```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>

typedef struct dll
{
   struct dll *prev;
   int info;
   struct dll *next;
}node;

node *head,*tail;

//Empty the list
void empty_list()
{
  head=tail=NULL;
}

//insert the value in the list
void insert()
{
    node *ptr;
    ptr=(node*)malloc(sizeof(node));
      cout<<"Enter the value: ";
      cin>>ptr->info;

    if(head==NULL)
    {
       head=tail=ptr;
       ptr->prev=ptr->next=NULL;
    }
    else
    {
       ptr->prev=tail;
       tail->next=ptr;
       tail=ptr;
       ptr->next=NULL;
    }
}

//traverse/display funtion
void traverse()
{
    node *ptr=head;

    if(head==NULL)
    {
      cout<<"List Is Empty";
    }
    else
    {
      cout<<"List Values : "<<endl;
      while(ptr!=NULL)
      {
              cout<<ptr->info<<endl;
              ptr=ptr->next;
      }
    }
}

void ins_beg(int item)
{
    node *ptr;
    ptr=(node*)malloc(sizeof(node));

    ptr->info=item;

       if(head==NULL)
       {
              head=tail=ptr;
              ptr->prev=ptr->next=NULL;
       }
       else
       {
              ptr->next=head;
              ptr->prev=NULL;
              head->prev=ptr;
              head=ptr;
       }
}

node* del(node *ptr)
{
  node *loc;

  if(ptr->next==NULL && head!=tail) //check is it last node
  {
      loc=ptr->prev;
      tail=loc;
      loc->next=NULL;
  }
  else if(head==tail) //check is only one node
  {
     head=tail=NULL;
  }
  else if(ptr->prev==NULL) //check is it First node
  {
      loc=ptr->next;
      head=loc;
      loc->prev=NULL;
  }
  else                 //rest of location
  {
    loc=ptr->prev;
    loc->next=ptr->next;
    loc=ptr->next;
    loc->prev=ptr->prev;
  }
   loc=ptr->next;
   free(ptr);
   return loc;
}
void chekup()
{
   node *ptr=head,*loc;
   int sum=0;

   if(head!=NULL)
   {

       while(ptr!=NULL)
       {
              if(ptr->info%5==0)
              {
                  sum=sum+ptr->info;
                  loc=ptr;
                  ptr=ptr->next;
                  ptr=del(loc);
              }
              else
              {
                ptr=ptr->next;
              }
       }
       ins_beg(sum);
   }
   else
   {
       cout<<"List Is empty";
   }
}

void main()
{
   int i,n;
   clrscr();
    empty_list();

      cout<<"Enter the Size of list: ";
      cin>>n;

      for(i=1;i<=n;i++)
      {
                insert();
      }

    cout<<endl<<endl<<"Display after inserttion"<<endl;
    traverse();

      chekup();
    cout<<"Display after checkup: "<<endl;
    traverse();
 getch();
}
```