

## Soma Máxima

Para conseguir chegar ao algoritmo que eu enviei eu passei por diversas versões. Na primeira eu sempre somava o numero atual à soma atual caso esta fosse positiva, isto me fez perceber que antes de somar numeros negativos à soma atual eu deveria antes verificar se esta era maior que o numero negativo.

Para isto eu criei uma variavel chamada menos e atribuía à ela a soma dos números negativos consecutivos e, ao receber um novo número positivo eu conferia se minha soma atual era maior que o valor de menos, caso não fosse eu conferia se ela era maior que a soma máxima, atualizando os valores e indices da soma máxima e zerava o indice e o valor da soma atual, caso fosse maior eu somava os valores da variavel menos e do numero atual à soma atual.

Procurei na internet sobre um algoritmo que me falaram durante uma reunião da maratona de programação e comparando ele com o meu reparei que haviam algumas coisas que eu ainda poderia otimizar, mas, como não queria copiar o algoritmo deixei o meu como estava.

Entretanto, como ele estava salvo no computador da faculdade tive que reescrevê-lo para enviar e acabei percebendo que se eu conferisse para cada número negativo lido se a soma atual é maior que seu valor absoluto para então somá-lo a soma atual e caso não fosse zerasse a soma atual e seu índice, esta nunca seria negativa o que eliminaria a necessidade da variavel menos e outros *else's*.

O pensamento por trás do algoritmo é ler cada número, verificar se este é maior que zero, caso seja soma este à soma atual e atualiza o índice da soma atual, caso seja menor ou igual a zero verifico se a soma atual é maior que a soma máxima, atualizando os valores e índices da soma máxima caso necessário, verifico então se a soma atual é maior que o valor absoluto do número negativo lido, caso seja soma este à soma atual, caso não seja eu zero a soma atual e seu índice.

Ao fim do loop se faz necessária também uma última verificação de se a soma atual é maior que a soma máxima, para o caso de que todos os números lidos sejam positivos. Por fim ocorre a impressão do resultado.

## Quadrado Mágico

A lógica deste algoritmo eu peguei a partir de leituras online e conversas com os amigos. Considerando que a lógica para os quadrados impares é a mesma e não leríamos números maiores que cinco eu fiz com que o programa chamasse uma função caso o número lido fosse impar, e outra caso fosse par.

A função ímpar primeiramente zera qualquer lixo que pode haver remanescente na matriz criada. A partir disso, iniciando da coluna do meio na primeira linha a função atribui o valor da contagem, que começa em um, para a posição alvo. O algoritmo determina a posição alvo primeiro verificando se a posição acima e à direita está ocupada, caso a posição atual esteja na linha mais acima a linha a ser verificada se torna a linha mais abaixo da matriz e, caso esteja na coluna mais à direita a coluna a ser verificada se torna a coluna mais a esquerda da matriz.

Caso a posição alvo esteja ocupada a posição uma linha abaixo da posição atual recebe o valor da contagem e se torna a posição atual. Meu algoritmo não verifica o caso da linha da posição atual ser a linha mais abaixo caso a posição alvo já esteja ocupada porque isso não ocorre no nosso espaço amostral (3 e 5).

O quadrado mágico 4x4 é obtido a partir do preenchimento de uma matriz a partir de uma contagem crescente e, em seguida, com o espelhamento de suas diagonais.

A função par verifica se o elemento está na posição diagonal principal da matriz, implementando o valor da contagem a sua posição espelhada caso esteja, utilizando a diagonal secundaria como eixo, então verifica se o elemento está na diagonal secundaria, caso esteja, implementa o valor da contagem a sua posição espelhada, utilizando desta vez a diagonal principal da matriz, em qualquer outro caso a posição recebe o valor que receberia caso a matriz estivesse sido preenchida crescentemente da esquerda para direita.