20/11/2020

**Experiment No:16**

**STUDENT LINKED LIST**

**AIM:**

The details of Student (Roll Number, Name, Total-Mark) are to be stored in a linked list. Write functions for the following operations:

1.Insert

2.Delete

3.Search

4.Sort on the basis of Roll Number.

5.Display the resultant list after every operation.

**DATA STRUCTURES USED:**

Linked List

**ALGORITHM:**

Algorithm INSERT ()

1. new= GetNode(Node)
2. //Input the details and Initialize it to the node
3. if (new = NULL) then
4. print”memory underflow”
5. Exit
6. Else
7. ptr=HEADER
8. While(ptr-> LINK!=NULL)do
9. ptr=ptr->LINK
10. Endwhile
11. ptr->LINK= new
12. new->LINK=NULL
13. Stop

Algorithm DELETE(KEY)

1. ptr1=HEADER
2. ptr=ptr1->LINK
3. while(ptr!=NULL)
4. if(ptr->rollno!= KEY)
5. ptr1=ptr
6. ptr=ptr->LINK
7. else
8. ptr1->LINK=ptr->LINK
9. ReturnNode(ptr)
10. Exit
11. Endif
12. Endwhile
13. If ptr=NULL
14. Print”Roll No Searched doesn’t exist”
15. Endif
16. Stop

Algorithm SEARCH(KEY)

1. ptr=HEADER->LINK
2. while(ptr!=NULL)
3. if(ptr->rollno!= KEY)
4. // Display the Details of the Node
5. Endif
6. Endwhile
7. If (ptr=NULL)
8. Print”Node with key doesn’t exist”
9. Endif
10. Stop

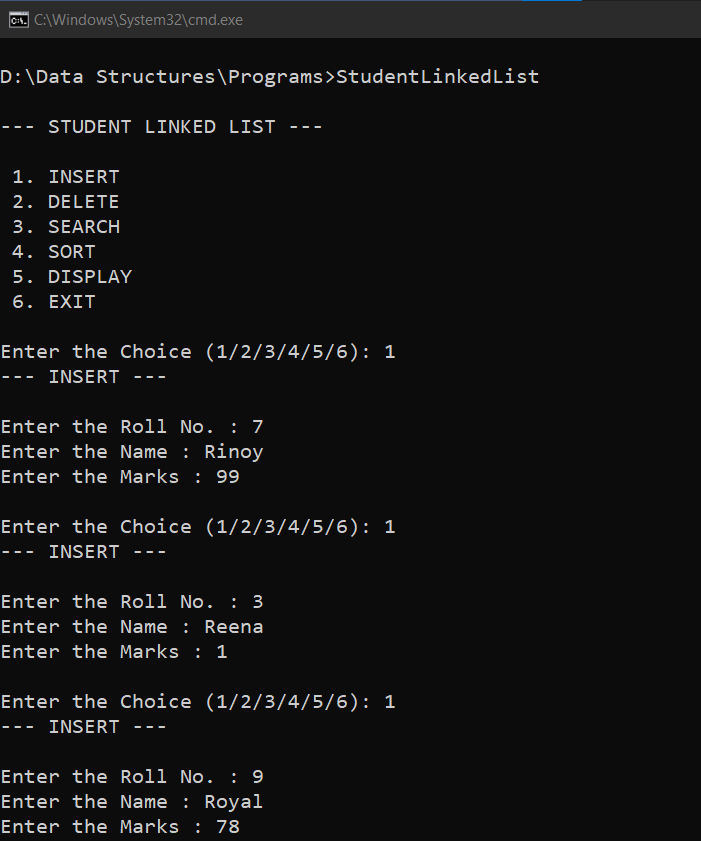
Algorithm SORT()

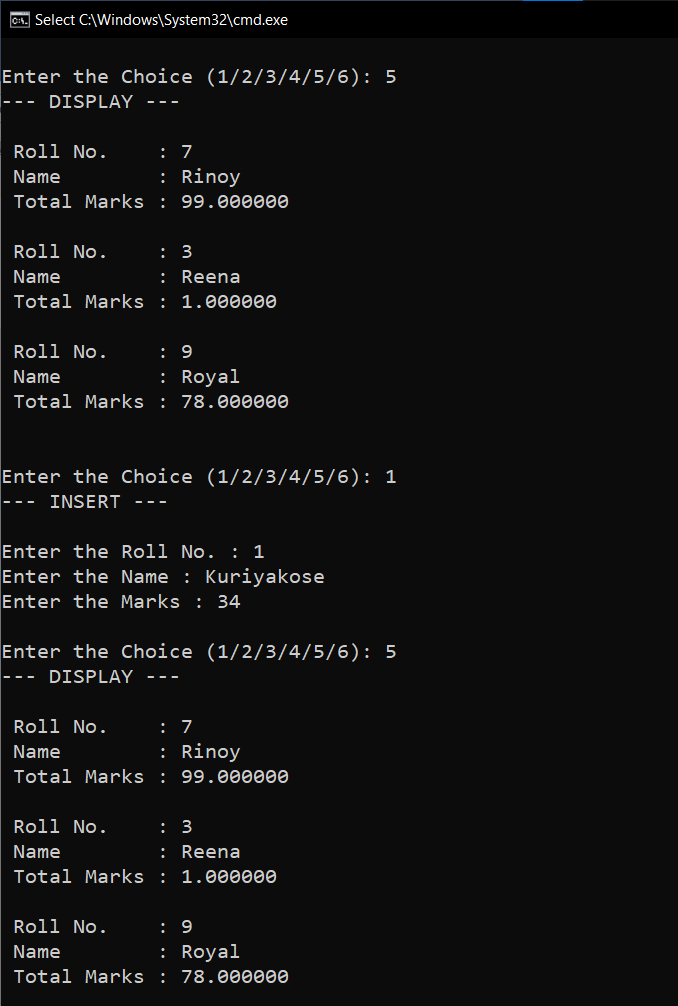
1. ptr1=HEADER->LINK
2. ptr2=HEADER->LINK
3. while(ptr1->LINK!=NULL)
4. ptr2=HEADER->LINK
5. while(ptr1->LINK!=NULL)
6. If(ptr2->rollno> ptr2->LINK->rollno)
7. // Interchange the Values in NODE ptr2 and ptr2->LINK
8. EndIf
9. EndWhile
10. EndWhile

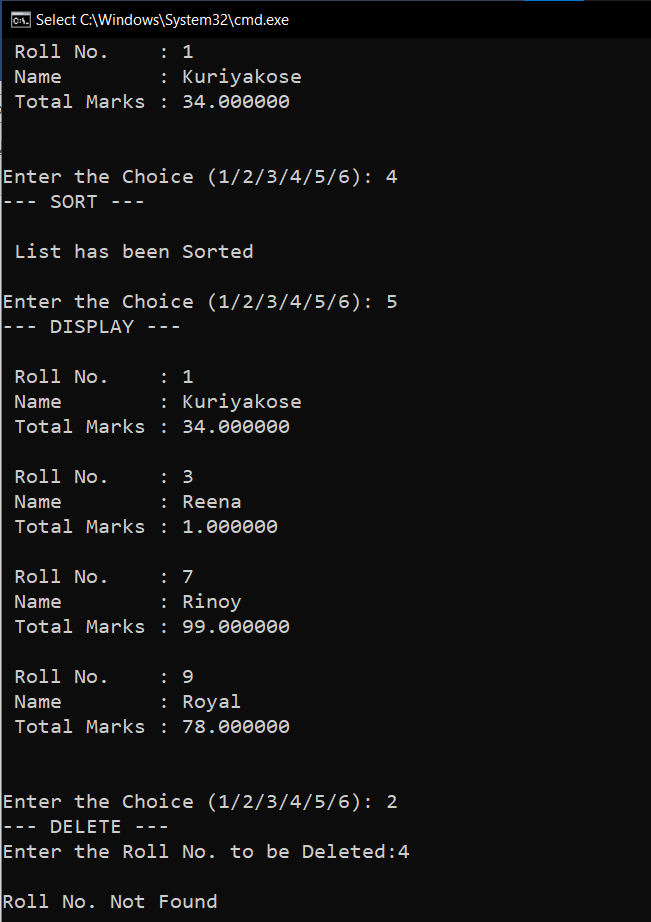
**PROGRAM:**

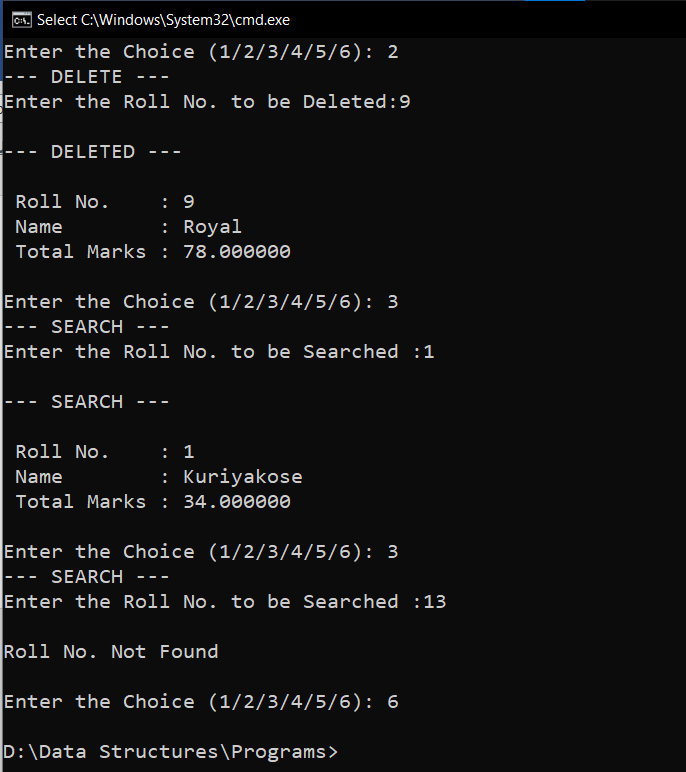
#include<stdio.h>  
#include<stdlib.h>  
#include<string.h>  
struct node{  
 int rollno;  
 char name[50];  
 float marks;  
 struct node \*link;  
};  
  
void insert(struct node\* header){  
 struct node\* new = (struct node\*)malloc(sizeof(struct node));  
 printf("\nEnter the Roll No. : ");  
 scanf("%d",&new->rollno);  
 printf("Enter the Name : ");  
 scanf("%s",new->name);  
 printf("Enter the Marks : ");  
 scanf("%f",&new->marks);  
 new->link=NULL;  
 if(new==NULL){  
 printf("\nMemory Underflow\n");  
 }else{  
 if(header->link==NULL){  
 header->link=new;  
 }else{  
 struct node\* ptr=header;  
 while(ptr->link!=NULL){  
 ptr=ptr->link;  
 }  
 ptr->link=new;  
 }  
 }  
}  
  
void delete(struct node\* header,int key){  
 struct node\* ptr1=header;  
 struct node\* ptr=ptr1->link;  
 if(ptr==NULL){  
 printf("\n List is Empty\n");  
 }else{  
 while(ptr->rollno!=key && ptr->link!=NULL){  
 ptr1=ptr;  
 ptr=ptr->link;  
 }  
 if(ptr->rollno==key){  
 ptr1->link=ptr->link;  
 printf("\n--- DELETED ---\n\n");  
 printf(" Roll No. : %d\n",ptr->rollno);  
 printf(" Name : %s\n",ptr->name);  
 printf(" Total Marks : %f\n",ptr->marks);  
 free(ptr);  
 }else{  
 printf("\nRoll No. Not Found\n");  
 }  
 }  
}  
  
void search(struct node\* header,int key){  
 struct node\* ptr=header->link;  
 if(ptr==NULL){  
 printf("\n List is Empty\n");  
 }else{  
 while(ptr->rollno!=key && ptr->link!=NULL){  
 ptr=ptr->link;  
 }  
 if(ptr->rollno==key){  
 printf("\n--- SEARCH ---\n\n");  
 printf(" Roll No. : %d\n",ptr->rollno);  
 printf(" Name : %s\n",ptr->name);  
 printf(" Total Marks : %f\n",ptr->marks);  
 }else{  
 printf("\nRoll No. Not Found\n");  
 }  
 }  
}  
void sort(struct node\* header){  
 struct node\* ptr=header->link;  
 struct node\* ptr1=header->link;  
 int rollno;  
 char name[50];  
 float marks;  
 if(ptr==NULL){  
 printf("\n List is Empty\n");  
 }else if(ptr->link==NULL){  
 printf("\n List has only One Element\n");  
 }else{  
 while(ptr1->link!=NULL){  
 ptr=header->link;  
 while(ptr->link!=NULL){  
 if(ptr->rollno > ptr->link->rollno){  
 rollno=ptr->rollno;  
 strcpy(name,ptr->name);  
 marks=ptr->marks;  
 ptr->rollno=ptr->link->rollno;  
 strcpy(ptr->name,ptr->link->name);  
 ptr->marks=ptr->link->marks;  
 ptr->link->rollno=rollno;  
 strcpy(ptr->link->name,name);  
 ptr->link->marks=marks;  
 }  
 ptr=ptr->link;  
 }  
 ptr1=ptr1->link;  
 }  
 printf("\n List has been Sorted\n");  
 }  
}  
void display(struct node\* header){  
 printf("\n");  
 struct node\* ptr=header;  
 while(ptr->link!=NULL){  
 ptr=ptr->link;  
 printf(" Roll No. : %d\n",ptr->rollno);  
 printf(" Name : %s\n",ptr->name);  
 printf(" Total Marks : %f\n\n",ptr->marks);  
 }  
}  
  
void main(){  
 int n,x,y,key;  
 char ans='y';  
 struct node\* header = (struct node\*)malloc(sizeof(struct node));  
 header->link=NULL;  
 printf("\n--- STUDENT LINKED LIST --- \n\n");  
 printf(" 1. INSERT \n");  
 printf(" 2. DELETE \n");  
 printf(" 3. SEARCH \n");  
 printf(" 4. SORT \n");  
 printf(" 5. DISPLAY\n");  
 printf(" 6. EXIT\n");  
 while(ans=='y'){  
 printf("\nEnter the Choice (1/2/3/4/5/6): ");  
 scanf("%d",&n);  
 switch(n){  
 case 1:printf("--- INSERT ---\n");  
 insert(header);  
 break;  
 case 2:printf("--- DELETE ---\n");  
 printf("Enter the Roll No. to be Deleted:");  
 scanf("%d", &key);  
 delete(header,key);  
 break;  
 case 3:printf("--- SEARCH ---\n");  
 printf("Enter the Roll No. to be Searched :");  
 scanf("%d", &key);  
 search(header,key);  
 break;  
 case 4:printf("--- SORT ---\n");  
 sort(header);  
 break;  
 case 5:printf("--- DISPLAY ---\n");  
 display(header);  
 break;  
 case 6:ans='n';  
 break;  
 default:printf("Enter a Valid Input\n");  
 }  
 }  
}

**OUTPUT:**









**RESULT:**

The given operations are performed on a Student linked list.