

PUC-Rio – Software Básico – INF1018
Prova 1 – Turma 3wb – 30/4/2015

1. (2,5 pontos) Considere o programa C a seguir:

```
#include <stdio.h>
void dump (void *p, int n) {
    unsigned char *p1 = (unsigned char *) p;
    while (n--) {
        printf("%p - %02x\n", p1, *p1);
        p1++;
    }
}

struct X {
    char c1;
    short s;
    int i;
    char c2;
};

int main (void) {
    struct X x = {-34, 515 << 2, -1022, 'd' + 1};
    dump (&x, sizeof(struct X));
    return 0;
}
```

Supondo que `x` seja alocado na posição de memória `0xffef8344`, diga o que esse programa irá imprimir quando executado. Coloque **PP** nas posições correspondentes a *padding*.

Considere que o valor do caracter 'a' na tabela ASCII é 97, e suponha que a máquina de execução é *little-endian* e que as convenções de alinhamento são as do Linux no IA-32 (vistas em sala).

(ATENÇÃO: mostre como você chegou aos valores exibidos. Valores sem contas **NÃO valem ponto!**)

2. (2 pontos) Implemente em C uma função `compactaX`, com o seguinte protótipo:

```
void compactaX (struct X *px, int n, unsigned char *buff);
```

A função `compactaX` recebe como parâmetros o endereço de um *array* de estruturas do tipo definido na questão anterior (`struct X`), o número de estruturas nesse *array* e o endereço de um *buffer* (um *array* de bytes). Ela deve armazenar no *buffer* os dados do *array* de estruturas de forma compactada (isto é, sem *padding*) e **em ordenação *big-endian***.

Considere que o *buffer* tem espaço suficiente para armazenar os dados de todas as estruturas.

3. Traduza as funções `foo` e `boo` abaixo para assembly IA-32 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C. Traduza o mais diretamente possível o código de C para assembly.

(Não se preocupe em entender o que as funções fazem, apenas traduza-as literalmente.)

Comente seu código!

(a) (3 pontos)

```
int foo (int *v1, int n) {
    int m = 0;
    int i;
    for (i=0;i<n;i++) {
        if ((v1[i] % 2) && (v1[i] > m)) {
            m = v1[i];
        }
    }
    return m;
}
```

(b) (2,5 pontos)

```
struct x {
    int v1;
    int v2;
    struct x *next;
};

int f(int i, int j);

void boo (struct x *px, int val) {
    while (px != NULL) {
        px->v2 = f(px->v1, val);
        px = px->next;
    }
}
```

Boa Prova!