Linked List Sheet

1.insert new node:

void insert\_head (node\* head\_ptr, int entry){

node \*new\_node;

new\_node→data=entry;

new\_node→next=head\_ptr;

head\_ptr=new\_node;

}

2. remove node:

void remove\_node(node\* p){

node \*remove\_ptr;

remove\_ptr=p;

p=p→next;

delete remove\_ptr;

}

3. print data:

void display(node\* head\_ptr){

node \*cursor;

for(cursor=head\_ptr;cursor!=NULL; cursor=cursor→next;){

cout<<cursor→data<<endl;

}

}

4. locate to next:

void locate\_next(node\* next\_ptr){

node \*locate\_ptr;

if(next\_ptr==NULL)

locate\_ptr=NULL;

else{

locate\_ptr→next=next\_ptr;

next\_ptr=locate\_ptr;

}}

5. copy linked list:

Node\* copy(Node\* &list,Node\* &x,Node\* &y,Node\* &z){

if(list==NULL)

cout<<"empty";

x=p;

y=q;

z=r;

Node \*res;

res->data=list->data;

res->link=copy(list->link);

return res;

}

6. sum function:

int sum(const node\* head\_ptr){

int sum;

node\* cursor;

cursor=head\_ptr;

if(cursor=NULL) return 0;

else{

while(cursor!=NULL)

sum=sum+cursor→data;

cursor=cursor→link;

}

cout<<sum;

}

7. product function:

int product(const node\* head\_ptr){

int product;

node\* cursor;

cursor=head\_ptr;

if(cursor=NULL) return 1;

else{

while(cursor!=NULL)

product=producr\*cursor→data;

cursor=cursor→link;

}

cout<<product;

}