Programação em Sistemas Computacionais
Primeiro Mini-teste na turma: **LI31D**, 26 de Outubro de 2012

Nome: _____ **Número:** _____ **Turma:** LI31D |A

Considere, em todas as questões, a utilização de uma arquitectura *little endian* a 32 bits.

1. **[5]** Considere a função `count_ones`

```
int count_ones(short v) {
   char c = (char)v;
   int i = c, cnt = 0;

   while (i != 0) {
      if (i & 1) cnt++;
      i = (unsigned int)i >> 1;
   }
   return cnt;
}
```

Indique o resultado apresentado na consola depois de executado o seguinte excerto de código:

```
printf("%d\n", count_ones(0xFF75)); printf("%d", count_ones(0x0085));
```

2. **[7]** Implemente a função `strtrim` que retorna a *string* `str` sem os caracteres delimitadores presentes no início e no fim de `str`. Caracteres delimitadores presentes no meio de outros caracteres não são removidos. Os caracteres delimitadores são recebidos no parâmetro `delm`.

Nota: pode recorrer, na sua implementação, a funções da biblioteca standard do C. Sugerem-se as funções `int strlen(const char *s)` e `char * strchr(const char *s, char c)`. A segunda função retorna o ponteiro para a primeira ocorrência de `c` em `s` ou `NULL` se não existir.

```
char * strtrim(char * str, const char * delm) {




}
```

3. **[8]** Considere a definição do tipo `Person`, o programa principal e o resultado da sua execução

```
#define MAX_NAME 19
#define MAX_ADDR 19
typedef struct penson_info {
    int number;
    char name[MAX_NAME+1];
    char addr[MAX_ADDR+1];
    char age;
    char sex;
} Person;
```

```
sizeof(p) = ___
11111,Nome 1,Rua 1,10,M
22222,Nome 2,Rua 2,20,M
33333,Nome 3,Rua 3,30,F
44444,Nome 4,Rua 4,40,F
    0,Nome 0,Rua 0, 0,F
```

```
int main (int argc, char * argv[]) {
   Person p[5];
   int i;
   char lines[5][40] = {
      "11111, Nome 1 | Rua 1 | 10 | M",
      "22222, Nome 2 | Rua 2 | 20 | M",
      "33333, Nome 3 | Rua 3 | 30 | F",
      "44444, Nome 4 | Rua 4 | 40 | F",
      "00000, Nome 0 | Rua 0 | 00 | F"
   };

   printf("sizeof(p) = %d\n", sizeof(p));
   for(i=0; i < sizeof(p)/sizeof(p[0]); i++)
   {
      person_init(_____, ________);
      person_print(_____);
   }
   return 0;
}
```

a. **[1]** Indique no resultado da execução a dimensão obtida da variável `p`

b. **[4]** Implemente a função `person_init` que inicia o objecto `Person` apontado por `p` com base na informação presente na *string* `to_parse`. A *string* `to_parse` tem a formatação apresentada na definição do *array* `lines`.

Nota: pode recorrer, na sua implementação, a funções da biblioteca standard do C e à função `strtrim` implementada na questão anterior.

```
void person_init(Person *p, char * to_parse) {









}
```

c. **[2]** Implementa a função `person_print` de acordo com o formato apresentado como resultado da execução

```
void person_print(Person *p) {




}
```

d. **[1]** Indique no programa principal os argumentos que deverão ser passados nas chamadas às funções `person_init` e `person_print` por forma a obter o resultado apresentado.

Bom mini-teste