

Nome: _____ Número: _____ Turma: **LI31D**Considere, em todas as questões, a utilização de uma arquitectura *little endian* a 32 bits.**1.** [5] Indique o resultado da execução dos dois `PRINT_INDEX` no programa seguinte.

```
#include <stdio.h>
#define PRINT_INDEX(t, v, i) printf("0x%x\n", *((t*)(v) + (i)))
int main() {
    char a[] = { 0x1, 0x23, 0x4, 0x5, 0x67, 0x8, 0x9, 0xab, 0xcd, 0xef };
    PRINT_INDEX(short, a, 1);    PRINT_INDEX(int, a, 1);    return 0;
}
```

2. [7] Considere as tabelas de símbolos resultantes da compilação das unidades de tradução `ex1_a.c`, `ex1_b.c` e `ex1_c.c`. A ligação destas três unidades de tradução apresenta erros.

```
ex1_a.o:
00000000 D a
00000000 b b
00000000 t c
          U d
00000020 T main
```

```
ex1_b.o:
          U a
          U b
00000000 D c
00000000 T d
```

```
ex1_c.o:
00000000 D a
```

2.a [4] Identifique os erros de ligação apresentando a razão dos mesmos.

2.b [3] Sugira correcções para os erros identificados.

3. [8] Implemente a função `getOdds` que retorna um objecto do tipo `ArrayInt` com o conjunto dos inteiros ímpares presentes no objecto do tipo `ArrayInt` recebido como parâmetro. O algoritmo implementado deverá estar optimizado na utilização do recurso memória. O tipo `ArrayInt` é uma estrutura constituída pelos campos `sz` e `a`. O campo `sz` indica o número de inteiros presentes no *array* de inteiros apontado pelo campo `a`.

```
typedef struct { int sz; int * a; } ArrayInt;
ArrayInt * getOdds( const ArrayInt * v ) {
```

```
}
```