

Project 1 Drawings

First Parts

This example is provided to assist you in understanding the drawings requirement in Project 1 as well as to provide a template by which you can easily organize your work.

You may **NOT** use the provided sample data file for this assignment.

Any submission using this data file will receive a grade of 0 for the entire project.

Sample File

WordRec Blueprint

token	string
count	int

```
Hockey Golf
Football Basketball
Football
Baseball Shotput Football
```

Final Result of Processing the Data: Filled Five element Array **capacity** is 5

0	1	2	3	4	wordCount
Baseball	Basketball	Football	Golf	Hockey	5
1	1	3	1	1	

*Note, Alphabetical

Array, Initially

0	1	2	3	4	wordCount
					0

Filling the Array

Hockey is read in first. Search returns 0, equals `wordCount`. This is Case 1.

Place *Hockey* (count=1) at end of array, in index `wordcount`, and increment `wordCount`

0	1	2	3	4	wordcount
Hockey					⊕ 1
1					

Next token is *Golf*. Search returns 0, and *Golf* != *Hockey*. This is Case 3.

Everything to the right of Index 0 must be moved over (working right to left) to make room for *Golf*

Move *Hockey* to clear Index 0; Place *Golf* in index 0 and increment `wordCount`

0	1	2	3	4	Wordcount
Golf	Hockey				⊕ 2
1	1				

Next token is *Football*. Search returns 0, and *Golf* != *Football*. This is Case 3.

Project 1 Drawings

First Parts

Everything to the right of Index 0 must be moved over (working right to left) to make room for *Football*
Move *Hockey* to Index 2; Move *Golf* to Index 1;
Place *Football* in index 0 and increment wordCount

0	1	2	3	4
Football	Golf	Hockey		
1	1	1		

Wordcount
2 3

Next token is *Basketball*. Search returns 0, and *Basketball* != *Football*. This is Case 3.

Everything to the right of Index 0 must be moved over (working right to left) to make room for *Basketball*
Move *Hockey* to Index 3; Move *Golf* to Index 2; Move *Football* to Index 1
Place *Basketball* in index 0 and increment wordCount

0	1	2	3	4
Basketball	Football	Golf	Hockey	
1	1	1	1	

Wordcount
3 4

Next token is *Football*. Search returns 1 and *Football* == *Football*. This is Case 2.

Increment the counter in Index 1 and DO NOT increment wordCount

0	1	2	3	4
Basketball	Football	Golf	Hockey	
1	2	1	1	

Wordcount
3 4

Next token is *Baseball*. Search returns 0, and *Baseball* != *Football*. This is Case 3.

Everything to the right of Index 0 must be moved over (working right to left) to make room for *Baseball*
Move *Hockey* to Index 4; Move *Golf* to Index 3; Move *Football* to Index 2, *Basketball* to Index 1
Place *Baseball* in index 0 and increment wordcount Array is now full – must continue

0	1	2	3	4
Baseball	Basketball	Football	Golf	Hockey
1	1	2	1	1

Wordcount
4 5

Next token is *Shotput*. Because the array is full, we only check if search returns an index whose **token** member equals the new token (so we can increment its counter) Search returns 5, and since this equals capacity, there is no room for the token. It is discarded. **Do NOT** increment wordCount

0	1	2	3	4
Baseball	Basketball	Football	Golf	Hockey
1	1	2	1	1

Wordcount
5

Next (and last) token is *Football*. Search returns 2 and *Football* == *Football*. This is Case 2.

Increment the counter in Index 2 and DO NOT increment wordCount

0	1	2	3	4
Baseball	Basketball	Football	Golf	Hockey
1	1	3	1	1

Wordcount
5

Project 1 Drawings

First Parts