HW8 Lookalike

Name:_____

1. For the word “TORPOR”, build the bad symbol table:

…	O	P	…	R	…	T	…	
6	1	2	6	3	6	5	6	

2. For the word “TORPOR”, build the good suffix table (it doesn’t have to be in table form):

*	d_2=	
OR	d_2(k=1)=	6
POR	d_2(k=2)=	3
RPOR	d_2(k=3)=	6
ORPOR	d_2(k=4)=	6
TORPOR	d_2(k=5)=	6

(* this isn’t part of the answer -- just a way to compare with ch7goodsuffix.pptx on dropbox)

3. Using Horspool’s, scan the text “HODORFROMMORDORISINTORPOR” for “TORPOR”, using the above tables (for the Test/HW, I will give you the tables for this type of question.

H	O	D	O	R	F	R	O	M	M	O	R	D	O	R	I	S	I	N	T	O	R	P	O	R	
T	O	R	P	O	R	a																			
						T	O	R	P	O	R	b													
									T	O	R	P	O	R	c										
												T	O	R	P	O	R	d							
																		T	O	R	P	O	R	e	
																			T	O	R	P	O	R	f

Horspool’s doesn’t use good shift table and only use last character

a) F!=R, BadTable(F) = 6 (‘F’ not in ‘TORPOR’)
b) M!=P, BadTable(R) = 3
c) D!=P, BadTable(R)=3
d) R!=I, BadTable(I) = 6 (‘I’ not in ‘TORPOR’)
e) R!=0, BadTable(O) = 1
f) Match!

4. Using Boyer-Moore, scan the text “HODORFROMMORDORISINTORPOR” for “TORPOR”, using the above tables (for the Test/HW, I will give you the tables for this type of question).

H	O	D	O	R	F	R	O	M	M	O	R	D	O	R	I	S	I	N	T	O	R	P	O	R	
T	O	R	P	O	R	a																			
						T	O	R	P	O	R	b													
										T	O	R	P	O	R	c									
																T	O	R	P	O	R	d			
																			T	O	R	P	O	R	e

(notice that we needed one less shift)

a) F!=R, BadTable(F) = 6 ('F' not in 'TORPOR'), k=0 so no good table
b) M!=P, k =2, d_1 = BadTable(M) – k = 4, $d_2(2) = 3$, d = max(d_1,d_2) = 4
c) I!=R, k=0, d = BadTable(I) = 6
d) T!=P, k=2, d1 =BadTable(T) -k = 5-2 = 3, d2(2) = 3, d = max(d_1,d_2) = 3
e) Match!

5. Insert the following keys into the following hash table, size 10:
   a. Our hash function is simply *key* **mod 10**
   b. Assume **Open hashing**

Keys: 453 909 414 895 9012 813 9999 411

Index	bucket
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Answer:

Index	bucket
0	
1	411
2	9012
3	453 → 813
4	414
5	895
6	
7	
8	
9	909 → 9999

6. Insert the following keys into the following hash table, size 10:
    a. Our hash function is simply *key* **mod 10**
    b. Assume **Closed Hashing** with **Linear Probing** as a **Collision Strategy**

Keys: 453 909 414 895 9012 813 9999 411

Index	bucket
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Answer:

Index	bucket
0	9999
1	411
2	9012
3	453
4	414
5	895
6	813
7	
8	
9	909