QUESTION 1 – In the Binary Search algorithm, it is suggested to calculate the mid as beg + (end – beg) / 2 instead of (beg + end) / 2. Why is it so?

Ans –

In Binary Search Algorithm it is suggested to use   
 beg +(end – beg) / 2 instead of (beg + end ) / 2 because   
 there is no guarantee that beg + end is representable   
 but in the second case, the intermediate values, as well   
 as the expected result, are no more than end , so there   
 is no danger of overflow.

Question 2 – Write the algorithm/function for   
 Ternary Search.

Ans –   
 #include <stdio.h>

int ternarySearch(int b, int e, int num, int ar[]);

void main()

{

int s,i,b, e, p, num;

int ar[5];

printf("\nEnter Size Of The Array - ");

scanf("%d",&s);

printf("\nEnter Elements - ");

for(i=0;i<s;i++)

{

scanf("%d",&ar[i]);

}

b = 0,e=s-1;

printf("\nEnter Element To Search - ");

scanf("%d",&num);

p = ternarySearch(b, e, num, ar);

printf("\nIndex of %d is %d\n", num, p);

}

int ternarySearch(int b, int e, int num, int ar[])

{

while (e >= b)

{

int mid1 = b + (e - b) / 3;

int mid2 = e - (e - b) / 3;

if (ar[mid1] == num)

{

return mid1;

}

if (ar[mid2] == num)

{

return mid2;

}

if (num < ar[mid1])

{

e = mid1 - 1;

}

else if (num > ar[mid2])

{

b = mid2 + 1;

}

else

{

b = mid1 + 1;

e = mid2 - 1;

}

}

return -1;

}

Output –

