20/11/2020

**Experiment No:17**

**DOUBLY LINKED LIST**

**AIM**:

Create a Doubly Linked List from a string taking each character from the string. Check if the given string is palindrome in an efficient method.

**DATA STRUCTURES USED:**

Doubly Linked List

**ALGORITHM:**

Algorithm INSERT(ITEM)

1. new= GetNodes(Node)
2. new->DATA=ITEM
3. new->lLINK=NULL
4. new->rLINK=NULL
5. if (new = NULL) then
6. print”memory underflow”
7. Exit
8. Else
9. If (FRONT->rLINK=NULL)
10. FRONT->rLINK=new
11. REAR->lLINK=new
12. new->lLINK=FRONT
13. new->rLINK=REAR
14. else
15. REAR->lLINK->rLINK= new
16. REAR->lLINK=new
17. new->lLINK=REAR->lLINK
18. new->rLINK=REAR
19. EndIf
20. EndIf
21. Stop

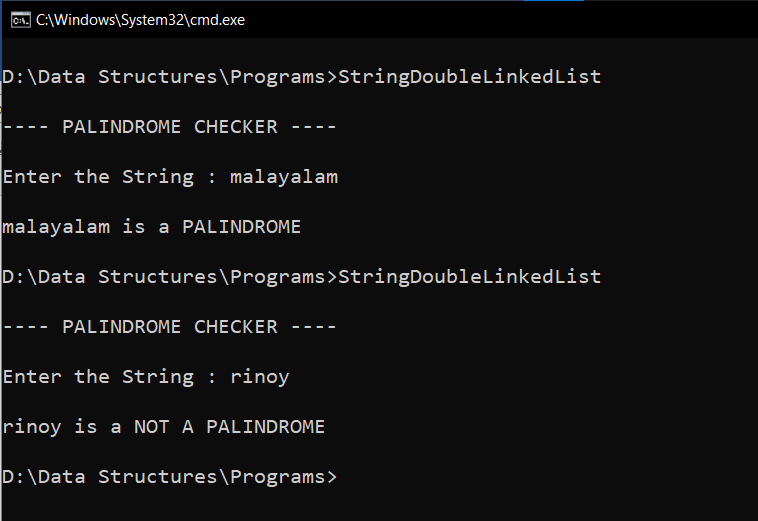
Algorithm CHECK\_PALINDROME()

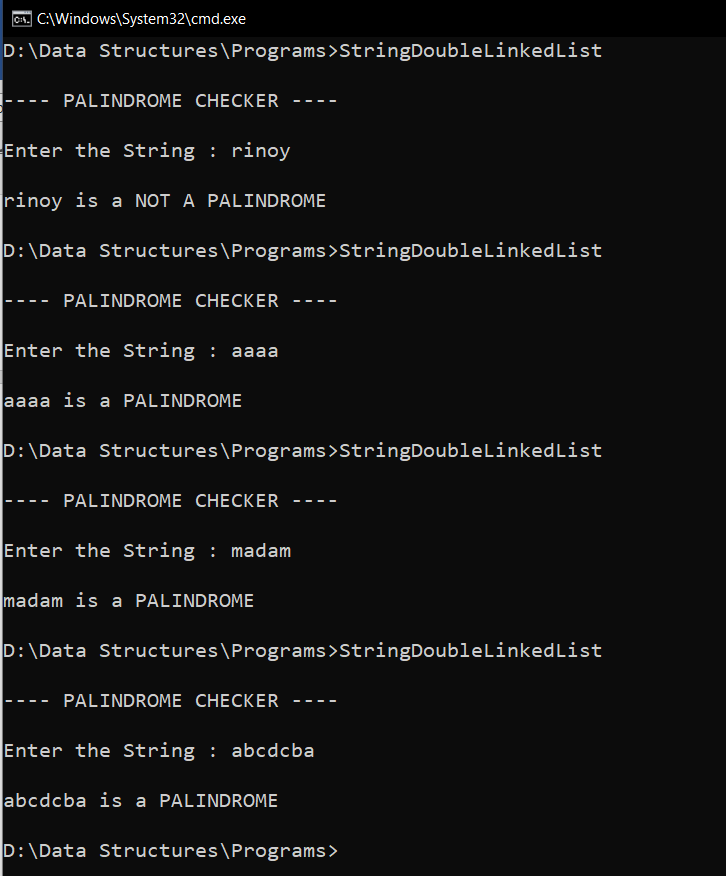
1. ptr1=FRONT
2. ptr2=REAR
3. while(ptr1!=ptr2)
4. if(ptr1->DATA!=ptr2->DATA)
5. Return 0
6. EndIf
7. Ptr1=ptr1->lLINK
8. Ptr2=ptr2->lLINK
9. EndWhile
10. Return 1

**PROGRAM:**

#include<stdio.h>  
#include<stdlib.h>  
struct node{  
 char data;  
 struct node \*rlink;  
 struct node \*llink;  
};  
  
void insert(struct node\* front,struct node\* rear,char x){  
 struct node\* new = (struct node\*)malloc(sizeof(struct node));  
 new->data=x;  
 new->rlink=NULL;  
 new->llink=NULL;  
 if(new==NULL){  
 printf("\nMEMORY Underflow\n");  
 }else{  
 if(front->rlink==NULL){  
 front->rlink=new;  
 new->llink=front;  
 rear->llink=new;  
 new->rlink=rear;  
 }else{  
 new->llink=rear->llink;  
 rear->llink->rlink=new;  
 new->rlink=rear;  
 rear->llink=new;  
 }  
 }  
}  
  
  
int check\_palindrome(struct node\* front,struct node\* rear){  
 struct node\* ptr1=front;  
 struct node\* ptr2=rear;  
 while(ptr1!=ptr2){  
 if(ptr1->data!=ptr2->data){  
 return 0;  
 }  
 ptr1=ptr1->rlink;  
 ptr2=ptr2->llink;  
 }  
 return 1;  
}  
  
void main(){  
 char string[50],\*arr;  
 struct node\* front = (struct node\*)malloc(sizeof(struct node));  
 struct node\* rear = (struct node\*)malloc(sizeof(struct node));  
 front->rlink=NULL;  
 front->llink=NULL;  
 rear->rlink=NULL;  
 rear->llink=NULL;  
 printf("\n---- PALINDROME CHECKER ----\n");  
 printf("\nEnter the String : ");  
 scanf("%s",string);  
 arr=string;  
 while(\*arr!='\0'){  
 insert(front,rear,\*arr);  
 arr++;  
 }  
 if(check\_palindrome(front,rear)){  
 printf("\n%s is a PALINDROME \n",string);  
 }else{  
 printf("\n%s is a NOT A PALINDROME \n",string);  
 }  
 }

**OUTPUT:**





**RESULT:**

The given string was checked for palindromes using Doubly Linked List.