ep04 (20min)

I (10 val.) Considere o seguinte programa em C.

```c
#include<stdio.h>
#include<stdlib.h>

int main() {
   int n, i;
   int *values;

   scanf("%d", &n);
   values = (int*) malloc(n*sizeof(int));

   for (i = n-1; i >= 0; i--)
      scanf("%d", values+i);

   for (i = 0; i < n; i++)
      if (((*(values+i)) % 2) == 1)
         printf("%d ", values[i]);
   printf("\n");

   return 0;
}
```

Descreva o que faz o programa acima. Sabendo que o input do programa é

```
5
1 2 3 4 5
```

indique qual será o output do programa.

II (10 val.) Considere o tipo `Entry` definido em baixo que representa uma entrada de uma matriz esparsa. Cada elemento do tipo `Entry` representa o valor da matriz numa dada linha e coluna.

```c
typedef struct {
      unsigned lin, col;
      double val;
} Entry;

double* getLineValues(Entry *mat, int n, int line, int nColumns);
```

Implemente, na linguagem C, a função `getLineValues` com protótipo acima que dado o vector das entradas da matriz `mat`, a sua dimensão `n`, a linha a extrair `line` e o número de colunas da matriz `nColumns`, devolve o vector com os valores da linha `line` na matriz. Assuma que as entradas não representadas em `mat` têm valor 0.

I. Solução:

O programa lê um conjunto de inteiros e guarda-os no vector pela ordem inversa que são introduzidos. O output do programa são os números impares introduzidos pelo utilizador, mas mostrados pela ordem inversa da sua introdução.
Neste caso, o output será: `5 3 1`

II. Solução:

```
1  double* getLineValues(Entry *mat, int n, int line, int nColumns) {
2     double* values = (double*) malloc(sizeof(double)*nColumns);
3     int i;
4
5     for (i = 0; i < nColumns; i++)
6        values[i] = 0;
7
8     for (i = 0; i < n; i++)
9        if (mat[i].lin == line)
10          values[mat[i].col] = mat[i].val;
11
12    return values;
13 }
```