Realocação de memória é o ato de precisar definir a memória que será usada durante a execução do programa, ao contrário de ser pré-definida ao compilar o mesmo, quando menciono realocação dinâmica de memória logo devo mencionar três funções próprias, a **malloc**, **calloc**, e a **realloc**

Malloc (Memory Allocation)

Função na qual nos solicita apenas um parâmetro, onde a informamos a quantidade de itens dentro de um vetor multiplicado pelas quantidades de bytes que o mesmo ocupa, onde quase nunca sabemos e utilizamos a syntax sizeof(tipo_desejado). A função calcula logo qual a quantidade ideal de bytes que será utilizado nos entregando o espaço de memória **com lixo de memória**.

Com isso dito, sua syntax ficaria assim: **malloc(10*sizeof(int));**

Calloc (Contíguos Allocation)

Praticamente a mesma coisa que a Malloc, o que as difere é que na calloc ela usa vírgula ao invés de multiplicar

Com isso dito, sua syntax ficaria assim: **calloc(29, sizeof(int));**

Realloc(Reallocation)

Praticamente a mesma coisa que a Malloc, o que as difere é que na realloc ela pede 2 parâmetros, a origem do vetor antigo, e a nova quantidade que será necessária multiplicada pelo sizeof do tipo desejado

Com isso dito, sua syntax ficaria assim: **realloc(vet, 29*sizeof(int));**

Ao concluirmos a execução do programa, precisamos liberar essa memória que foi realocada para que se evite o vazamento de memória. Caso não seja liberada essa memória ao terminar a execução, pode resultar em consumo excessivo de memória e eventualmente esgotar recursos que estão disponíveis.

Onde sua syntax ficaria assim: **free(vetor_utilizado);**