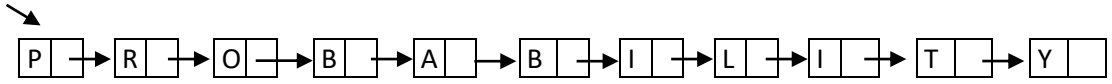
Algorithms and Programming

Write a program that determines whether two ***single-linked lists*** are identical or not.
If they are identical, the program should print matched letters.

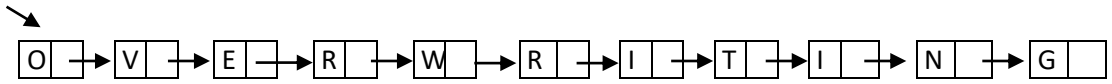
Two single-linked lists are *identical* if they have the same pattern of letter repetitions.

Example:
probability and *overwriting* are identical.

head



head



Matched letters:

```
P ↔ O
R ↔ V
O ↔ E
B ↔ R
A ↔ W
L ↔ T
I ↔ I
T ↔ N
Y ↔ G
```

Examples:
murmur and *tartar* are identical, because they both have the same pattern "abcabc".

"seneye gider" and "zamana bırak" are identical.

AAABBCbbaaa and 11122322111 are identical

Matched letters:

```
A ↔ 1
B ↔ 2
C ↔ 3
```

Write a *Single-Linked List class* that includes necessary operations such as;

- *add()*
- *display()*
- *etc.*

Write a *main program* that takes two strings from the user and inserts each character of them into two different Single-Linked Lists.
After that, the program may call a procedure or a function to compare two single-linked lists and determine whether they are identical or not.

You must use ONLY single linked list. Don't use other data structures such as array, stack, queue etc.

Don't use string in the solution part of the program. You can use "**string**" ONLY at the beginning of the program (when taking sentences from the user and inserting into single linked lists).