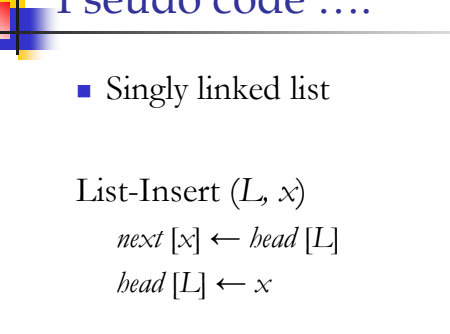
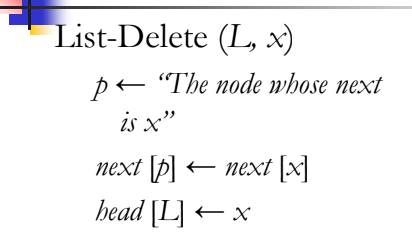
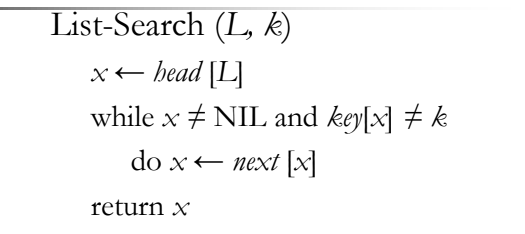
1a.







1b.ans

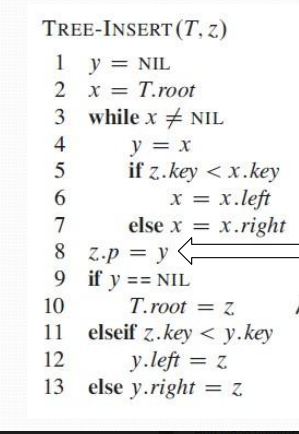
20

11

22

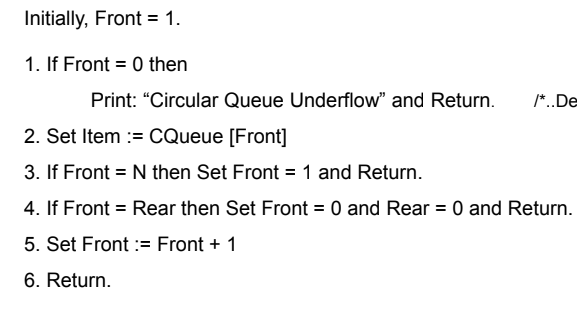
41

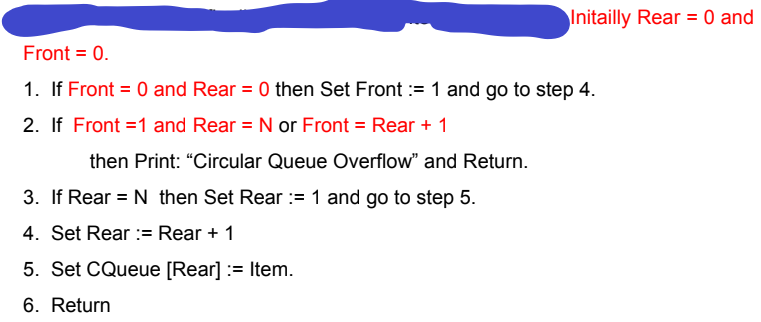
2a.ans



2b.

3a.





4a.

void Merge(int arr[],int lo,int mid,int high)

{

int i = lo , j = mid+1 , temp[100] , k = 1 ;

while(i<=mid&&j<=high)

{

if(arr[i]<arr[j])

{

temp[k] = arr[i] ;

k=k+1;

i=i+1;

}

else

{

temp[k] = arr[j] ;

k=k+1;

j=j+1;

}

}

while(i<=mid)

{

temp[k] = arr[i] ;

k=k+1;

i=i+1;

}

while(j<=high)

{

temp[k] = arr[j] ;

k=k+1;

j=j+1;

}

k = 1 ;

for(i=lo;i<=high;i=i+1)

{

arr[i] = temp [k] ;

k=k+1;

}

}

5a.

repeater(int x)

{

if(x<=1000)

{

printf("\nHello World");

x++;

repeater(x);

}

}

5b.

#include<iostream>

using namespace std;

int main() {

int n=5, sum=0, i;

for(i=1;i<=n;i++)

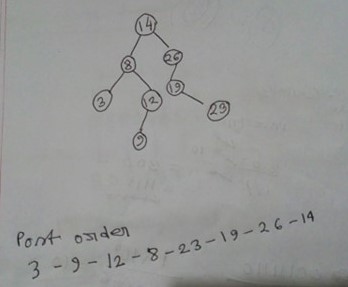
sum=sum+i;

cout<<"Sum of first "<<n<<" natural numbers is "<<sum;

return 0;

}

6a.



6b.

7a. Divide and Conquer method is a class of algorithm which can be applied to problem which can be solved easily if they are broken into sub-problems recursively until we get a sub problem which is trivial to solve.

7b.

* Recursive solution is always logical and it is very difficult to trace. In recursive we must have an if statement somewhere to force the function to return without the recursive call being executed, otherwise the function will never return. Recursion uses more processor time.