Pastebin link: https://paste.ubuntu.com/25509160/

*/\* Mo's Algorithm*  
 *\**  
 *\* Approx. Complexity: O(n\*sqrt (n)+ q\*sqrt(n))*  
 *\* Actual Complexity: O(q\*s + n\*n/s)*  
 *\* where s is the bucketsize*  
 *\**  
 *\* Assumptions: 0-based indexing all throughout*  
*\*/*  
  
**int** n, q; *//Array size and number of queries*  
  
**struct** queries{  
 **int** l, r, ind;  
 **int** sol;  
}qa[QUERIES];  
**int** bucket\_size;  
**int** bucket\_no[SIZE];  
**bool** cmp(queries a, queries b){  
 **if**(bucket\_no[a.l]==bucket\_no[b.l]){  
 **if**(a.r==b.r) **return** a.l<b.l;  
 **if**(!bucket\_no[a.l]%2)**return** a.r<b.r;  
 **return** a.r>b.r;  
 }  
 **return** bucket\_no[a.l]<bucket\_no[b.l];  
}  
  
**bool** cmp2(queries a, queries b){  
 **return** a.ind<b.ind;  
}  
  
**inline** **void** add(**int** i){  
 *//Add the ith element onto the subarray*  
}  
  
**inline** **void** remv(**int** i){  
 *//Remove the ith element from the subarray*  
}  
  
**inline** **void** moveL(**int** curr, **int** nxt){  
 **while**(curr>nxt){  
 curr--;  
 add(curr);  
 }  
 **while**(curr<nxt){  
 remv(curr);  
 curr++;  
 }  
}  
  
**inline** **void** moveR(**int** curr, **int** nxt){  
 **while**(curr<nxt){  
 curr++;  
 add(curr);  
 }  
 **while**(curr>nxt){  
 remv(curr);  
 curr--;  
 }  
}  
  
**inline** **void** init(){  
 *//Perform any initializations*  
 *//specific to the problem*  
}  
  
**inline** **int** solve(){  
 *//Return the solution*  
}  
  
**inline** **void** brute(**int** l, **int** r){  
 **for**(**int** i=l; i<=r; i++){  
 add(i);  
 }  
}  
  
**inline** **void** mo(){  
 bucket\_size=sqrt(n);  
 **int** bno=0, bcnt=0;  
 **for**(**int** i=0; i<n; i++){  
 **if**(bcnt==bucket\_size){  
 bno++;  
 bcnt=0;  
 }  
 bucket\_no[i]=bno;  
 bcnt++;  
 }  
 sort(qa, qa+q, cmp);  
 init();  
 brute(qa[0].l, qa[0].r);  
 qa[0].sol=solve();  
 **for**(**int** i=1; i<q; i++){  
 moveL(qa[i-1].l, qa[i].l);  
 moveR(qa[i-1].r, qa[i].r);  
 qa[i].sol=solve();  
 }  
 sort(qa, qa+q, cmp2);  
}