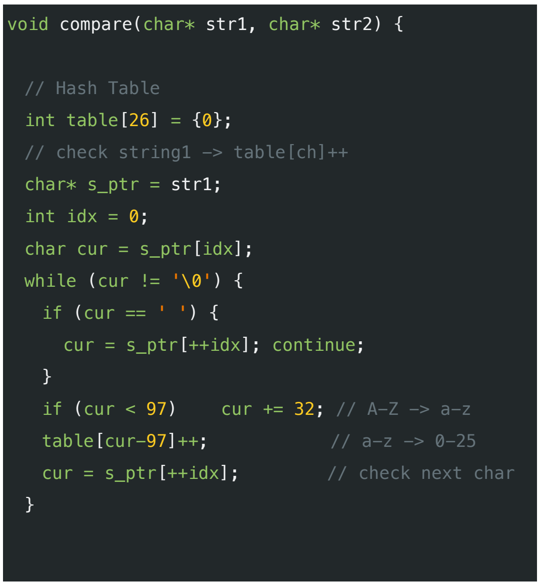
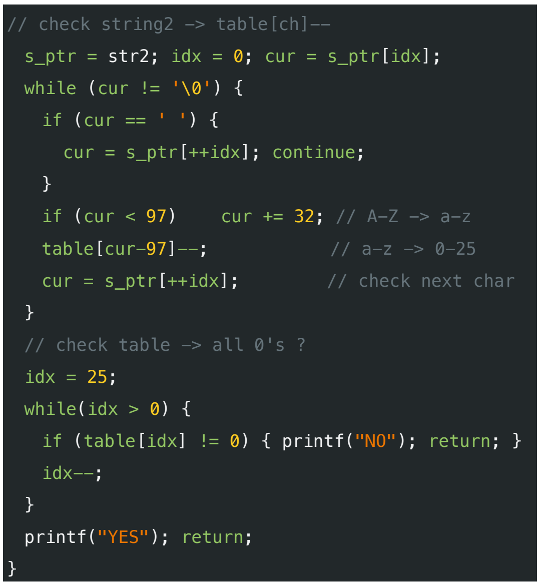
Report for LAB-2

1 Algorithm: How to solve the problem?



```
void compare(char* str1, char* str2) {

  // Hash Table
  int table[26] = {0};
  // check string1 -> table[ch]++
  char* s_ptr = str1;
  int idx = 0;
  char cur = s_ptr[idx];
  while (cur != '\0') {
    if (cur == ' ') {
      cur = s_ptr[++idx]; continue;
    }
    if (cur < 97)    cur += 32; // A-Z -> a-z
    table[cur-97]++;             // a-z -> 0-25
    cur = s_ptr[++idx];          // check next char
  }
```



```
// check string2 -> table[ch]--
  s_ptr = str2; idx = 0; cur = s_ptr[idx];
  while (cur != '\0') {
    if (cur == ' ') {
      cur = s_ptr[++idx]; continue;
    }
    if (cur < 97)    cur += 32; // A-Z -> a-z
    table[cur-97]--;             // a-z -> 0-25
    cur = s_ptr[++idx];          // check next char
  }
  // check table -> all 0's ?
  idx = 25;
  while(idx > 0) {
    if (table[idx] != 0) { printf("NO"); return; }
    idx--;
  }
  printf("YES"); return;
}
```

2 Code (with comments)

```
        ; main program
        .ORIG   x3000;
        LEA R3, Table;
        ; check str1
        LD  R0, Addr1;
        LDR R0, R0, #0;    R0 = &str1[0]
Loop1   LDR R1, R0, #0;    R1 = str1[i]
        BRz Fin1;          str1[i] == '\0'
        ADD R0, R0, #1;    i++;
        ; if ' '
        LD  R2, NegSp;
        ADD R2, R1, R2;
        BRz Loop1;
        ; elif A~Z
        LD  R2, Nega;
        ADD R1, R1, R2;
        BRzp Count;
        LD  R2, offset;
        ADD R1, R1, R2;
        ; table[ch] ++
Count   ADD R2, R1, R3;    R2 = &table[ch]
        LDR R1, R2, #0;
        ADD R1, R1, #1;
        STR R1, R2, #0;
        BRnzp Loop1;
        ; check str2
Fin1     LD  R0, Addr2;
```

```
         LDR R0, R0, #0; R0 = &str2[0]
Loop2    LDR R1, R0, #0; R1 = str2[i]
         BRz Fin2;       str2[i] == '\0'
         ADD R0, R0, #1;
        ; if ' '
        LD  R2, NegSp;
        ADD R2, R1, R2;
        Brz Loop2;
        ; elif A~Z
        LD  R2, Nega;
        ADD R1, R1, R2;
        Brzp Dcount;
        LD  R2, offset;
        ADD R1, R1, R2;
        ; table[ch]--
Dcount   ADD R2, R1, R3;
        LDR R1, R2, #0;
        ADD R1, R1, #-1;
        STR R1, R2, #0;
        BRnzp Loop2;
        ; check table[0~25]
Fin2     LD  R1, len;
Check    ADD R2, R1, R3;
         LDR R2, R2, #0;
         BRnp Fail;
         ADD R1, R1, #-1;
         BRzp Check;
```

```
      ;
Success LEA R0, YES;
        PUTS;
        BRnzp Fin;
        ;
Fail    LEA R0, NO;
        PUTS;
Fin     HALT;
; boundary
NegSp   .FILL #-32;    -asc(' ')
Nega    .FILL #-97;    =asc('a')
offset  .FILL #32;
; result string
YES     .STRINGZ "YES";
NO      .STRINGZ "NO";
; addr
Addr1   .FILL x4000;
Addr2   .FILL x4001;
; table for each letter
len     .FILL 25;
Table   .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
```

```
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .FILL #0;
       .END;

       ; load strings
       .ORIG x4000;
       .FILL Str1;        init addr of str1
       .FILL Str2;        init addr of str2
Str1   .STRINGZ "dirty room";
Str2   .STRINGZ "dormitory";
       .END;
```

3 Q & A

1. **How to solve the problem?**

   Shown in section 1 & 2.

2. **How to get the address of strings?**

   Since the offset is too long, we should complete this in two steps. Take string 1 for example:

```
LD  R0, Addr1;       R0 = x4000 = &(&str)
LDR R0, R0, #0;      R0 = Mem[x4000] = &str
Addr1 .FILL x4000;
```