1) Traduza a função valfoo a seguir para assembly IA-64, utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/Linux. Não se preocupe em entender o que a função faz, mas traduza-a literalmente. O seu código não precisa compilar (Mas é boa prática testar antes de enviar, compilando e executando, comparando o seu funcionamento com o código original em C, etc.).

```
float valfoo(double a, float b, int c, double d, float e) {
    float r = (float)(c*d - a*b);
    if(r > e)
        return r;
    else
        return d;
}
```

2) Considere a sequência de 32 bits a seguir:

   01100011101011010010000111000011

   Considerando que ela representa um número em ponto flutuante com regras semelhantes às IEEE 754, mas utilizando 1 bit para o sinal, 7 bits para expoente e o restante para a parte fracionária, expresse o valor em decimal. Explique como você chegou a esse resultado.

3) Considere o fragmento de código apresentado a seguir com a chamada para a função foo no endereço 11C1:

   a) Que valor deve ser preenchido nos quatro bytes seguintes ao opcode E8 que está no endereço 11C1 para se chamar a função foo que está no endereço 1020? Explique o seu cálculo.
   b) Explique o processo de chamada de função usando o padrão exposto em aula. Inclua o processo de passagem de parâmetros de números inteiros e reais, como a função chamadora e a função chamada manipulam a pilha na chamada e no retorno da função.