//REG # FA14-BSSE-037

//ADS ASSIGNMENT NO:4 (QUEUE (ADT) APPLY ON (SLL) LINK LIST)

#include<iostream>

#include<string>

#include<Windows.h>

using namespace std;

struct item // for one bill store as node data part

{

int b;

item \*next;

};

class list //list of bills of specific delivery boy

{

private:

unsigned int count,s;

int xount,deliver\_times;

item \* head;

item \* temp, \* ptr, \*pre;

public:

list()

{

head=NULL;

count=0;

s=0;

deliver\_times=0;

}

void push(int n) //add bill in que of nodes

{

++count; //count the number of bills store in delivery boy queue

if(head==NULL)

{

temp=new item;

temp->b=n;

temp->next=NULL;

head=temp;

}

else

{

ptr=head;

while(ptr->next!=NULL)

{

ptr=ptr->next;

}

temp=new item;

temp->b=n;

temp->next=NULL;

ptr->next=temp;

}

}

///////////////////////////////////////////////

void show()

{

ptr=head;

while(ptr!=NULL)

{

cout<<"\n"<<ptr->b<<endl;

ptr=ptr->next;

}

}

/////////////////////////////////////////////

void remove()

{

//xount--; //count of numbers

++deliver\_times; //delivered items from specific boy

temp =head; //reference save in temp so that start from head

if(temp==NULL)

{

cout<<"\nno bill enter in queue\n";

return;

}

else

{

pre=temp->next; //2nd node save in pre

delete temp; //ist node del (del from front in Queue)

cout<<endl;

head=pre; //re\_connect mean pre again connect to head;

}

}

////////////////////////////////////////////////////

void add\_first(int v)

{

item \* temp, \* pre;

if(head==NULL)

{

head=new item;

head->b=v;

head->next=NULL;

}

else

{

ptr=head; //start from head

temp=new item;

temp->b=v;

temp->next=ptr; //first connect with already ist node

head=temp; //then new ist node store in head

}

}

////////////////////////////////////////////////////

int status()

{

s=count-deliver\_times;

cout<<"current items in delivering process Queue are : "<<s<<endl;

return s;

}

///////////////////////////////////////////////////////

int delivery\_record()

{

return deliver\_times;

}

};

////////////////////////////////////////////////////////

class bill

{

private:

int price[5];

int top,r,c;

int s;

public:

bill()

{

top=s=0;

}

void save(int v)

{

//int a[5];

price[++top]=v;

}

//////////////////////////////////////

int sum()

{

s=0; //for no repetitiion exact calculating

for(int j=1;j<=top;j++)

{

s+=price[j];

}

return s;

}

};

///////////////////////////////////////////

int main()

{

list s1,s2,s3,s4;

int k;

char c;int op;

string st,ch;

int n,p,q;

char opt;

do

{

cout<<"\nFood supliers company...........\n";

cout<<"\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\t\t\t\t\t Fast Food company \n";

cout<<"\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

bill d1;

do

{

cout<<"\nfood manue!\n";

cout<<"\t\t-------\t\t-----------\t\t----------------\n";

cout<<"\t\tITEM\_ID\t\tDESCRIPTION\t\tUNIT PRICE (Rs.)\n";

cout<<"\t\t-------\t\t-----------\t\t----------------\n";

cout<<"\t\t [01] \t\t Pizza \t\t 200 \n"

"\t\t [02] \t\t Shwarma \t\t 100 \n"

"\t\t [03] \t\t Burger \t\t 150 \n"

"\t\t [04] \t\t Sandwitch\t\t 90 \n";

cout<<"\nEnter choice : ";

cin>>n;

switch(n)

{

case 1:

cout<<"\nyou select pizza";

cout<<"\nEnter quantity : ";

cin>>q;

p=200\*q;

d1.save(p);

break;

case 2:

cout<<"\nyou select shwarma";

cout<<"\nEnter quantity : ";

cin>>q;

p=100\*q;

d1.save(p);

break;

case 3:

cout<<"\nyou select zinger burger";

cout<<"\nEnter quantity : ";

cin>>q;

p=150\*q;

d1.save(p);

break;

case 4:

cout<<"\nyou select sandwitch";

cout<<"\nEnter quantity : ";

cin>>q;

p=90\*q;

d1.save(p);

break;

case 5:

cout<<"\nTotal bill : ";

cout<<d1.sum()<<endl;

break;

}

cout<<"\n\t\t\tDo any more (y/n) : ";

cin>>opt;

}

while(opt=='y');

///////////////////////////////////

cout<<"\n\n\n";

int t;

//-----------------------------------------

do

{

cout<<"1-Apply for order \n2-For delevery sent\n3-show pending deleiveres\n4-status\n"

"5-no of delivered items\n6-add firsts\t\tEnter choice : ";

cin>>op;

switch(op)

{

case 1:

//cout<<"Enter bill to add : ";

cout<<"\nfirst check which delivery boy free : \n";

cout<<"all delivery boys status : \n";

cout<<"Boy 1 : \n";

s1.status();

cout<<"Boy 2 : \n";

s2.status();

cout<<"Boy 3 : \n";

s3.status();

cout<<"Boy 4 : \n";

s4.status();

cout<<"Enter boy id ";

cin>>k;

if(k==1)

{

cout<<"Boy 1 delivered u : ";

t=d1.sum();

s1.push(t);

}

else if(k==2)

{

cout<<"Boy 2 delivered u : ";

t=d1.sum();

s2.push(t);

}

else if(k==3)

{

cout<<"Boy 3 delivered u : ";

t=d1.sum();

s3.push(t);

}

else if(k==4)

{

cout<<"Boy 4 delivered u : ";

t=d1.sum();

s4.push(t);

}

else

cout<<"choice is wrong : \n";

break;

case 2:

//-------------------------

cout<<"enter delevery boy id when u deliver : ";

cin>>k;

if(k==1)

s1.remove();

else if(k==2)

s2.remove();

else if(k==3)

s3.remove();

else if(k==2)

s4.remove();

else

cout<<"choice is wrong : \n";

//------------------------

cout<<"\n ordered delivered sucessfully!\n";

break;

case 3:

cout<<"enter delevery boy id when u deliver : ";

cin>>k;

if(k==1)

s1.show();

else if(k==2)

s2.show();

else if(k==3)

s3.show();

else if(k==2)

s4.show();

else

cout<<"choice is wrong : \n";

cout<<"\npending delivery list is : ";

break;

case 4:

cout<<"enter delevery boy to see status : ";

cin>>k;

if(k==1)

s1.status();

else if(k==2)

s2.status();

else if(k==3)

s3.status();

else if(k==2)

s4.status();

else

cout<<"choice is wrong : \n";

break;

case 5:

cout<<"Enter boy id to check out there deliveries : ";

cin>>k;

if(k==1)

cout<<"Boy 1 delivered bills are : "<<s1.delivery\_record()<<endl;

else if(k==2)

cout<<"Boy 1 delivered bills are : "<<s2.delivery\_record()<<endl;

else if(k==3)

cout<<"Boy 1 delivered bills are : "<<s3.delivery\_record()<<endl;

else if(k==4)

cout<<"Boy 1 delivered bills are : "<<s4.delivery\_record()<<endl;

else

cout<<"\nERROR---boy id not correct so record not found\n";

break;

case 6:

s1.add\_first(2);

break;

}

cout<<"do any more delivery (y/n) : ";

cin>>c;

system("cls");

}

while(c=='y');

cout<<"Do any more action : ";

cin>>ch;

}

while(c=='y');

exit(1);

system("pause");

return 0;

}