This code is designed to create a hash key value to match a bid value in an excel sheet. The key is used to find entries in the list at a faster rate than other algorithms. It was a little difficult to create, but using the video linked in the materials and previous labs, I was able create the code well. I did have one or two issues, but those were only a mistyped = symbol or an l where an i was supposed to be.

**Fixme 1 – Define structures**

DEFINE Struct Node {

DEFINE Bid bid

DEFINE unsigned key

DEFINE Pointer Node next

DEFINE Default Node constructor {

SET key to uint max

SET next to nullptr

}

DEFINE Node constructor with bid tempBid{

SET bid to tempBid

}

DEFINE Node constructor with bid tempbid and key tempkey {

SET key to tempKey

}

}

DEFINE Vector of Nodes nodes

DEFINE unsigned hashSize

**Fixme 2 – Hash Table – Initialize**

DEFAULT CONSTRUCTOR {

RESIZE nodes vector with hashSize

}

Constructor w/unsigned size {

SET this hashsize to size

RESIZE nodes vector with hashSize

}

**Fixme 3 – Hash Table – Free Storage**

ERASE nodes vector with nodes begin

**Fixme 4 – Hash Table – Key value**

RETURN key modulus hashSize

**Fixme 5 – Insert**

DEFINE unsigned key

SET key to hash bid id with atoi

DEFINE Pointer prevNode

SET prevNode to pointer of key of nodes

IF prevNode is nullptr {

DEFINE Pointer Node newNode

SET newNode to new node with bid and key

INSERT newNode in vector ndoes

}

ELSE {

IF prevNode’s key is equal to uint max {

SET prevNode key to key

SET prevNode bid to bid

SET prevNode next to nullptr

}

ELSE {

WHILE prevNode next is not nullptr{

SET prevNode to prevnode next

}

SET prevNode next to NEW node with bid and key

}

}

**Fixme 6 – Print All**

FOR all nodes in node {

If node is NOT empty

PRINT key at node

CALL displayBid func with bid at node

**Fixme 7 – Remove bid**

DEFINE unsigned key

SET key to hash bid id with atoi

ERASE nodes vector with begin plus key

**Fixme 8 – Search**

DEFINE Bid bid

SET key to hash bid id with atoi

DEFINE Pointer Node justNode

SET justNode to pointer of key of nodes

IF justNode equal nullptr OR justNode key equals uint max {

RETURN bid

}

IF justNode is not nullptr AND justNode key is not uint max AND justNode bid id compare is 0{

RETURN bid of justNode

}

WHILE justNode is not nullptr{

IF key of justNode is not uint max AND key of justNode bid id compare is 0 {

RETURN bid of justNode

}

SET justNode to justNode next

}

RETURN bid