

- 1) No formato IEEE 754, números não inteiros são representados através de um sinal, um expoente e uma fração de 1, 8 e 23 bits ou 1, 11 e 52 bits para números do tipo float e double respectivamente. Considere um novo padrão de números reais de 16 bits que utiliza o mesmo algoritmo de codificação do que o formato IEEE 754, porém os campos sinal, expoente e fração possuem 1, 7, 8 bits respectivamente. Escreva em binário o seguinte número decimal no novo padrão de 16 bits, justificando o resultado com as contas necessárias.

Resultado sem justificativa não será considerado.

Número na base 10: **13,28125**

- 2) No formato IEEE 754, números não inteiros são representados através de um sinal, um expoente e uma fração de 1, 8 e 23 bits ou 1, 11 e 52 bits para números do tipo float e double respectivamente. Considere um novo padrão de números reais de 16 bits que utiliza o mesmo algoritmo de codificação do que o formato IEEE 754, porém os campos sinal, expoente e fração possuem 1, 7, 8 bits respectivamente. Escreva em binário o seguinte número decimal no novo padrão de 16 bits, justificando o resultado com as contas necessárias.

Resultado sem justificativa não será considerado.

Número na base 10: **-0,34375**

- 3) Traduza a função foo abaixo para assembly IA-64 com base no assembly visto em aula restrito às instruções que estão no PDF em [http://www4.inf.puc-rio.br/~inf1018/docs/sbinst64\\_Instrucoes\\_X64\\_padrao\\_AT\\_T.pdf](http://www4.inf.puc-rio.br/~inf1018/docs/sbinst64_Instrucoes_X64_padrao_AT_T.pdf). Não serão aceitas respostas com instruções fora desta lista ou que apresentem construções típicas de um compilador.

Utilize 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, comentando seu código, sendo que não é necessário entender a finalidade da função para responder à pergunta.

```
struct X {
    unsigned char cad;
    double val;
    struct X *prox;
};

void foo(double *v, struct X *x, int n) {
    int i=0;

    while (i < n) {
        x->cad = i;
```

```
        x->val = *v;
        x = x->prox;
        v++;
        i++;
    }
}
```

- 4) Traduza a função foo abaixo para assembly IA-64 com base no assembly visto em aula restrito às instruções que estão no PDF em [http://www4.inf.puc-rio.br/~inf1018/docs/sbinst64\\_Instrucoes\\_X64\\_padrao\\_AT\\_T.pdf](http://www4.inf.puc-rio.br/~inf1018/docs/sbinst64_Instrucoes_X64_padrao_AT_T.pdf). Não serão aceitas respostas com instruções fora desta lista ou que apresentem construções típicas de um compilador.

Utilize 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, comentando seu código, sendo que não é necessário entender a finalidade da função para responder à pergunta.

```
float boo(double d, int n) {
    int i;
    double v[n];

    v[0] = d;
    for (i=1; i<n; i++)
        v[i] = v[i-1] + i;

    return (float) v[n-1];
}
```