

Questão 1) Dado o código a seguir, responda:

```
float funcRec1(int a) {  
    if (a == 1) {  
        return 0.5;  
    } else if (a == 0) {  
        return 0;  
    } else {  
        return funcRec1(a - 2) + 1;  
    }  
}
```

a) Quantas e quais são as condições de parada para a função funcRec1()?

Resposta: as condições de parada são duas: uma quando o valor recebido for 1 (retornando meio), e a outra quando o valor recebido for 0 (retornando zero).

b) Quantas e quais são as chamadas recursivas na função funcRec1()?

Resposta: As chamadas recursivas ocorrem quando o número fornecido for maior que 1 (considerando o contexto de receber inteiro). Um número maior faz com que a função chame-se novamente como parâmetro o 'a' do contexto menos 2. Em algum momento isso dará um ou zero e a função deixa de ser recursiva e fornece os retornos e depois se encerra.

c) Explique, com suas palavras, o que a função funcRec1() calcula.

Resposta: Aparentemente o programa calcula uma divisão por dois, já que cada chamada recursiva subtrai 2 do 'a' daquele contexto (adicionando 1 ao resultado do retorno da função chamada ali e retornando esse valor). Como consequência, sempre que a função tiver como parâmetro um número par, o '+1' da penúltima resposta (segunda chamada) irá dar zero e assim retornar zero pro contexto da primeira chamada. Se o número for ímpar, o mesmo comportamento: a penúltima resposta retornará 1, fazendo a última resposta retornar meio, já que pegando um número ímpar e sucessivamente subtraindo 1 em algum momento o resultado será 1. Como o '+1' do retorno está fora do parâmetro da função recursiva, os retornos sempre avançarão '-2' antes de dar o retorno ao contexto onde foram chamados. Se o retorno para 0 é 0 e para 1 é meio, parece uma evidência da divisão (já que qualquer ímpar dividido por dois dá um inteiro mais meio). Tomei a liberdade de testar o código com números inteiros negativos e aparentemente isso faz o código 'explodir', gerando muitas chamadas e terminando em erro. Usando alguns prints para ver os retornos (num computador de lab da PUC), a última chamada retorna o inteiro -129284, dando a entender que o erro é um estouro de pilha de funções, já que não alcança o máximo de inteiro.

Código com o teste:

```
#include <stdio.h>  
#include <stdlib.h>  
  
float funcRec1(int a){  
    if(a==1){  
        printf("0.5\n");  
        return 0.5;  
    }  
    else if(a==0){  
        printf("0\n");  
        return 0;  
    }  
    else{
```

```
        printf("(%d-2)+ 1 = %d\n", a, (a-2+1));
        return funcRec1((a-2)) + 1;
    }
}

int main()
{
    printf("Começo!\n");
    funcRec1(-1);
    return 0;
}
```