

שאלה 2 (15 נקודות)

בשאלה זו נתייחס למימוש המחסנית שהוצג בהרצאות, ונרחיב את הדוגמה כבתרגיל בית 6.

א. [9 נקודות] בסעיף זה הנכם מתבקשים לממש את הפעולה `st_push` עבור מחסנית אשר גודלה המירבי אינו מוגבל. נתונה לכם דוגמת מימוש של הפונקציה `st_push` עבור מחסנית שגודלה נקבע בזמן ההקצאה. עליכם לממש את אותה הפונקציה עבור מחסנית שגודלה משתנה בהתאם לתכולתה. להזכירכם כאשר המחסנית מלאה עליכם להגדיל את המחסנית פי 2 מגודלה הנוכחי (תוך שמירה על תכולתה).

```
typedef struct stack {  
    double *    array;  
    unsigned int top;  
    unsigned int alloc_size;  
} stack_t;
```

```
typedef  
enum { ST_NULL_STACK = -1, ST_OK, ST_MEMORY, ST_EMPTY, ST_OVERFLOW }  
st_code_t;
```

```
st_code_t st_push (stack_t * stack, double obj) {  
    if ( !stack ) return ST_NULL_STACK;  
    if ( stack->top == stack->alloc_size ) return ST_OVERFLOW;  
    stack->array[stack->top++] = obj;  
    return ST_OK;  
}
```

רשמו את הפתרון כאן :

```
st_code_t st_push (stack_t * stack, double obj)  
{
```

```
}
```

ב. [6 נקודות] נתון לכם המימוש של הפונקציה `st_alloc()` . כתבו את המימוש של הפונקציה `st_free()` .

```
stack_t* st_alloc(int size)
{
    stack_t* stack = (stack_t*)malloc(sizeof(stack_t));
    if(stack)
    {
        stack->array = (double*)malloc(sizeof(double)*size);
        if(!stack->array)
        {
            free(stack);
            return NULL;
        }
        stack->top = 0;
        stack->alloc_size = size;
    }
    return stack;
}
```

רשמו את הפתרון כאן

```
void st_free(stack_t* st)
{

```

```
}
```