Create structure named ‘showroom’

Defining variables of structure:

String variables named company and name of size 17;

Integer variables named name price, quantity, manucost;

Structure pointer named next to link nodes;

->Space Complexity for this Structure:O(n)

Create display() function of datatype void and pass a structure pointer named ‘head’ as argument

Now print "Company Name Price Quantity”

create structure pointer named temp and assign head to it

while temp not equal to NULL

Now print the following

"Company Name Price Quantity”

Change temp to its next node;

End the function

->Time Complexity for this function:

Best Case:O(1) Worst Case:O(n)

->Space Complexity for this function : O(1)

Create create() function of datatype structure and pass a structure pointer named ‘head’ as argument

Assign memory to head pointer dynamically

Ask the user to enter vehicle company, name, manufacturing cost, vehicle price, no.of vehicles

Assign those given values to head’s structure variables

Assign head’s next pointer as NULL

create structure pointer named temp and assign head to it

Create integer variable named ch;

Now print the following

"Enter 1 to enter new vehicle and 0 to stop vehicle insertion"

Assign scanned value to ch variable

Create integer variable named i and initialize it to 2;

while(ch is greater than 0)

Create structure pointer named newv and allocate memory to it dynamically

Ask the user to enter vehicle company, name, manufacturing cost, vehicle price, no.of vehicles

Assign those given values to newv’s structure variables

Increment variable i by 1

Assign newv’s next pointer as NULL

Assign temp’s next pointer as newv

Assign newv pointer to temp

Now print the following

“ Enter 1 to enter new vehicle and 0 to stop vehicle insertion”

Assign scanned value to ch variable

Exit function by returning head

End the function

->Time Complexity for this function:

Best Case:O(1) Worst Case:O(n)

->Space Complexity for this function: O(n)

Create search() function of datatype integer and pass a structure pointer named ‘head’ as argument

Create structure pointer named newv and allocate memory to it dynamically

Ask the user to enter vehicle company, name

Assign those given values to newv’s structure variables

Create structure pointer named temp and assign head to it

Create integer variable named i and initialize it to 0;

while(temp pointer not equal to NULL)

Increment variable i by 1

Compare strings to proceed

if(Company and name variable of temp and newv structure matches along with quantity variable of pointer being greater than 0)

Now print the following

"Yes,available"

Enter number (which is stored in i) in sell option to sell\

Exit function by returning 1;

else

Assign temp pointer as temp next

if(temp pointer is NULL)

Now print the following

“Not Available”

Exit function by returning 0;

End the function

->Time Complexity for this function:

Best Case:O(1) Worst Case:O(n^2)

->Space Complexity for this function: O(1)

Create sell() function of datatype integer and pass a structure pointer named ‘head’ as argument

Create integer variable named profit and initialize it to 0;

Now print the following

"If the customer know what vehicle he want to buy,Press 1

If customer wants to buy what is available,Press 0”;

Create structure pointer named temp and assign head to it

Create integer variable named f;

Assign scanned value to f variable

if(f is equal to 1)

Create integer variable named k and assign integer returned by search() function;

if (k is equal to 0)

Exit function by returning 0;

Else

call display() function

Create integer variables named j,n;

Enter vehicle serial number, number of vehicles

Assign those given values to j and n variables

for(Create integer variable I and initialize it to 0 and increment by 1 till it is just less than j)

Assign temp pointer as temp next;

if(if quantity variable of temp pointer greater than or equal to n)

update temp->quantity as temp->quantity-n;

Now print The vechiles left using temp->quantity;

Assign value to profit variable as (n) times (that particular vehicle’s price-manucost);

Now print earning from that item using profit variable;

else if(if quantity variable of temp pointer less n)

Now print the following using quantity variable of temp pointer

"Those many vehicles not available there are only “temp->quantity” left\n"

Exit function by returning profit;

End the function

->Time Complexity for this function:

Best Case:O(1) Worst Case:O(n)

->Space Complexity for this function: O(1)

Create add() function of datatype void and pass a structure pointer named ‘head’ as argument

Create structure pointer named newv and allocate memory to it dynamically

Ask the user to enter vehicle company, name, no.of vehicles

Assign those given values to newv’s structure variables

Create structure pointer named temp and assign head to it

Create structure pointer named prev and point to NULL

while(temp pointer not equal to NULL)

if(Company and name variable of temp and newv structure matches)

update quantity variable of temp pointer to = quantity of temp + quantity of newv;

break the loop and exit loop;

update prev pointer as temp;

update temp as temp next;

if(temp pointer is equal to NULL)

Ask the user to enter manufacturing cost, vehicle price

Assign those given values to newv’s structure variables

update newv next pointer as NULL;

update prev next as newv;

End the function

->Time Complexity for this function:

Best Case:O(1) Worst Case:O(n^2)

->Space Complexity for this function: O(1)

Create main() function of datatype integer [ Time complexity=O(1)]

Create integer variables named bnet,cnet and initialize both of them to 0;

Create structure pointer named chead, bhead;

Now print the following

"Enter all bike details available in the morning“

Assign pointer returned from create() function to bhead;

Now print the following

"Enter all car details available in the morning“

Assign pointer returned from create() function to bhead;

Now print the following

Enter:

1 to display

2 to sell

3 to add

4 to calculate profit

5 to exit;

while(1)

Ask user to make choice

Create a integer variable named a;

Scan and assign it to a

switch(a)

case 1:

Ask user to enter b or c

Create a character variable named x;

Scan and assign it to x

if(x is equal to b)

Call display() function passing bhead as argument

else if(x is equal to c)

Call display() function passing chead as argument

End case by breaking loop;

case 2:

Ask user to enter b or c

Create a character variable named x;

Scan and assign it to x

if(x is equal to b)

update bnet variable as bnet+ integer returned by sell() function when bhead is passed as argument;

else if x is equal to c)

update bnet variable as cnet+ integer returned by sell() function when chead is passed as argument;

End case by breaking loop;

case 3:

Ask user to enter b or c

Create a character variable named x;

Scan and assign it to x

if(x is equal to b)

call add() function by passing bhead as argument

else if(x is equal to c)

call add() function by passing chead as argument

End case by breaking loop;

case 4:

Print the following

Total profit for the day using bnet variable + cnet variable

End case by breaking loop;

case 5:

if user presses 5,we exit from switch case

End main function by returning;

->Time Complexity for this function:

Best Case:O(1) Worst Case:O(n^)

->Space Complexity for this function: O(1)

The overall space complexity for the code if O(n).

-X-X-X-