#define LEN 0xffff

typedef struct {

    int \*data;

    int top;

} MinStack;

MinStack\* minStackCreate() {

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

    obj->data = malloc(sizeof(int)\*LEN);

    obj->top = -1;

    return obj;

}

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

    obj->top++;

    obj->data[obj->top] = val;

    //printf("top=%d -> Push=%d ",obj->top,obj->data[obj->top]);

}

void minStackPop(MinStack\* obj) {

  obj->top--;

}

int minStackTop(MinStack\* obj) {

  return obj->data[obj->top];

}

int minStackGetMin(MinStack\* obj) {

    int min= obj -> data[0];

    //printf("top=%d ",obj->top);

    for(int i=0; i <= obj->top ;i++){

        //printf("i=%d=>%d ",i,obj->data[i]);

        if( obj->data[i] < min){

            min=obj->data[i];

        }

    }

  return min;

}

void minStackFree(MinStack\* obj) {

    free(obj);

}

/\*\*

 \* Your MinStack struct will be instantiated and called as such:

 \* MinStack\* obj = minStackCreate();

 \* minStackPush(obj, val);

 \* minStackPop(obj);

 \* int param\_3 = minStackTop(obj);

 \* int param\_4 = minStackGetMin(obj);

 \* minStackFree(obj);

\*/