

1.Algorithm

We have two String ,denote it by A and B. We will check it char by char ,First run check A[i], if A[i] belong to a-z,then add BUCKET[A[i]],means the character A[i] count ++,if A[i] belong to A-Z,then convert it to a-z by plus x20, Second run do the same except BUCKET[A[i]]--.Finally We check BUCKET from a-z whether its all count == 0,if so ,then print YES ,else NO

```
char * p = A
while(*p != '\0') {
    bucket[*p]++
    p++
}
p=B
while(*p != '\0'){
    bucket[*p]++
    p++
}
i=bucket
for(i='a';i!=bucketend;i++)
if(*i!=0){
    print "NO"
    break;
}
print "YES"
```

2.Essential parts of code with sufficient comments

```
LDI R1,aptr;
LEA R3,BUCKET; R3 = BUCKET 's array first address
LD R7,UP;
LOOP1
    LDR R0,R1,#0;    R0 = *p
    BRZ LOOP1END; if *p == 0 then next loop
    LD R6,MASK1 ; check ==x40
    AND R2,R6,R0    ;check 'A' <= *P <= 'Z'
    BRZ ADDYES1    ;ifso *p = *p + x20
    LD R6,MASK2
    AND R2,R6,R0
    BRZ ADDYES1
NEXT1
    ADD R5,R3,R0;    get BUCKET + R0
    LDR R4,R5,#0    ;    R4 = letter[*p]
    ADD R4,R4,#1;    R4 ++
    STR R4,R5,#0;    ST R4 -> letter[*p]
    ADD R1,R1,#1    ;p++
    BR LOOP1;
```

```

ADDYES1
    ADD R0,R7,R0 ;;
    BR NEXT1
LOOP1END

LOOP2 ... ;omit Like Loop1 except ADD R4,R4,#-1

LD R2,BUCKEToff ;BUCKEToff is used to posite 'a',it's 96 away from &BUCKET
ADD R1,R3,R2; p = BUCKET ; WE check from Bucket['a'],and now R1 = &Bucket['a']
LOOP3
    LEA R5,BUCKETEND ;if temporary positon is BUCKETEND,then print YES
    NOT R5,R5
    ADD R5,R5,#1
    ADD R5,R5,R1 ;R5 = p-BUCKETEND
    BRZ YES ;ALL a-z has been checked no problem
    LDR R0,R1,#0 ;R0 = *p
    BRnp NO ;if R0 is not 0,means there's different in A and B,print NO
    ADD R1,R1,#1; p++
    BR LOOP3
YES ;answer output
    LEA R0,YESstr
    TRAP x22;
    HALT
NO
    LEA R0,NOstr
    TRAP x22;
    HALT
...
BUCKEToff .FILL #96 ;BUCKET create
BUCKET .BLKW #124
BUCKETEND .FILL x0fff
...

```

3. Questions

1. How do you print YES or NO

I use YESstr .STRINGZ "YES" ; NOstr .STRINGZ "NO" store the output previously and then use LEA R0 YESstr ; Trap x22 to print it

2. How do you implement a BUCKET

I was directly creating a large array of buckets to use. BLKW, and add a tag BUCKETEND after it, because I am traversing two strings to the BUCKET read and write process directly use the ASCII letter as the corresponding address value, so here the BUCKET subscript corresponds to the ASCII table subscript.

I'm going to start at 97, which is lowercase 'a', and break when the current address p==BUCKETEND

The advantage of this is that since I will convert uppercase letters to lowercase letters, and only see if the 26 lowercase letters' positions are 0, So it can ignore the illegal input like ' ' or others. The disadvantage is that this is a waste of memory, because the space before 96 is actually not needed