
Lista de exercícios: Ponteiros e Alocação de memória

1. Elaborar um programa que leia dois valores inteiros (a e b) na função **main**. Em seguida faça uma função que retorne $2a + 2b$. Além disso, a função deve alterar os valores de a e b do escopo **main**. Nesse sentido, a deve valer $2a$ enquanto b deve valer $2b$.
2. Crie um programa que contenha um vetor de inteiros contendo 7 elementos. Utilizando apenas aritmética de ponteiros (ou seja, navegar o vetor utilizando a propriedade de ponteiros), leia esse vetor do teclado e imprima a soma dos elementos que estão em posição par.
3. Faça uma função que receba um parâmetro inteiro n e aloca na memória **heap** um vetor de n posições e responda as questões abaixo:
 - (a) Relacione cada posição do vetor com o número da sequência de Fibonacci daquela posição. Após isso, a função deve retornar o endereço inicial do vetor.
Exemplo de vetor para $n = 3$: **[1, 1, 2]**
Exemplo de vetor para $n = 6$: **[1, 1, 2, 3, 5, 8]**
Exemplo de vetor para $n = 8$: **[1, 1, 2, 3, 5, 8, 13, 21]**
 - (b) Na função principal (main), chame a função da questão anterior (n digitado pelo usuário), e armazene o endereço retornado em um ponteiro do tipo int^* . Após isso, use esse ponteiro para imprimir os elementos do vetor.
 - (c) Na função principal (main), utilize a função **realloc** para aumentar a quantidade de posições do vetor. Nesse caso, o vetor terá duas vezes o seu tamanho original. Nas posições novas do vetor, espelhe as posições originais de tal forma que as novas posições tenham 3 vezes o valor das posições originais.
Exemplos:
O vetor **[1, 1, 2]**, depois do **realloc**, deverá ter os elementos:
[1, 1, 2, 3, 3, 6].
O vetor **[1, 1, 2, 3]**, depois do **realloc**, deverá ter os elementos:
[1, 1, 2, 3, 3, 3, 6, 9].
O vetor **[1, 1, 2, 3]**, depois do **realloc**, deverá ter os elementos:
[1, 1, 2, 3, 5, 3, 3, 6, 9, 15].
 - (d) Imprima os valores do novo vetor (após o **realloc**). Depois disso, libere o vetor da memória e encerre o programa.