```c
// Inversions

#include<stdio.h>
int merge(int arr[],int temp[],int left,int mid,int right){
    int i, j, k,count=0;

    i = left;
    j = mid;
    k = left;
    while ((i<=mid-1)&&(j<=right)) {
        if (arr[i] <= arr[j]) {
            temp[k++] = arr[i++];
        }
        else {
            temp[k++] = arr[j++];
            count =count + (mid - i);
        }
    }
    while (i <= mid - 1)
        temp[k++] = arr[i++];
    while (j <= right)
        temp[k++] = arr[j++];

    for(int i=left;i<=right;++i)
          arr[i]=temp[i];
  return count;
}
int merge_count(int arr[],int temp[],int left,int right){
    int mid;
    int count=0;
    if(right>left){
        mid=(left+right)/2;
        count=merge_count(arr,temp,left,mid);
        count+=merge_count(arr,temp,mid+1,right);
        count+=merge(arr,temp,left,mid+1,right);
    }
    return(count);
}



int main(){
    int arr[5]={2,4,3,1,5};
    int temp[5];
    int p=merge_count(arr,temp,0,4);
    printf("Ans: %d",p);
}
```