# Room Number

이 문제는 다음과 같다.

* 철수는 새로 이사를 했다. 자기 방 번호를 플라스틱 숫자 카드를 구매해서 문에 붙이려고 한다.
* 플라스틱 숫자 카드는 한 set로 판다. 한 set에는 0번부터 9번까지 숫자가 하나씩 들어있다.
* 철수의 방 번호가 주어졌을 때, 필요한 set의 개수의 최소값을 출력하시오.
* 6은 9를 뒤집어서 이용할 수 있고, 9는 6을 뒤집어서 이용할 수 있다.

이 문제에서 고려해야 하는 점은 다음과 같다.

1. 0~9까지 숫자 각각의 총 입력 개수
2. 6과 9는 같은 취급하여 6,9 총 입력 개수의 절반으로 필요 세트수 고려.
3. 각 숫자들의 총 입력 개수 중 최대값을 비교하여 최종적으로 필요한 세트수 찾기

고려사항 1,2 를 풀기 위해 숫자의 개수를 저장할 레지스터 총 9개를 설정하였다.

숫자 0의 개수는 레지스터 r9에, 숫자 1-8의 개수는 레지스터 r1-r8에, 9는 6과 함께 레지스터 r6에 저장하였다.

해당 레지스터들을 사용하기 위해 eor를 사용하여 초기화 시켰고 이는 ascending sort에서 자세히 설명했다.

텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명고려사항들을 어떻게 해결했는지 코드와 함께 자세히 설명해보겠다.

1. 0~9까지 숫자 각각의 총 입력 개수 count

Roomnumber는 순서대로 하나씩 입력할 것이기 때문에 입력을 함과 동시에 해당 숫자의 개수를 count 하는 code를 구현하였다.

bl scan으로 입력을 받은 후 bl print\_char 로 입력 받은 값을 즉시 출력한다. 이렇게 입력이 반복되면 console 창에서 roomnumber로 보이게 된다.

cmp r0, #’0’ // addeq r9, r9, #1을 통해 0을 입력 받았다면 r9 레지스터에 존재하는 값에 1을 더해 다시 r9 레지스터에 값을 저장해준다. 이러한 코드를 1,2,3,4,….를 입력받았을 때도 마찬가지로 count 되도록 오른쪽과 같이 코드를 작성해준다.

입력이 종료될 때 까지 b start에 의해 해당 과정이 iteration되므로 room number를 계속해서 입력한다. 입력하는 즉시 출력, count가 반복되면서 해당 숫자들이 몇 개인지 각 레지스터에 저장한다.

room number 입력이 끝나면 space bar를 눌러 cmp r0, #’ ‘ // beq set 69에 의해 set69로 분기된다.

1. 6과 9는 같은 취급하여 6,9 총 입력 개수의 절반으로 필요 세트수 고려.

6입력, 9입력시 모두 r6 레지스터에 count를 했다. 숫자 한 set에 6,9는 총 2개가 있으므로 6,9가 2개 있다면 1세트만 있어도 되는 것이다. 만약 6,9가 총 3개 or 4개 있다면 2set가 필요하고 5개 or 6개 가 있따면 3set가 필요하다. 이를 위해서 총 6,9의 개수가 홀수 일 때, 짝수일 때를 고려하여 2를 나눈 뒤 필요 set수를 고려해야 한다.

이 과정 또한 ascending sort, find prime number에서 사용한 div\_mod를 사용해서 구현할 수 있겠지만, 이 경우는 특별하게도 제수가 2이기 때문에 shift right 개념으로 접근할 수 있다. 이때, mov r6, r6, LSR #1 코드를 사용해 한칸 shift right를 진행했고 이는 r6 / 2 를 의미한다. shift right를 통해 나온 결과는 정수부분만 존재하고 소수부분은 없어지기 때문에 나누기 전 r6에 저장된 값에 1을 더해 shift right를 진행했다.

이렇게 할 경우 r6에 저장된 값이 홀수, 짝수임을 굳이 구별할 필요가 없어진다.

만약 r6에 저장된 값이 3이라면 ( 3 + 1 ) / 2 = 2

r6에 저장된 값이 4라면 (4 + 1) / 2 = 2.5 = 2

이와 같이 우리가 원하는 결과(필요 set 수) 를 얻어낼 수 있다.

텍스트, 폰트, 화이트, 공구이(가) 표시된 사진

자동 생성된 설명이를 실제 코드로 구현한 것은 오른쪽과 같다.

먼저 1을 더해주고, 2로 나눈다.

그럼 우리가 원하는 값을 얻어낼 수 있다.

이 작업을 마친다면 필요한 밑작업은 끝났다.

이제 최종적으로 필요한 set의 개수가 몇 개인지 확인하기 위해 b needset에 의해 needset으로 분기한다.

1. 각 숫자들의 입력 개수를 통해 필요한 set수를 비교하여 최대 필요 set수 찾기

maximum 값을 찾기 위해서 앞에서부터 차례대로 필요 set수를 비교하였다. 이때 필요한 maximum 값을 r9에 최신화 시키며 총 8번의 비교를 통해 필요 set수를 r9에 저장하였다.

오른쪽과 같은 코드를 활용해 r9에 저장된 값, r1에 저장된 값을 비교하여 r9에 저장된 값이 크다면 그 값을 유지하고, r1에 저장된 값이 크다면 r1에 저장된 값을 r9에 저장하여 최신화 시켜준다.

텍스트, 영수증이(가) 표시된 사진

자동 생성된 설명이 과정을 (r9, r2), (r9, r3), (r9, r4), … , (r9, r8)에 대해 모두 수행해주는 것이다. 이후 r9에 저장된 값을 출력하기 위해 print\_needset으로 분기하여 출력을 한다.

이때 사용한 함수는 print\_integer로 ascending sort에서 설명했기에 자세한 설명은 하지 않도록 하겠다.

오른쪽은 위 설명들을 정리한 코드이다.

텍스트, 폰트, 라인, 영수증이(가) 표시된 사진

자동 생성된 설명위 과정들을 통해 방번호를 표시하는 데 필요한 카드 set 수를 찾는 코드를 구현했고, 여러가지 예시들을 수행해본 결과는 아래와 같다.

왼쪽이 방번호, 오른쪽이 필요한 set 수이다.

첫번째는 1을 10번 입력하여 10set가 필요하고

두번째는 6,9를 총 26번 입력하여 13set가 필요하다.

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

이때, 필요 set수 99개 까지 구현 가능하다.