Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<stdlib.h>
int compare(const void*a,const void*b) {
    return(*(int*)a-*(int*)b);
}
int main()
{
    int T;
    scanf("%d",&T);
    while(T--) {
        int N,M;
        scanf("%d %d",&N,&M);
        int arr[N];
        for(int i=0;i<N;i++) {
            scanf("%d",&arr[i]);}
            qsort(arr,N,sizeof(int),compare);
            int minsum=0,maxsum=0;
            for(int i=0;i<N-M;i++)
            {
                minsum+=arr[i];}
                for(int i=M;i<N;i++) {
                    maxsum+=arr[i];}
                    printf("%d\n",maxsum-minsum);}}
```

	Input	Expected	Got	
✓	1 5 1 1 2 3 4 5	4	4	✓

Passed all tests! ✓

```c
#include<stdio.h>
#include<stdlib.h>
int compare (const void*a,const void*b){
    return(*(int*)a-*(int*)b);
}
int main(){
    int n;
    scanf("%d",&n);
    int vac[n],mid[n];
    for(int i=0;i<n;i++){
        scanf("%d",&vac[i]);
    }
    for(int i=0;i<n;i++){
        scanf("%d",&mid[i]);
    }
    qsort(vac,n,sizeof(int),compare);
    qsort(mid,n,sizeof(int),compare);
    int cancure=1;
    for(int i=0;i<n;i++){
        if(vac[i]<=mid[i]){
            cancure=0;
            break;
        }
    }
    if(cancure){
        printf("Yes\n");
    }
    else{
        printf("No\n");}
}
```

	Input	Expected	Got	
✓	5 123 146 454 542 456 100 328 248 689 200	No	No	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define Max 1000003
typedef struct{
    int key;
    int value;
}HashEntry;
HashEntry hashTable[Max];
int hash(int key){
    return(key%Max+Max)%Max;
}
void insert(int key,int*count){
    int hashindex=hash(key);
    while(hashTable[hashindex].key!=0&&hashTable[hashindex].key!=key){
        hashindex=(hashindex+1)%Max;
    }
    if(hashTable[hashindex].key==0){
        hashTable[hashindex].key=key;
        hashTable[hashindex].value=1;
    }
    else{
        *count
        +=hashTable[hashindex].value;
        hashTable[hashindex].value++;
    }
}
int main(){
    int n;
    scanf("%d",&n);
    int *arr=
    (int*)malloc(n*sizeof(int));
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    memset(hashTable,0,sizeof(hashTable));
    int paircount=0;
    for(int i=0;i<n;i++){
        insert(arr[i],&paircount);
    }
    printf("%d\n",paircount);
    free(arr);
}
```

	Input	Expected	Got	
✓	5 1 3 1 4 3	2	2	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<stdlib.h>
typedef struct{
    int value;
    int index;
}
Element;
int compare(const void*a,const void*b){
    Element*elem1=(Element*)a;
    Element*elem2=(Element*)b;
    return elem1->value-elem2->value;
}
int main(){
    int m;
    scanf("%d",&m);
    Element arr[m];
    for(int i=0;i<m;i++){
        scanf("%d",&arr[i].value);
        arr[i].index=i;
    }
    qsort(arr,m,sizeof(Element),compare);
    for(int i=0;i<m;i++){
        printf("%d ",arr[i].index);
    }
    printf(" \n");
}

```

	Input	Expected	Got	
✓	5 4 5 3 7 1	4 2 0 1 3	4 2 0 1 3	✓

Passed all tests! ✓