**Ques1**: Why we prefer **beg+(end-beg)/2**  over **(beg+end)/2**  in binary search to find the mid value ?

Ans: beg: beginning index of array, end: ending index of array

CASE1: mid= **beg+(end-beg)/2**  CASE2: mid= (**beg+end)/2**  If we take the first case then we can also use pointers, because pointer subtraction is allowed whereas pointer addition is not defined and gives type error Thus we prefer CASE1 over CASE2.

Secondly if suppose we take **beg=INT\_MAX, end= INT\_MAX** ; where INT\_MAX stands for maximum integer value .In CASE1: we get correct results provided that **end-beg does not overflow.**

But **CASE2** results into ***overflowing and gives segmentation fault*.** So we prefer **beg+(end-beg)/2**