PUC-Rio – Software Básico – INF1018
Prova 2 – Turma 3wa – 16/06/2016

1. (2,0 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 {
  double d;
  float f;
  int i1, i2;
} x = {-0.375, 2051.0, -1, 256};

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

   Supondo que `x` seja alocado na posição de memória 0x601040, mostre o que esse programa irá imprimir quando executado. Considere que a máquina de execução é *little-endian* com as convenções de alinhamento do Linux no IA-64 (vistas em sala). Coloque **PP** nas posições correspondentes a *padding*.

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

2. Traduza as funções `foo` e `boo` abaixo para assembly IA-64 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/linux. 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) (2,5 pontos)

```
#define MAX 5
int g1(float i);

int boo(double *a, double *b, int n) {
  int i, indice;
  double acc = 0.0;
  for (i = 0; i < n; i++) {
    indice = g1((float)*a); a++;
    if (indice >= MAX) indice = MAX;
    acc += b[indice];
    b[indice] = *a;
  }
  return (int) acc;
}
```

(b) (2,5 pontos)

```
struct x {
  int a;
  double b;
};

double h (struct x *X, int qtos);

float boo (float f1) {
  struct x localX[10]; int i;
  struct x *plocalX = localX;
  for (i=0; i<10; i++) {
    plocalX->a = i;
    plocalX->b = f1+i;
    plocalX++;
  }
  return h(localX,10);
}
```

3. Considere o módulo abaixo:

```
#include <stdio.h>

int foo (double a);
int b = -1;

void boo (void) {
  b = 1024;
}

int main (void) {
  int res = foo(1.0);
  printf ("%d, %d\n", res, b);
  return 0;
}
```

(a) (0,8 ponto) Liste os símbolos exportados (definidos neste módulo) e importados (referências externas) para este arquivo, ou seja, o que apareceria como D (dados exportados), T (código exportado) ou U (undefined) na listagem do `nm`.

(b) (1,2 ponto) Considere agora o seguinte módulo:

```
extern char b;

int bar (void) {
  b = 0;
  return b;
}

double foo (double num) {
  bar();
  return 100+num;
}
```

Se este módulo fosse "ligado"com o módulo anterior para compor um programa executável, o que seria impresso por este programa? **Justifique a sua resposta!**

Boa Prova!