Pseudo code

FixMe (1): Define a structure to hold the bids

Create the struct container

Create a bid variable

Create an unsigned variable for the key

Create a node pointer for the next value

Create a default constructor

Assign key variable to the max value “UINT\_MAX”

Assign next pointer to null

Create Constructor with a bid argument

Assign the bid variable to the bid argument

Create Constructor with a bid argument and key argument

Assign the key variable to the key argument

Create a vector of nodes

Create a variable for the table size and assign to the constant value of default size

FixMe (2): Initialize the structures used to hold bids

Under the default constructor

Call resize of the vector and pass the table size variable

Create an alternative constructor with a variable for the size

Change the table size variable to the passed size variable

FixMe (3): Implement logic to free storage when class is destroyed

Call the erase method on the nodes vector and pass the starting location as an argument

FixMe (4): Implement logic to calculate a hash value

Assign the key to key modulo table size

FixMe (5): Implement logic to insert a bid

Create an unsigned key variable and assign it the hash method passing the “atoi” method, passing the bid’s bid id as a string.

Create a node pointer old node and assign it to the address of the node of its key

Check if old node is null

Create a new node and pass the bid and the key

Call the nodes insert method and pass the beginning location plus key and the new node pointer.

If not

Check if the old node’s key equals the “UINT\_MAX”

Change old nodes key to key

Change old nodes bid to bid

Change old nodes next to null

If not

Loop as long as old nodes next pointer is not null

Change old node to old node’s next

Change old node’s next to a new node passing the bid and key

FixMe (6): Implement logic to print all bids

Loop through all of the nodes in nodes

Create a new bid to the element in the current location of the loop

Print out the bid id, bid title, bid amount, and the bid fund of each element

FixMe (7): Implement logic to remove a bid

Create an unsigned key variable and assign it the hash method passing the “atoi” method, passing the bid’s bid id as a string.

Call the erase method on the nodes vector passing the beginning location plus the key

FixMe (8): Implement logic to search for and return a bid

Create an unsigned key variable and assign it the hash method passing the “atoi” method, passing the bid’s bid id as a string

Create a node pointer node and assign it to the address of the node of its key

Check if node equals null and node’s key equals “UINT\_MAX”

Return the bid

Check if node is not null, node’s key equals "UINT\_MAX”, and node’s bid id compared to the bid id passed do not match

Return the node’s bid

Loop while the node is not null

Check if node’s key is not equal to “UINT\_MAX” and node’s bid id compared to the bid id passed do not match

Return the node’s bid

Assign node to the nodes next variable