Name: RITHIKA KATHIRVEL
Roll number: B200055CS
Batch: B
Question: 1

```
//pseudocode for hash_func_h(w,n)
Int len;
Int word=0;
Int index
len=strlen(w)
if(len==1){
word=w[0];
index=(2*w)%
}
Else{
word=w[0]+w[1]+w[len-1]+w[len];
index=(word/len)%n
}
Return index
```

```
//pseudocode for hash_func_g(w,n)
Int len;
Int word=0;
Int index;
len=strlen(w)
for(int i=0;i<=len;i++){
   word=word+w[i];
}
index=(word/len)%n
Return index
```

```
//pseudocode for calculating_linked_list_length(H[i].head)
 if(H[i].head==NULL) return 0
Else{
Int k=0;
Struct node* temp;
temp=H[i].head;
while(temp->next_node!=NULL){
k=k+1
temp=temp->next_node
}
Return k
}
```

```
//pseudocode for store_word(H,w,n)
Int k1,k2;
k1=hash_func_h(w,n)
k2=hash_func_g(w,n)
Int len_chain;
Int len_chain_1;
len_chain=calculate_linked_list_length(H[k1].head)
len_chain_1=calculate_linked_list_kength(H[k2].head)
if(len_chain<len_chain_1){
   insert_in_linked_list(H[k1].head,w)

}
Else if(len_chain_1<len_chain)  insert_in_linked_list(H[k2].head,w)
Else if( len_chain==len_chain_1)  insert_in_linked_list(H[k1].head,w)
```

```
//pseudocode for hash_func_h(w,n)
Int len;
Int word=0;
Int index
len=strlen(w)
if(len==1){
word=w[0];
index=(2*w)%
}
Else{
word=w[0]+w[1]+w[len-1]+w[len];
index=(word/len)%n
}
Return index
```

```
//pseudocode for find_hash1(H,w,n)
Int k1;
k1=hash_func_h(w,n)
 return k1
```

```
//pseudocode for hash_func_g(w,n)
Int len;
Int word=0;
Int index;
len=strlen(w)
for(int i=0;i<=len;i++){
   word=word+w[i];
}
index=(word/len)%n
Return index
```

```
//pseudocode for find_hash2(H,w,n)
Int k2;
k2=hash_func_g(w,n)
 return k2
```

```
//pseudocode print_linked_list(H[i].head)
if(H[i].head==NULL) print NULL
Else{
Struct node* temp;
temp=H[i].head;
while(temp!=NULL){
print(“%s”,temp->word)
temp=temp->next_node
}
printf(“\n”)
```

```
//pseudocode for print_table(H,w,n)
for(int i=0;i<n;i++){
 if(H[i].head==NULL)
  printf(“NULL\n”)
 Else{
  print_linked_list(H[i].head);
 }
}
```