Student Details linked list

VAISHAK S(R3-258)

**Algorithm**

Input: Student details and provides operations like search,delete and sorting in student lists

Output: Sorted student list

Steps:

Define a structure student

struct student

{

int rno;

char name[20];

int mark;

struct student \*next;

}\*head;

**insert( )**

1. Start
2. Declare structure pointers p and temp
3. Print "Enter the Roll No., Name and Mark of the student"
4. scanf("%d",&temp->rno)
5. scanf("%s",temp->name)
6. scanf("%d",&temp->mark)
7. create a node temp
8. if(head==NULL)
9. head=temp
10. p=temp
11. else
12. p->next=temp
13. p=temp
14. Stop

**sort( )**

1. Start
2. Declare variables temprno & tempmark and also char variable tempname[20]
3. Declare structure pointers p and q
4. p=head and q=head->next
5. if(head==NULL)
6. print "The list is empty"
7. return
8. while(p!=NULL)
9. if(p->rno > q->rno)
10. Tempname= p->name
11. temprno=p->rno
12. tempmark=p->mark
13. p->name =q->name
14. p->rno=q->rno
15. p->mark=q->mark
16. q->name=tempname
17. q->rno=temprno
18. q->mark=tempmark
19. q=q->next
20. p=p->next
21. Stop

**delete( )**

1. Start
2. Declare structure pointers p and q
3. p=head,q=NULL
4. if(head==NULL)
5. print "The list is empty"
6. return
7. Print “Enter the Roll number of student to be deleted: "
8. Read num
9. if(head->rno==num)
10. head = head -> next;
11. Delete node p
12. return
13. while(p->next!=NULL)
14. if(num==p->next->rno)
15. q=p->next
16. p->next=q->next
17. Delete node q
18. display()
19. return
20. p=p->next
21. Print "Student not present in the list"
22. Stop

**search( )**

1. Start
2. Declare structure pointer ptr
3. if(head==NULL)
4. Print “The list is empty
5. Return
6. Print "Enter the Roll No. of the student to be found: “
7. Read roll\_no
8. while(ptr!=NULL)
9. if(roll\_no==ptr->rno)
10. Print ptr->rno,ptr->name,ptr->mark
11. ptr=ptr->next
12. Stop

**display( )**

1. Start
2. Declare structure pointer disp
3. disp=head
4. if(head==NULL)
5. print "The list is empty"
6. Return
7. while(disp!=NULL)
8. Print disp->rno, disp->name, disp->mark
9. disp=disp->next;
10. Stop

**main( )**

1. Start
2. Declare variables ch and ans
3. Print "1. Insert Student Details"
4. Print "2. Delete Student Details"
5. Print "3. Search Student Details"
6. Print "4. Display Student List"
7. Print "Enter your choice: "
8. Read choice ch
9. If(ch==1)
10. Call function insert( )
11. Call function sort( )
12. Call function display( )
13. Else if(ch==2)
14. Call function delete( )
15. Call function display( )
16. Else if(ch==3)
17. Call function search( )
18. Else if(ch==4)
19. Call function display( )
20. Else
21. Print “Invalid choice”
22. Print “Do you want to continue? (y/n)”
23. Read ans
24. If(ans==’y’ or ans==’Y’)
25. Goto step 3
26. Stop

**Program**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct student

{

int rno;

char name[20];

int mark;

struct student \*next;

}\*head;

void insert()

{

struct student \*p,\*temp;

printf("Enter the Roll No., Name and mark of the student\n");

scanf("%d",&temp->rno);

scanf("%s",temp->name);

scanf("%d",&temp->mark);

temp=(struct student\*)malloc(sizeof(struct student));

if(head==NULL)

{

head=temp;

p=temp;

}

else

{

p->next=temp;

p=temp;

}

}

void sort()

{

int temprno,tempmark;

char tempname[20];

struct student \*p=head,\*q=head->next;

if(head==NULL)

{

printf("\nThe list is empty\n");

return;

}

while(p!=NULL)

{

while(q!=NULL)

{

if(p->rno > q->rno)

{

strcpy(tempname,p->name);

temprno=p->rno;

tempmark=p->mark;

strcpy(p->name,q->name);

p->rno=q->rno;

p->mark=q->mark;

strcpy(q->name,tempname);

q->rno=temprno;

q->mark=tempmark;

}

q=q->next;

}

p=p->next;

}

}

void delete()

{

int num;

struct student \*p=head,\*q=NULL;

if(head==NULL)

{

printf("\nThe list is empty");

return;

}

printf("\nEnter the Roll number of student to be deleted: ");

scanf("%d",&num);

if(head->rno==num)

{

head = head -> next;

free(p);

return;

}

while(p->next!=NULL)

{

if(num==p->next->rno)

{

q=p->next;

p->next=q->next;

free(q);

display();

return;

}

p=p->next;

}

printf("\nStudent not present in the list");

}

void search()

{

struct student \*ptr;

int roll\_no;

if(head==NULL)

{

printf("\nThe list is empty");

return;

}

printf("Enter the Roll No. of the student to be found: ");

scanf("%d",&roll\_no);

while(ptr!=NULL)

{

if(roll\_no==ptr->rno)

printf("%d\t%s\t%d\n",ptr->rno,ptr->name,ptr->mark);

ptr=ptr->next;

}

}

void display()

{

struct student \*disp=head;

if(head==NULL)

{

printf("\nThe list is empty");

return;

}

while(disp!=NULL)

{

printf("%d\t%s\t%d\n",disp->rno,disp->name,disp->mark);

disp=disp->next;

}

}

void main()

{

char ans;

int ch;

do

{

printf("1. Insert Student Details");

printf("\n2. Delete Student Details");

printf("\n3. Search Student Details");

printf("\n4. Display Student List");

printf("\nEnter your choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1:

insert();

sort();

display();

break;

case 2:

delete();

display();

break;

case 3:

search();

break;

case 4:

display();

exit(0);

default:

printf("\nInvalid choice\n");

}

printf("\nWant to continue? (y/n)");

scanf("%s",&ans);

}while(ans=='y'||ans=='Y');

}