

Prova Final de Software Básico – 2002.1

Prof. Renato Cerqueira
Departamento de Informática — PUC-Rio

1. (4.0 pontos) Considere o código do programa em C abaixo:

```
#include <stdio.h>

void dump (int n, void* p) {
    unsigned char* pc = (char*) p;
    int i;
    printf("=====\n");
    printf("N. de bytes: %d\n",n);
    for (i = 0; i < n; i++,pc++)
        printf("%p - %02x\n", pc, *pc);
    printf("=====\n");
}

struct {
    char c;
    int i;
    float f;
    double g;
    short s;
} x[] = { {'e', -640, 64.53125, 0.0, 1028},
         {-300, 130, -130.7, -1.0, -2}};

int main (void) {
    dump(sizeof(x), x);
    return 0;
}
```

Considerando que x é carregado a partir do endereço 0x00237A20, mostre qual será a saída desse programa, **justificando** com os cálculos necessários.

Obs. 1: o valor ascii de 'a' é 97 (0x61).

Obs. 2: assumo a arquitetura de computador que temos visto no laboratório (int de 4 bytes, *little-endian* e alinhamento “usual”).

2. (1.0 ponto) Utilizando a linguagem C, escreva uma função para testar se sua plataforma de execução (processador) é uma arquitetura *big-endian* ou não. O protótipo dessa função deve ser

```
int isBigEndian(void);
```

Esta função deve retornar verdadeiro (1) se a plataforma de execução for *big-endian* e falso (0) caso contrário.

3. (5.0 pontos) Traduza as funções abaixo para a linguagem Assembly da família de processadores Intel Pentium, utilizando as convenções "usuais" de alinhamento e de passagem de parâmetros e resultados em C. Comente o seu código.

```
(a) struct Point {
    int x;
    int y;
};

struct Point * cria_ponto (int x, int y) {
    struct Point* p = malloc(sizeof(struct Point));
    if (x < 0) {
        p->x = 0;
        p->y = y;
    }
    else if (y < 0) {
        p->x = x;
        p->y = 0;
    }
    else {
        p->x = x+y;
        p->y = x-y;
    }
    return p;
}
```

(b) double cos (double a);

```
void calc (int n, double* r, double* a, double* b) {
    double temp;
    int i;
    for (i = 0; i < n; i++) {
        temp = 1.0 + a[i]*b[i];
        r[i] = cos(temp);
    }
}
```