Instruction:
The datafile and sourcecode shuold be under the same file
compile in cmd by the following command line:
“
g++ aa.cpp -o evenSumRange
evenSumRange dataX rangeX
”

Modification to BST insert:
Add a field “evensum” to indicate even or odd of Sum of all the node by always looking at root.evensum
Every time when inserting a new node, root.evensum is updating by compare it to the evensum of new node.
After insertion we only need to look at root.evensum

PSEUDOCODE
Function insertion(x, t)
// x is the value we want to insert, t is current node. Implement insertion by recursion, so insertion always goes from root to bottom. when insert a new node, always add newnode’s evensum(newevensum) to current node, then begin recursion move to the next layer

```
if(t == NULL):
    t.key = x
    t.left = t.right = NULL
    if x %2 == 0: t.evensum = 0
    else: t.evensum =1
else if(x<=t.key):
    if x %2 == 0: newevensum = 0
    else: newevensum =1
    if newevensum != t.evensum:t.evensum = 1
    else t.evensum = 0
    t.left = insert(x,t.left) // recursive call
else if(x>t.key):
    if x %2 == 0: newevensum = 0
    else: newevensum =1
    if newevensum != t.evensum:t.evensum = 1
    else t.evensum = 0
    t.left = insert(x,t.right) // recursive call
```

Function report(T) // T is the tree after insertion, get evensum by get evensum of root
x = T.root
Return x.evensum