ŞTEFAN MALANIK
12.01.2021.

DSA Exam S1

a)
```
typedef struct agenda {
    char name[35];
    char phone[10];
    int year;
    int income;
} agenda;

agenda arr[N]; // global array, N could be a define
```

b)
```
// Sort the array using select sort, O(n)=n^2, even if
// it's not the most efficient, we can't use a sort on the
// array, because we have to ignore the elements outside
// year ∈ [30,40].

void sort_arr() {
    for (int i=0; i<N; i++) {
        if(arr[i].year >40 || arr[i].year <30 ){
            continue;
        }
        for (int j=i+1; j<N; j++) {
            if (arr[i]
```

```
// else compare and swap if in reverse order
if(strcmp(arr[i].name, arr[j].name)<0){
    //reverse order - swap -> decreasing order
    agenda temp = arr[i];
    arr[i] = arr[j];
    arr[j]= temp;
}
}
}
// space complexity (n), time complexity n^2
}
```

a)
```
// most efficient way is to store the 2 agendas
// and shift the whole array by 2 at once
void move() { int a1, a2; int counter=0;
for(int i=0; i<N; i++){
    if(arr[i].income <1000) {
        if(counter==0) {
            a1 = i;
            counter++;
        }
        if(counter==1) {
            a2=i;
            break;
        }
```

a)

```
// after a1, a2 stored index, we delete shift elements
agenda ag1 = arr[a1], ag2 = arr[a2]; // store them.
for (int i = a1, i < a2-1; i++){
    arr[i] = arr[i+1];
    }
for ( int i = a2-1, i < N-2, i++){
    arr[i] = arr[i+2];
}
arr[N-2] = ag1;
arr[N-1] = ag2;
// O(n) = n - linear algorithm.
```