Task 1

void insertith(node\*& head, int position, int item)

{

node \* newnode = new node;

node \* temp = head;

node \* tempprev = head;

newnode->item = item;

if (position == 0)

{

head = newnode;

newnode->next = temp;

return;

}

for (int i = 1; i <= position; i++)

{

if (temp->next == nullptr)

{

temp->next = newnode;

newnode->next = nullptr;

return;

}

tempprev = temp;

temp = temp->next;

}

tempprev->next = newnode;

newnode->next = temp;

}

Task 2

#ifndef Stock\_h

#define Stock\_h

using namespace std;

#include <iostream>

class Stock

{

public:

Stock(string name, double price);

private:

double price;

string name;

};

#endif

#include "Stock.h"

#include<iostream>

using namespace std;

Stock::Stock(string name, double price)

{

this->price = price;

this->name = name;

}

#ifndef MyStock\_h

#define MyStock\_h

#include "Stock.h"

class MyStock : public Stock

{

public:

MyStock(int shares, string name, double price);

private:

int shares;

};

#endif

#include "MyStock.h"

#include "Stock.h"

#include<iostream>

using namespace std;

MyStock::MyStock(int shares, string name, double price) : Stock(name, price)

{

this->shares = shares;

}