

פתרון

חלק א

- א. 1.
- ג. 2.
- ב. 3.
- ב. 4.
- ג. 5.
- א. 6.
- א. 7.

התוכנית אמורה לעבור על מילה והופכת אות גדולה לקטנה

ב. יש להוריד ב if את הסוגריים

```
if((*str>64) &&(*str<91))
    *str+=32;
str++;
```

- ג. 8.
- ג. 9.
- ג. 10.

חלק ב

11.

א. נרצה לשמור את מספר האפשרויות ואת העמודות התפוסות כמשתנים סטטיים.

ב.

```
int countRooks(int row){
    static int takenColumn[N] = {false};
    static int count = 0;

    if(row == N){
        count++;
        return 0;
    }

    for(int column=0; column<N; column++){
        if(!takenColumn[column]){
            takenColumn[column]=true;
            countRooks(row+1);
            takenColumn[column]=false;
        }
    }

    return count;
}
```

.12

```
typedef struct ListNode{
    void (*callback)();
    struct ListNode* next;
} ListNode;

void consume(ListNode** head){
    (*head)->callback();
    ListNode* temp=*head;
    *head=(*head)->next;
    free(temp);
}
```

.13 .x

```
typedef struct heap{
    int arr[100];
    int size;
}heap;
```

.ب

```
void swap(int* a, int* b){
    int temp=*a;
    *a=*b;
    *b=temp;
}

int getParent(int index){
    if (index == 0)
        return index;

    return index%2 ? index/2 : index/2-1;
}

void addHeap(heap* mHeap, int elem){
    mHeap->arr[mHeap->size]=elem;

    //put in the last place
    int index = mHeap->size;
    int parentIndex=getParent(index);

    // heapify up
    while(elem > mHeap->arr[parentIndex]){
        swap(&mHeap->arr[parentIndex],&mHeap->arr[index]);
        index=parentIndex;
        parentIndex=getParent(index);
    }
    mHeap->size++;
}
```

חלק ג

14.

א.

מספר כתובת	כתובת	מיקום
1	arr1 (11)	גלובלי
2	&(a1) (15)	מחסנית
3	a1.arr (16)	גלובלי
4	&(a1.size) (17)	מחסנית
5	&a2 (18)	מחסנית
6	a2 (18)	לא מוגדר
7	a2 (19)	ערימה דינמית
8	arr2 (20)	מחסנית
9	a2->arr (25)	מחסנית
10	&(a2->size) (26)	ערימה דינמית

ב. אחרי שורה 36:

```
free(a2);
```

ג. הפלט הוא:

First array: 1, 2, 3, 4,

Second array: 0, 4, 8, 12,