코드 설명하기 앞서, 조교님이 제공해주신 값 두개를 입력 받아 더해서 출력하는 예제 코드에서 사용한 print, scan, print\_char는 상황에 맞게 적절히 변형하여 사용하였으나, 기본 틀은 그대로 가져와 사용하였다.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

# Ascending Sort

이 문제의 조건은 다음과 같다.

1. 5개의 수가 주어졌을 때, 이를 오름차순으로 정렬

2. 직접 5개의 수를 임의로 입력

3. 오름차순으로 정렬된 수를 출력

여기서 고려해야하는 것을 추려본다면 1. 입력, 2. 정렬, 3. 출력으로 정리할 수 있다.

입출력에 대해서는 양수뿐만 아니라 음수도 입출력이 가능해야한다. 또한, 한자리 숫자가 아닌 두자리 숫자의 입출력이 가능해야 한다. 한자리 양의 정수만 입력을 받고 출력을 한다면 문제가 굉장히 쉬웠겠지만, 두자리 음수, 양수의 입출력을 구현하는 것이 중요하다.

이를 구현에 성공한다면 ascending sort하는 알고리즘은 간단하다. 자세한 내용은 코드를 보면서 설명하도록 하겠다.

텍스트, 영수증, 화이트, 대수학이(가) 표시된 사진

자동 생성된 설명총 5개의 숫자를 저장하기 위한 레지스터를 r7-r11을 사용할 것이다. 이를 사용하기 위해 초기화 시켜주는 코드로 eor r7,r7,r7 를 사용했다. 같은 레지스터에 대해 exclusive or를 할 경우 모두 0으로 바뀌기 때문에 0으로 초기화 할 수 있다.

R1은 자릿수를 표현하기 위해 10이라는 숫자를 저장하였다. 이 레지스터의 활용은 두자리수 입력에서 자세히 다룬다.

R12는 음수 양수를 표현하기 위해 1이라는 숫자를 저장하였다.

이 또한 뒤에서 자세히 다룬다.

1. **두자리 이상 음수, 양수 입력 받기**

이를 구현한 코드는 아래와 같다.

텍스트, 스크린샷, 폰트, 영수증이(가) 표시된 사진

자동 생성된 설명

* 1. 두자리 이상 수 입력 받기

scan 으로 r0에 입력을 받아 sub r2, r0, #’0’으로 ASCII to integer로 변환해준다. 이후 mla r7, r1, r7, r2로 decimal number를 r7에 저장한다. 이때 이 코드의 의미는 r7 = r1 \* r7 + r2이고 위에서 r1 = 10을 저장했기 때문에 r1\*r7은 decimal number에서 r7의 자릿수를 늘려주는 역할을 한다. 첫 입력에선 r7에 아무것도 저장되어 있지 않기 때문에 r7 = r2가 되어 한자리 숫자가 입력이 되고 이후 입력되는 것에 의해서 십의 자리, 백의 자리, 천의 자리로 첫 입력받은 수가 이동하게 되고 새롭게 입력된 수는 계속해서 일의 자리를 차지하게 된다. 이는 mla 뒤에 b num1을 입력하여 iteration 돌게 하여 구현하였다.

이러한 입력은 space bar를 누를 때 까지 받게 된다. 이는 cmp r0, #’ ‘ // beq final\_num1으로 구현하였다. Space bar를 입력 받으면 beq 조건을 만족하게 되어 final\_num1으로 분기된다.

Final\_num1에서의 동작은 1-3에서 다룬다.

* 1. 음수 입력 받기

첫 입력으로 ‘ – ‘를 받는다면 이는 음수임을 의미한다. 하지만 ‘ – ‘는 컴퓨터가 받아들이기에 단순히 기호일 뿐 음수의 의미를 내포하고 있진 않다. 따라서 음수의 의미를 부여해주기 위해 따로 코드를 구현해야 한다. 이는 store\_sign1에서 역할을 수행해준다. Num1에서 입력으로 ‘ – ‘를 받았을 때 cmp r0, #'-' // beq store\_sign1 에 의해 store sign\_1로 분기 된다. 이때 -를 받으면 음수임을 표현하기 위해 r12에 -1을 저장한다. 첫 입력으로 ‘ – ‘가 입력되지 않는다면 r12값은 초기 설정 값인 1로 유지된다.

* 1. 최종 입력 값 r7 레지스터에 저장하기

num1, store\_sign1에서 절댓값, 부호 정보를 각각 r7, r12에 저장하였다. Final\_num1에서는 mul r3, r7, r12 // mov r7, r3를 통해 r7, r12에 저장된 값을 곱한 값을 최종적으로 r7에 저장한다. 이때, mul 코드는 같은 레지스터에 대해 사용할 수 없어 r3에 먼저 저장하고 이후 r7에 다시 값을 옮기는 작업을 수행했다. R7에 원하는 값을 저장후 r12값이 -1로 바뀌었을 수 있으니 다시 1로 초기화 시킨 후 다음 숫자를 입력 받기 위해 b num2 코드를 사용하였다.

이후 위 과정들을 반복하여 다섯개의 숫자를 r7- r11에 차례대로 저장한다.

1. 오름차순 정렬

오름차순은 Bubble Sort 개념을 이용해 구현하였다. 구현한 코드의 일부는 다음과 같다.

텍스트, 영수증, 공구, 디자인이(가) 표시된 사진

자동 생성된 설명cmp r7, r8 // bgt change1 코드를 활용해 r7<r8 이라면 다음 대소 관계 비교를 위해 AS2로 넘어간다. r7 > r8 인 경우 change1으로 분기한다. 이후 mov r12, r7 // mov r7, r8 // mov r8, r12 코드를 이용해 r7, r8에 저장된 값을 swap 한다. Swap 한 이후 다음 대소 관계 비교를 위해 AS2로 넘어간다.

이런 식으로 총 5개의 레지스터를 모두 상호 비교해준다.

{ (r7, r8) } , { (r8, r9), (r7, r8) } , { (r9, r10), (r8, r9), (r7, r8) }, { (r10, r11), (r9, r10), (r8, r9), (r7, r8) } 순서로 총 10번 비교를 하여 위와 같은 과정을 거쳐준다면 r7-r11에 오름차순으로 정렬이 된다.

1. 정렬된 숫자의 출력

텍스트, 스크린샷, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명

정렬된 숫자를 console 창에 출력하기 위해선 두가지를 고려해야 한다. 두 자리 수 출력, 음수 출력. 이를 차근차근 나눠서 설명해보겠다.

3-1 두 자리 수 출력하기

레지스터에 저장된 두 자리 수를 한번에 console 창에 출력할 수 없기에, 이를 위해 나눗셈의 개념이 필요하다. 나눗셈을 이용해서 십의 자리, 일의자리를 따로 출력을 하여 마치 두 자리 수가 출력이 된 것처럼 보이게 하는 것이다. 아쉽게도 arm assembly에선 나눗셈 기능을 제공하지 않아 직접 나눗셈 모듈을 만들어 사용하였다. 사용한 나눗셈 모듈은 다음과 같다.

3-1-1 나눗셈 기능 구현

텍스트, 영수증, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명곱셈이 더하기의 반복이듯 나눗셈은 뺄셈의 반복이다. 그래서 나누기의 코드는 다음과 같다.

피제수에서 제수를 빼는 과정을 반복하면서 1번 반복할 때마다 몫을 1씩 올린다.

피제수가 제수보다 작아지면 반복은 끝나게 되고 남은 것은 나머지에 해당하게 된다.

결국 r1 / r2 = r0…r1 과 같은 결과가 나오게 된다.

이를 위 코드와 결합하여 여러 자리 숫자 출력을 구현할 수 있게 된다.

3-1-2 두 자리 수 출력

Print\_integer에서 스택프레임을 할당하여 함수가 시작된다. 스택프레임 할당 후 print\_integer1으로 분기한다. 출력하고 싶은 register가 r0에 존재한다. 이때 r0에 존재한 숫자가 한자리 수라면 cmp r0, #10 // blt print\_integer\_2 에 의해 print\_integer\_2로 분기한다.

print\_integer\_2로 분기한 숫자는 한자리 정수이기 때문에 그냥 print\_char를 통해 출력하게 된다.

R0 에 존재하는 숫자가 두 자리 수라면 b loop\_div에 의해 loop\_div 로 분기한다. 여기서 부터 3-1-1에서 설명한 나눗셈 모듈을 활용한다. 나눗셈 모듈에서 r1/r2 를 수행하므로, r0에 저장되어 있는 나누고자 하는 숫자를 r1에 저장해준다. R2에는 10을 저장하여 r1/10을 수행할 수 있는 환경 세팅 후 div\_mod로 분기하여 나눗셈을 실행한다. 나눗셈 실행 후 십의 자리 숫자는 r0, 일의 자리 숫자는 r1에 저장된다. Loop\_div로 다시 돌아와서 add r0, r0, #'0'를 통해 integer to ASCII로 변환해주고 r0(십의자리 숫자)를 출력한다. 이후 r1에 저장된 일의 자리 숫자도 위 과정과 동일하게 반복하여 준다. 이때 print\_char를 수행할 때 r1레지스터에 주소값이 저장되어 r1에 있어야 할 일의자리 숫자 정보가 사라져 임시로 r4 레지스터에 이동하였다가 r0으로 이동하여 print\_char로 출력하였다.

이러한 과정을 통해 십의 자리 숫자, 일의 자리 숫자를 차례대로 출력한 후 print\_integer\_end로 분기하여 스택프레임을 해체 하고 pc값에 저장된 곳으로 돌아간다.

3-2 음수출력

양의 정수만 출력하는 것이라면 위 3-1 과정들을 통해 구현이 마무리 되겠지만, 우리는 입력을 받을 때 음수도 입력을 받았기 때문에 출력에서도 마찬가지로 음수 출력 기능을 구현해야 한다. 여기서 음수를 ‘출력’하는 데에 초점을 맞췄고, 출력 이후 해당 값을 계속해서 저장해두진 않았다. 혹, 출력 이후에도 해당 값이 필요하다면 잠시 다른 곳에 저장해두는 코드를 추가하면 될 것이다.

음수를 출력해야 한다면 print\_integer1에서 cmp r0, #0 // blt make\_pos 를 통해 r0<0이라면 make\_pos로 분기한다. make\_pos코드는 다음과 같다.

텍스트, 영수증, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명음수의 ‘ – ‘ 기호와 절댓값을 따로 떼어 내서 생각했다.

음수라는 것을 인식해서 make\_pos로 왔기 때문에 일단 cosole 창에 ‘ – ‘ 를 출력하였다. ‘ – ‘를 출력하였다면 바로 뒤에 이어서 절댓값을 출력해주면 된다.

이를 위해 r7에 0xFFFFFFFF를 저장하고 r0와 r7을 exclusive or연산하여 r0의 보수를 구해준 뒤, 1을 더해 r0 에 저장되어 있던 음수 값을 양수로 변환해주었다. 즉, 절댓값을 씌워준 것이다. 이후 print\_integer1으로 분기하여 3-1에서 설명한 과정을 거쳐 정수의 절댓값이 출력된다.

console창에 ‘ – ‘부호 출력 후 곧바로 절댓값을 출력했기 때문에 마치 음수가 출력된 것 처럼 보이게 된다.

위 과정들을 통해 오름차순으로 정렬하는 코드를 구현했고, 여러가지 예시들을 수행해본 결과는 아래와 같다.

윗줄이 입력, 아랫줄이 출력이다.

한자리텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명 양수에 대해서도 정상 작동하며,

두자리 음수가 혼합되어 있는 수들의 조합에서도 정상 작동함을 알 수 있다.

마지막은 ppt예제 이외에 다른 숫자들을 입력하여 예제 이외에도 잘 돌아가는 지 확인을 해보았다.