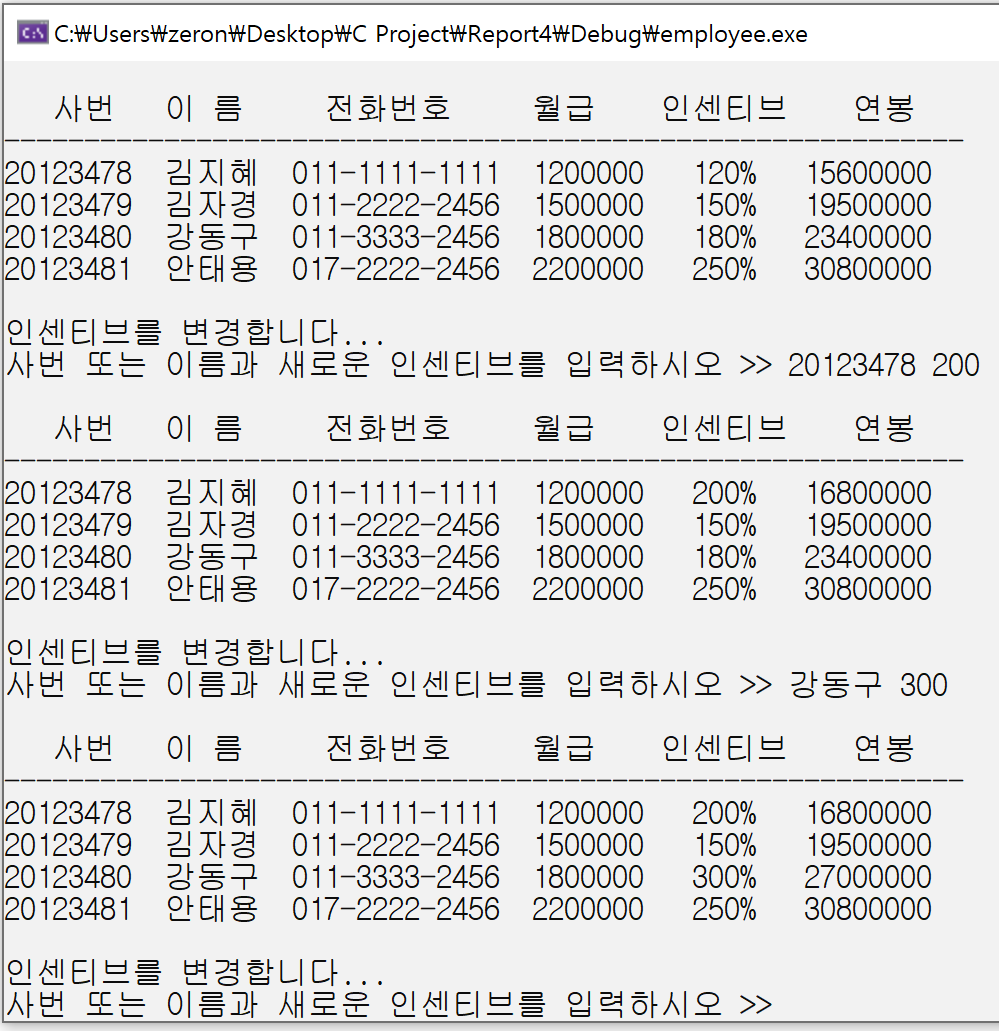


calYearMoney()함수이다. 월급 \* (12 + 인센티브 비율 ) / 100 로 계산한다. 이때, 과제 설명 파일에서 각 연봉을 실수 계산한 것이 아니라 정수 계산을 하여 이 함수에서도 int를 기반으로 계산하였다. 정확한 계산을 위해서라면 float로 계산하여야 한다.



메인 함수이다. 먼저 employee 4명에 대해 배열을 선언하고 각각의 정보를 초기화한다. 그리고 바꿀 사원의 이름 및 사번을 저장하는 target 문자배열을 선언하고, 변경할 인센티브를 저장할 raise 변수를 선언한다. while문에서 먼저 사원의 정보를 출력하고, scanf를 통해 사번/이름 과 인센티브를 입력 받는다. 이후 for문을 통해 4명의 employee를 검사하는데, 각각의 employee의 사번이나 이름을 strcmp를 사용하여 비교해 둘 중에 하나라도 같은 경우 그 사람에 대한 인센티브를 변경한다. 그리고 다시 while문으로 돌아가 사원의 정보를 출력한다.

결과 출력창이다.



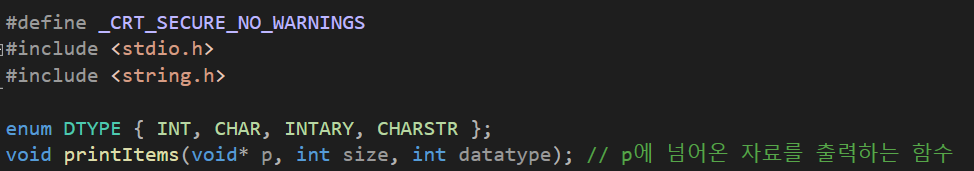


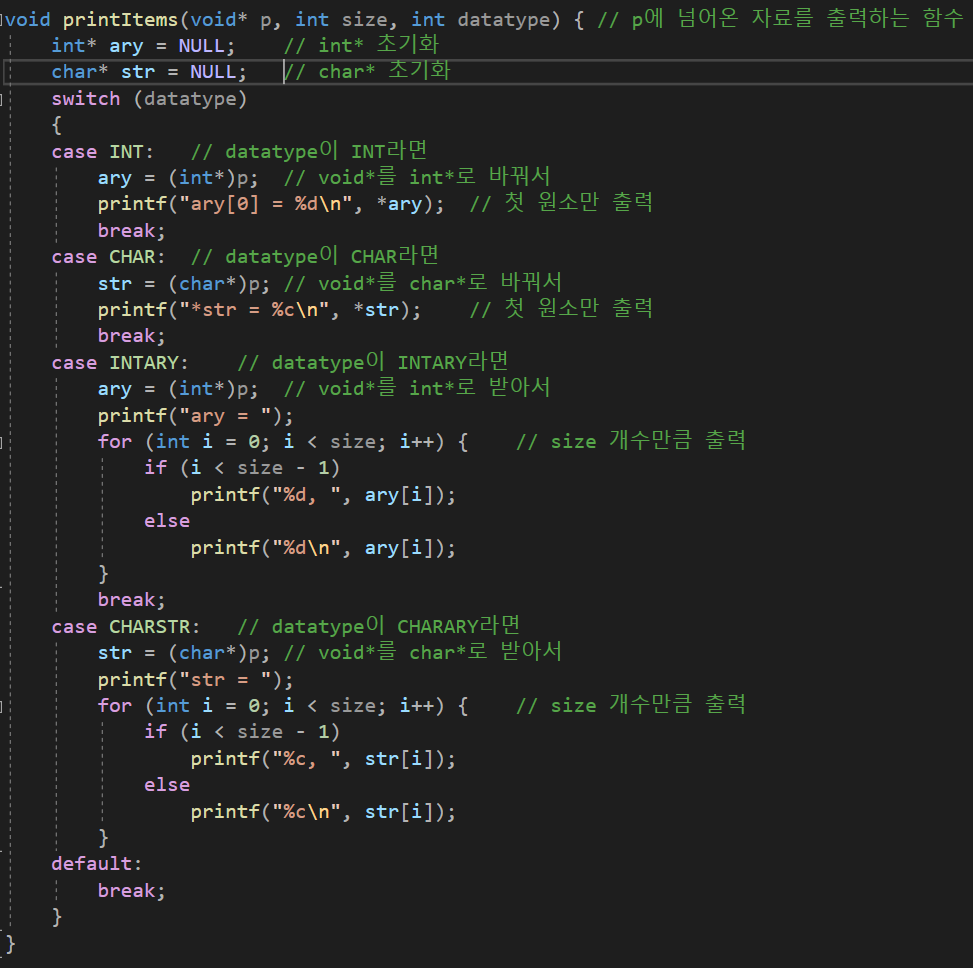
Report 4-2. 정수, 문자, 정수 배열, 문자열을 출력할 수 있는 함수 정의

실행파일: varprint.c

이 프로그램은 정수, 문자, 정수 배열, 문자열을 출력할 수 있는 통합 함수를 정의하는 것이다.

바로 코드를 통해 알아보자.

필요한 헤더와 열거형, 함수 정의이다. string.h은 strlen을 위하여 include 하였고, DTYPE을 통해 함수를 실행할 옵션을 결정한다. 그리고 통합 함수인 printItems를 선언하였다.



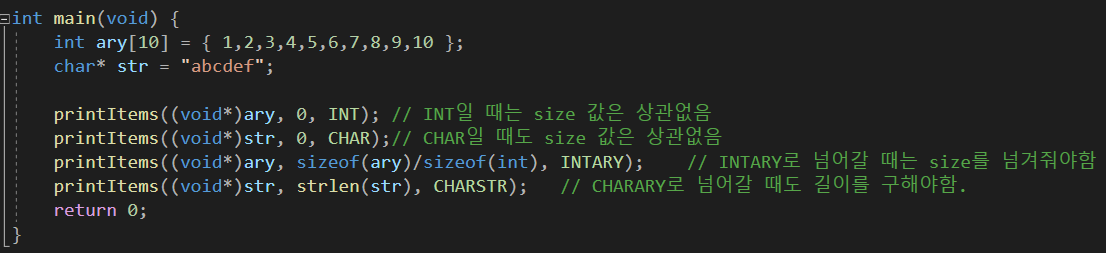
printItems()함수이다. 첫번째 void\* p는 정수 배열 혹은 문자열이 넘어오기 때문에 void\*로 선언하였고, size는 출력할 개수를 정하는 변수이다. datatype은 넘어온 자료를 무엇으로 출력할지 정하는 옵션이다. 먼저 int\* ary와 char\* str를 NULL로 초기화하여 나중에 사용할 때 할당할 수 있도록 한다.

만약 datatype이 INT라면 정수 하나를 출력한다. 그러므로 p를 int\*로 변환하여 ary에 넣고, ary에서 첫번째 원소만 출력하기 때문에 \*ary를 통해 출력한다.

datatype이 CHAR이면, p를 char\*로 바꾸고, \*str를 통해 첫번째 문자만 출력한다.

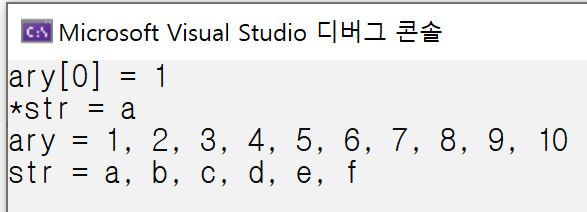
datatype이 INTARY이면, p를 int\*로 바꾼 다음, size 개수만큼 정수를 출력한다.

datatype이 CHARSTR이면, p를 char\*로 바꾼 다음, size 개수만큼 문자를 출력한다.



메인 함수이다. 먼저 필요한 ary와 str를 정의하고, 각각의 함수를 호출한다. 먼저 함수의첫번째 argument가 void\* 이기 때문에 각각의 ary와 str을 void\*로 형변환 하여 넘겨준다. 그 다음 INT, CHAR로 datatype을 넘길 때에는 하나의 문자만 출력하기 때문에 size에 어떤 값이 들어가도 상관없다. INTARY, CHARSTR로 넘어갈 때는 각각 배열의 원소 개수를 size로 넘기거나, 문자열의 길이를 size로 넘겨 출력하도록 한다.

결과를 살펴보자



Report 4-3. 특정 파일 내부에 검색한 단어가 있는지 확인하는 프로그램

실행파일: searchword.c

이 프로그램은 문자열을 입력 받고, 실행 인자로 파일을 하나 입력 받아, 해당 파일 내부에 입력된 문자열을 포함한 줄의 모든 문자를 줄 번호와 함께 출력하는 프로그램이다. 이 프로그램의 흐름은

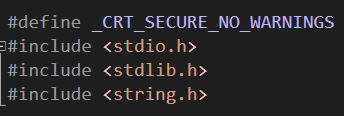
1번. 검사할 File을 open한다.

2번. 찾을 단어를 입력 받는다.

3번. 입력된 단어가 나타난 줄의 줄 번호와 내용을 출력한다.

이 프로그램에서는 file 입출력을 사용하고, 문자열 함수를 사용하여 문자열이 입력된 한 줄에 있는지 검사한다.

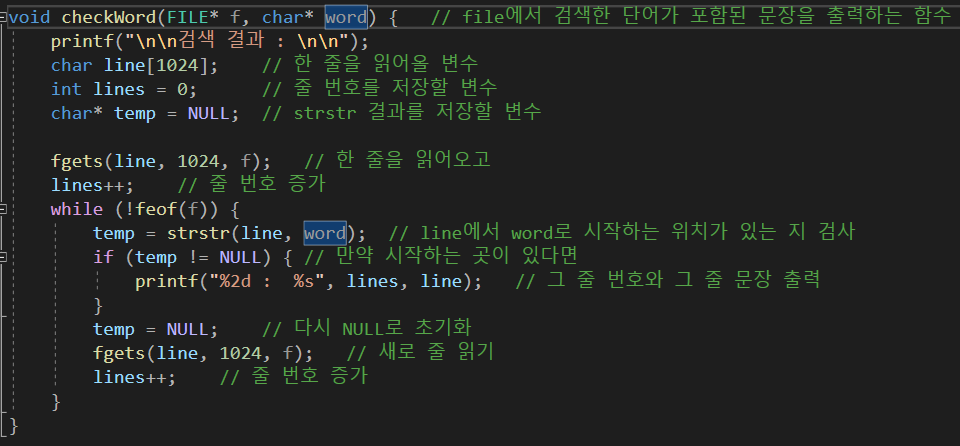
자세한 것은 코드를 살펴보자.



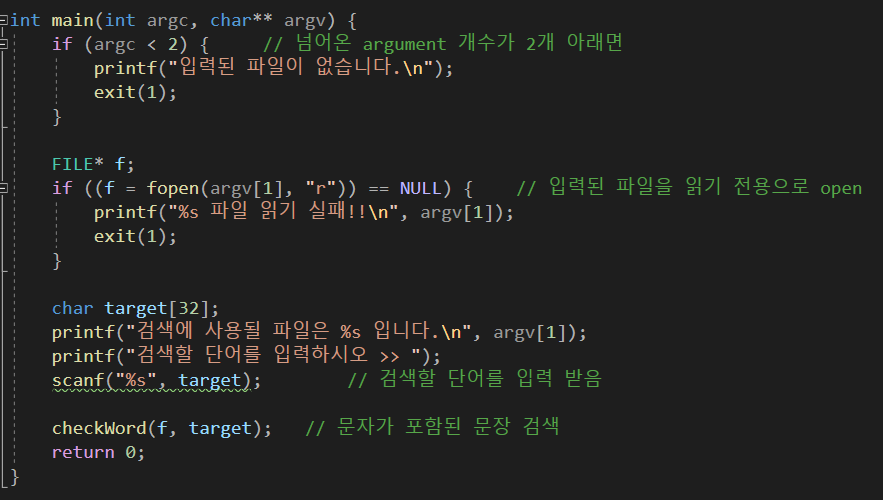
필요한 헤더를 include 한다. 이때 string.h는 strstr() 함수를 사용하기 위해 include 하였다.



단어가 포함된 문자열을 찾고 출력하기 위한 함수 선언이다.

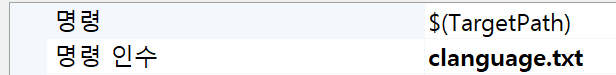


함수 정의 부분이다. file에서 한 줄을 읽어오기 위한 buffer인 line 변수를 선언하고, 줄 번호를 저장할 lines 변수를 선언한다. 그리고 strstr() 함수 결과를 저장하기 위한 char\* temp 변수를 NULL로 초기화 해준다. 먼저 fgets를 이용해 FILE stream f에서 한 줄 읽어오고 줄 번호를 증가시킨다. 그리고 EOF가 아니면, strstr(line, word)를 사용하여 line 내부에 word 문자열이 포함되어 있으면 그 위치를 temp에 저장하게 만든다. 이때, temp != NULL 이란 것은 문자열 내부에 찾는 단어가 있다는 것이기 때문에 그 줄 번호와 그 줄 문자열인 line을 출력한다. 이후 다시 temp = NULL로 초기화하고, 다시 fgets를 통해 한 줄을 읽고 줄 번호를 증가시킨다. 이것을 EOF가 나올 때까지 반복하면, 단어가 포함된 모든 줄을 알 수 있다.

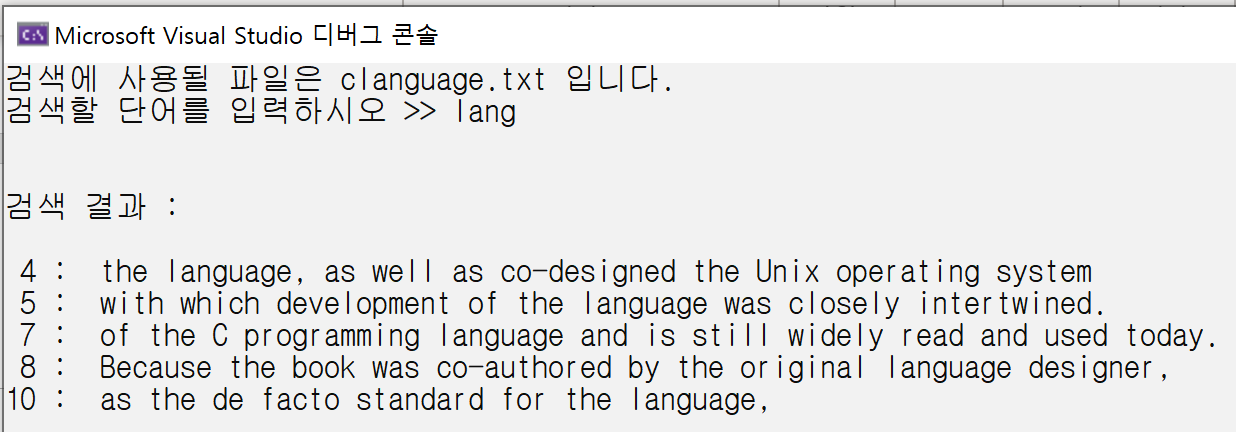


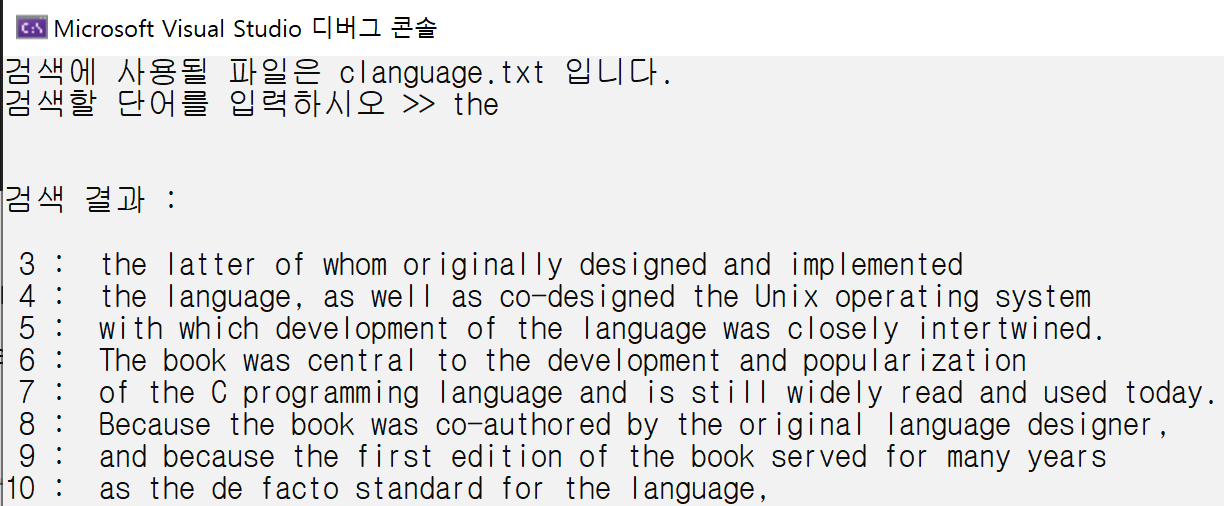
메인 함수이다. 먼저 argument로 2개 미만이 들어오면 오류를 출력하고 프로그램을 종료한다. 만약 넘어온 파일이 있다면, 읽기 전용으로 file을 open한다. 이후 검색할 단어를 target에 넣고 checkWord 함수를 사용하여 검사를 시작한다.

테스트를 위하여 명령 인수를 clanguage.txt를 넘겨주었다.



출력 결과이다.





Report 4-4. in.txt에서 구조체를 읽어 각 학생의 점수의 합을 구하여 출력하고 out.txt파일에 저장하는 프로그램

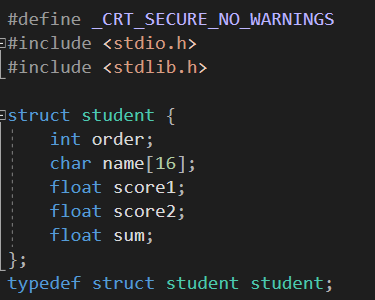
실행파일: score3.c

이 프로그램은 in.txt에서 한 줄 씩 읽어 그 정보를 구조체에 저장하고, 그 구조체로부터새로운 정보를 저장해 콘솔에 출력, out.txt에 출력하는 프로그램이다. 이 프로그램의 흐름은 아래와 같다.

1번. in.txt file에서 한 줄 씩 읽어 구조체에 저장한다.

2번. 저장한 구조체를 가지고 점수 합 계산 및 콘솔에 출력, out.txt file에 쓰기

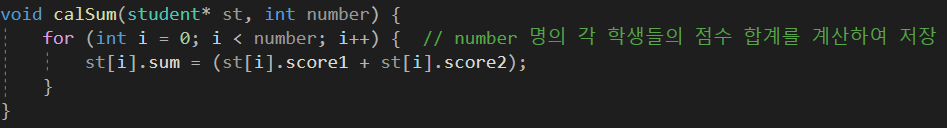
이 문제해결을 위한 코드를 살펴보자.



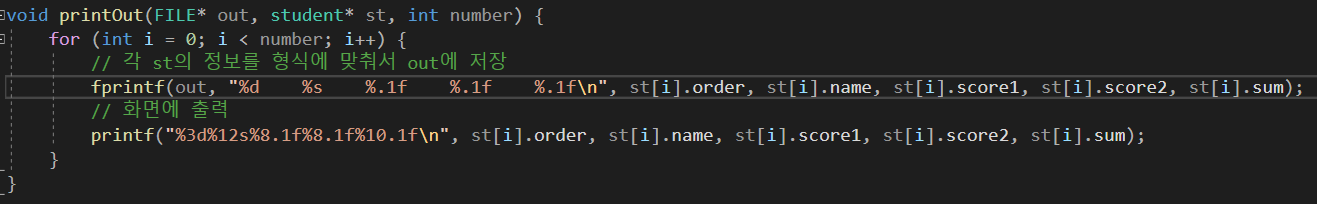
먼저, 필요한 헤더를 include 하였다. 그리고 student 구조체를 정의하였다. order에는 순번이 들어가고, name에는 이름, score1에는 점수1, score2에는 점수2가 들어가고 sum에는 score1 + score2 점수 합이 저장된다. 이때, sum 전까지는 in.txt에서 받아오고 sum은 직접 계산해주어야 한다.

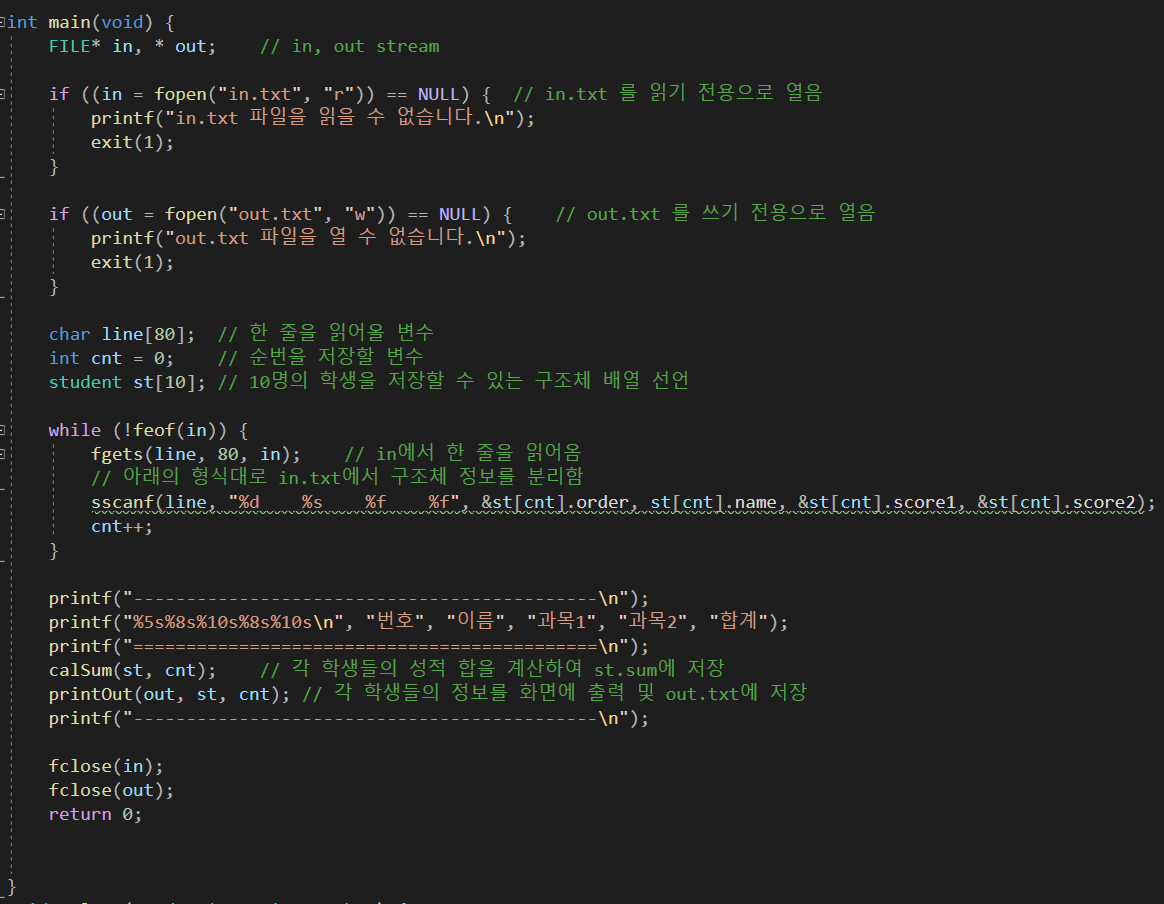


프로그램에서 사용될 함수이다. calSum은 student 배열과 학생 수를 입력 받으면, 각각의 점수의 합을 계산해 sum에 저장하도록 하는 함수이다. printOut은 학생들의 정보를 out file stream에 출력하는 함수이다.



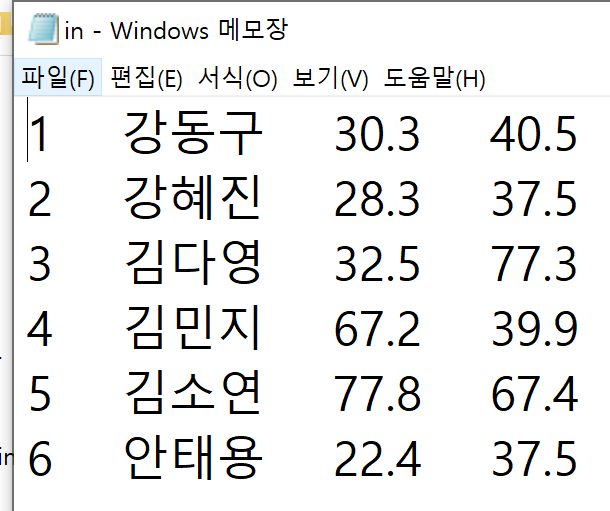
calSum 함수이다. for문을 사용하여 number 개수만큼의 학생들의 점수 합을 계산하여 그 구조체의 sum 변수에 할당한다.

printOut 함수이다. for 문을 통해 number 개수만큼의 학생들의 정보를 fprintf를 사용하여 out file에 출력하고, printf를 사용하여 콘솔에도 출력한다.

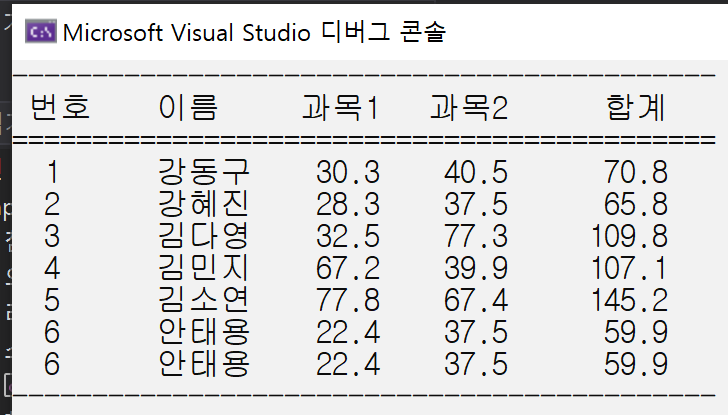


메인 함수이다. 먼저 in에는 in.txt file을 읽기 모드로 open하고, out에는 out.txt file을 쓰기 모드로 open한다. 그리고 in.txt에서 한 줄을 읽어올 변수 line과 순번을 저장할 변수 cnt, 최대 10명의 학생들을 저장할 수 있는 구조체 배열을 선언한다.

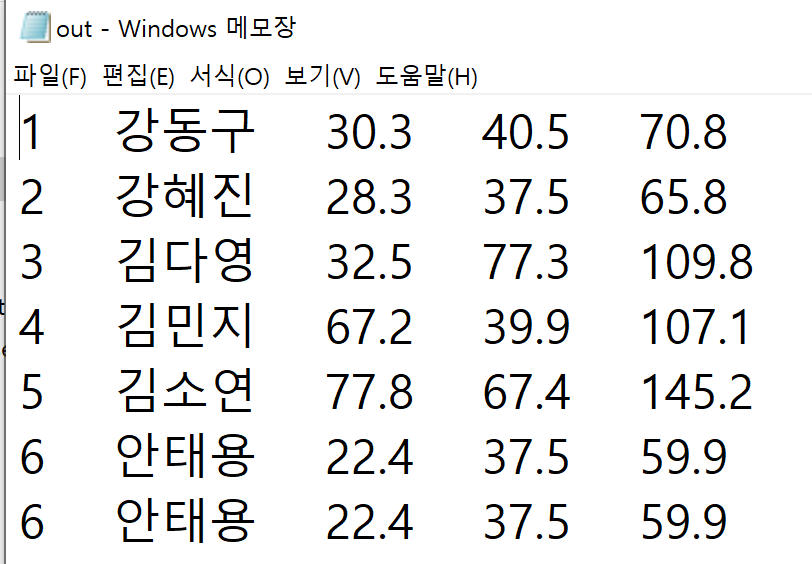
이후 in이 EOF가 아닐 때까지, fgets를 통해 한 줄을 읽어오고, sscanf를 통해 입력된 문자열 line으로부터 정보를 분할하여 st의 0번 index부터 차례대로 학생들의 정보를 저장한다. 이후, calSum(st, cnt)를 통해 cnt 개수만큼의 학생 수의 점수 합을 계산하고, printout(out, st, cnt) 함수를 호출하여, 학생 정보를 콘솔과 out.txt에 출력한다.



in.txt 파일 내용이다.



실행 결과 위의 in.txt에 있는 정보가 잘 읽혀진 것을 확인할 수 있고, 그에 대한 합계도 잘 출력되는 것을 확인할 수 있다.



out.txt에는 점수 합계를 포함한 구조체의 정보들이 쓰인 것을 확인할 수 있다.