Find the missing number in Arithmetic Progression

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>

int findMissingElement(int *arr, int start, int end, int difference)
{
        if(end <= start)
                return INT_MAX;

        int middle = start + (end - start) / 2;

        if (arr[middle+1] - arr[middle] != difference)
                return arr[middle] + difference;


   if (middle > 0 && arr[middle] - arr[middle - 1] != difference)
      return (arr[middle - 1] + difference);

   if (arr[middle] == arr[0] + middle*difference)
      return findMissingElement(arr, middle + 1, end, difference);

   return findMissingElement(arr, start, middle - 1, difference);
}

int main()
{
        int *arr, size, difference;
        printf("Enter size of an array\n");
        scanf("%d", &size);
        //allocate memory for array
        arr = (int *)malloc(size * sizeof(int));

        printf("Enter Array elements ");
        for(int index = 0; index < size; index++)
                scanf("%d", &arr[index]);

        //find difference of arithemetic progression
        difference = (arr[size - 1] - arr[0]) / size;

        printf("The missing element in AP is = %d\n",
                findMissingElement(arr, 0, size-1, difference));
        return 0;
}
```

Time Complexity: O(logn)
Space Complexity: O(logn)