typedef struct {

    int \*stack;

    int top;

} MinStack;

MinStack\* minStackCreate() {

    MinStack\* obj=malloc(sizeof(MinStack));

    (\*obj).top=-1;

    (\*obj).stack=malloc(1\*sizeof(int));

    return(obj);

}

void minStackPush(MinStack\* obj, int val) {

    (\*obj).stack=realloc((\*obj).stack,(++(\*obj).top+1)\*sizeof(int));

    \*((\*obj).stack+(\*obj).top)=val;

}

void minStackPop(MinStack\* obj) {

    (\*obj).top=(\*obj).top-1;

}

int minStackTop(MinStack\* obj) {

    return(\*((\*obj).stack+(\*obj).top));

}

int minStackGetMin(MinStack\* obj) {

    int i;

    int min=\*((\*obj).stack+0);

    for(i=1;i<=(\*obj).top;i++)

    {

        if(\*((\*obj).stack+i)<min)

            min=\*((\*obj).stack+i);

    }

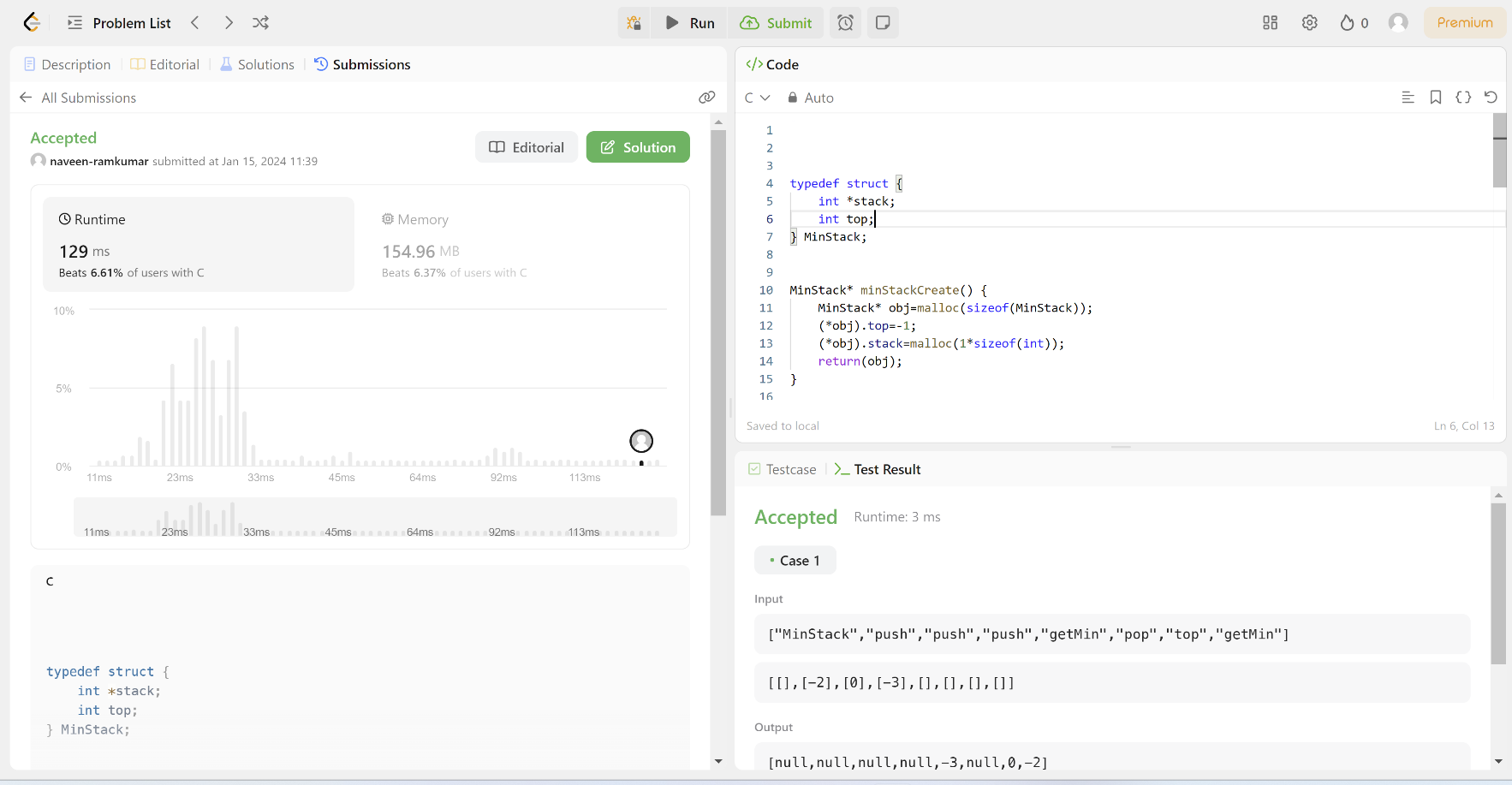
    return(min);

}

void minStackFree(MinStack\* obj) {

    free(obj);

}

****