1.Why beg+(end-beg)/2 is preferred method for calculating middle of an array over (beg+end)/2?

**Ans.**(beg+end)/2 there will be problem with this approach,when both Start and end has INT\_MAX it will cause overflow if end<0 or beg<0 but with beg+(end-beg)/2 it will give correct middle of an array.

//program for calculating middle of the array

#include<stdio.h>

#include<limits.h>

int main()

{

int start=INT\_MAX,end=INT\_MAX;

Printf(“start=%d”,start);

Printf(“end=%d”end);

//method1

int mid1=beg+(end-beg)/2

Printf(“mid using beg+(end-beg)/2=%d”,mid1);

//method2

int mid2=(beg+end)/2

Printf(“mid using (beg+end)/2=%d”,mid2);

Return 0;

}

2.//program for ternary search

int ternary\_search(int l,int r, int x)

{

if(r>=l)

{

int mid1 = l + (r-l)/3;

int mid2 = r - (r-l)/3;

if(ar[mid1] == x)

return mid1;

if(ar[mid2] == x)

return mid2;

if(x<ar[mid1])

return ternary\_search(l,mid1-1,x);

else if(x>ar[mid2])

return ternary\_search(mid2+1,r,x);

else

return ternary\_search(mid1+1,mid2-1,x);

}

return -1;

}