Dsa-Lab-Task : 6

Roll No: 24P-0706 **Dept: BS-CS**

Name: Aazan Noor Khuwaja **Section : 3D**

Qns1:

```
#include<iostream>
#include<string>
using namespace std;
class node{
   public:
   int id,price,quantity;
   string name;
   node *next,*previous;
   node(int i,string n,int p,int q)
   {
      id=i;
      name=n;
      price=p;
      quantity=q;
      next=previous=NULL;
   }

};
class dL{
   public:
```

```
node *head,*tail;
dL(){
   head=tail=NULL;
}
void sort()
{
   int size=0;
   node *tp=head;
   do
   {
      size++;
      tp=tp->next;
   }while(tp!=head);

   for(int i=0;i<size-1;i++){
      tp=head;
      for(int j=0;j<size-i-1;j++){
         if(tp->id>tp->next->id){
            int swap=tp->id;
            tp->id=tp->next->id;
            tp->next->id=swap;
         }
          tp=tp->next;
      }

   }
}
```

```
void add_product()
{
    int i,p,q;
    string n;
    cout <<"enter unique id:"<<endl;
    cin>>i;
    cin.ignore();
    cout<<"enter name :"<<endl;
    getline(cin,n);

    cout <<"Enter Price:"<<endl;
    cin>>p;

    cout<<"Quantity:"<<endl;
    cin>>q;

    node *n_node=new node(i,n,p,q);
    if(head==NULL){
        head=tail=n_node;
        head->next=head;
        head->previous=head;
        return;
    }
    tail->next=n_node;
    n_node->previous=tail;
    tail=n_node;
    tail->next=head;
```

```
    head->previous=tail;
    sort();
}



void remove_product()
{
    int i;
    cout <<"\nEnter the id of product which you want to delete:"<<endl;
    cin>>i;

    if(head==NULL)
    {
        cout << "\nInventory is already empty" <<endl;
        return;
    }
    else if(head->id==i)
    {
        head=head->next;
        return;
    }
    node *h=head;
    do
    {
        if(h->id==i)
        {
            node *temp=h;
```

```
            temp->previous->next=temp->next;
            temp->next->previous=temp->previous;
            delete  temp;
            return;
          }
          h=h->next;
       }while(h!=head);
       cout <<"Sorry! Your this id:"<<i<<" Does not exist! "<<endl;
       sort();
    }
    void display_product(){
       node *tp=head;
       do
       {   cout <<"ID: "<<tp->id<<" Name: "<<tp->name<<" Price: "<<tp->price<<" Quantity:
"<<tp->quantity<<endl;
          tp=tp->next;
       }while(tp!=head);



    }
    void update_price(){
       cout <<"Enter the id of the product to update its price: "<<endl;
       int p;
       cin>>p;

        node *h=head;
       do
```

```
    {
      if(h->id==p)
      {
        int pr;
        cout <<"\nEnter the price to update the price of ID: "<<h->id<<" Enter the Price:
"<<endl;
        cin>>pr;

        h->price=pr;
        cout <<"\nThe price of your desired product is updated!"<<endl;
        return;
      }
      h=h->next;
    }while(h!=head);
    cout <<"Sorry Your Product with this id:"<<p<<" Does not exist! "<<endl;
  }
  void find_product(){
     cout <<"Enter the ID of the product:"<<endl;
    int p;
    cin>>p;

     node *h=head;
    do
    {
      if(h->id==p)
      {
        cout <<"Product Found :"<<endl;
```

```
            cout <<"ID: "<<h->id<<" Name: "<<h->name<<" Price: "<<h->price<<" Quantity: "<<h->quantity<<endl;
            return;
          }
          h=h->next;
       }while(h!=head);
       cout <<"Sorry! Your Product with this id:"<<p<<" Does not exist! "<<endl;

   }
};
int main()
{
dL d;
d.add_product();
d.add_product();
d.add_product();
d.display_product();
d.remove_product();
d.display_product();
d.update_price();
d.display_product();
d.find_product();
}
```

Qns 2:

```
#include<iostream>
```

```
using namespace std;
class node
{
   public:
   char data;
   node *next,*previous;
   node(char d){
      data=d;
      next=previous=NULL;
   }
};

class word{
public:
node *head,* tail;
   word()
   {
   head=tail=NULL;
   }
   void insert(char ins)
   {
      node *n_node=new node(ins);
      if(head==NULL)
      {
         head=tail=n_node;
         return ;
      }
      tail->next=n_node;
      n_node->previous=tail;
```

```
    tail=n_node;
}
void cipher_the_msg()
{
    int shift;
    cout<<"\nEnter the position with which you want to shift:"<<endl;
    cin>>shift;
    node *temp=head;
    shift=shift%26;
    while(temp!=NULL)
    {
        //for capital letters
        if(temp->data >= 'A' && temp->data <='Z')
        {
            if(shift>='Z' ){
                shift =('A'-1) + (shift%'Z');
                 temp->data=temp->data+shift;
            }
            else{
                 temp->data=temp->data+shift;
                 if(temp->data>'Z'){
                   temp->data=('A'-1)+(temp->data%'Z');
                 }
            }

        }



        //for small letters
```

```
   if(temp->data >='a' && temp->data<='z'){
      if(shift>'z'){
         shift=('a'-1)+(shift%'z');
          temp->data=temp->data+shift;
      }
      else{
           temp->data=temp->data+shift;
           if(temp->data>'z'){
             temp->data=('a'-1)+(temp->data%'z');
           }
      }
   }
   temp=temp->next;
}

temp=head;
while(temp!=NULL){
   cout<<temp->data;
   temp=temp->next;
}

//now for decode
char c;
cout <<"\nDo you want to decode: y/n "<<endl;
cin >> c;
temp=head;
switch(c){
case 'y':
{
```

```
while(temp!=NULL)
{
   //for capital letters
   if(temp->data >= 'A' && temp->data <='Z')
   {
      if(shift>='Z'){
         shift =('A'-1) + (shift%'Z');
          temp->data=temp->data-shift;
      }
      else{
           temp->data=temp->data-shift;
           if(temp->data<'A'){
            temp->data=temp->data+ 26;
           }
           if(temp->data>'Z'){
            temp->data=('A'-1)+(temp->data%'Z');
           }
      }

   }


   //for small letters
   if(temp->data >='a' && temp->data<='z'){
      if(shift>'z'){
         shift=('a'-1)+(shift%'z');
          temp->data=temp->data-shift;
      }
      else{
```

```
            temp->data=temp->data-shift;
             if(temp->data<'a'){
              temp->data=temp->data+ 26;
             }
            if(temp->data>'z'){
              temp->data=('a'-1)+(temp->data%'z');
            }
        }
      }
      temp=temp->next;

   }
   temp=head;
   while(temp!=NULL){
      cout<<temp->data;
      temp=temp->next;
   }
      break;
}

   case 'n':
   {
      break;
   }
   default:
   {
      cout <<"Not a valid input!"<<endl;
      break;
   }
```

```
    }



    }



    void display()
    {
        node *temp=head;
       while(temp!=NULL){
         cout<<temp->data;
          temp=temp->next;
       }
    }

};

int main()
{
    word w;
    w.insert('A');
     w.insert('a');
     w.insert('z');
      w.insert('Z');
       w.insert('n');
       w.display();
       w.cipher_the_msg();
```

}