

ESTRUTURA DE REPETIÇÃO - WHILE (ENQUANTO)	ESTRUTURA DE REPETIÇÃO - WHILE (ENQUANTO)
O que é uma Estrutura de Repetição?	Estruturas de Repetição, também conhecidas como Loopings ou Laços são funções utilizadas nas Linguagens de Programação para repetir determinado trecho de instruções por um número de vezes até que uma condição seja satisfeita - essa condição pode ser um número determinado de repetições ou um desvio condicional específico . As Estruturas de Repetição existentes são: ENQUANTO (WHILE) , REPITA (DO WHILE) , PARA (FOR) . Cada uma delas possui uma usabilidade diferente que pode ser determinada pelo número repetições ou por uma determinada condição que, caso aconteça, interrompe o laço de repetição.
O que é uma Variável Contadora?	As Estruturas de Repetição geralmente precisam de variáveis ou constantes que delimitem a quantidade de repetições, ou que acumulem valores a medida que os laços vão repetindo. Damos a essas variáveis os nomes de: Contadora e Acumuladora, elas são SEMPRE DO TIPO INTEIRO . Variáveis Contadoras são usadas para atribuir uma quantidade repetições de um looping , geralmente atribuímos a elas um determinado valor que deverá ser incrementado ou decrementado dependendo da aplicação do looping, até atingir um número limite e resultar na saída do looping. Por exemplo: enquanto(const x <= 10) faça -> x += 1; //quando " x " chegar em 10 para
O que é uma Variável Acumuladora?	Variáveis Acumuladoras, também conhecidas como Somadoras , são variáveis usadas para acumular valores cada vez que um laço é executado . Um acumulador é uma variável que recebe ela mesma mais o valor de outra variável, diferente dos Contadores, cujo valor adicionado é sempre constante . Geralmente variáveis Acumuladoras são usadas para acumular valores que serão impressos para o usuário, por exemplo: enquanto (const x <= 10) faça { x += 1 e faça acumulador+= x ; apresente acumulador na tela;} //Perceba que a cada laço o acumulador terá somado a ele o número de " x "
Como se dá a Estrutura de Repetição While?	While (Enquanto) , é uma estrutura de repetição que sempre VERIFICA ANTES de cada execução se a condição expressa nos parênteses da função é verdadeiro ou falso . Se o resultado for verdadeiro, o bloco de instrução será executado, se for falso, todo o bloco WHILE é pulado. Geralmente usamos WHILE quando queremos uma estrutura de repetição onde: Não sabemos o número de repetições e/ou possuímos uma expressão que deve ser avaliada antes de uma estrutura de repetição ser iniciada e reiniciada . Para usar laços contados em WHILE, é necessário o uso de uma variável contadora para delimitar um número limite de repetições.
Ilustre o uso de uma Estrutura de Repetição While	//Algoritmo: faz soma e pergunta se você deseja continuar a somar, enquanto você disser que sim "s", ele vai continuar somando... int main(){ int n1 , n2 , sum; char resp = 's'; while (resp == 's'){ printf("Vamos somar 2 numeros?\n"); printf("Digite o primeiro numero..."); scanf("%d", & n1); printf("\n"); printf("Digite o segundo numero..."); scanf("%d", & n2); printf("\n"); sum = n1 + n2 ; printf("A soma dos valores eh... %d\n\n", sum); printf("Deseja continuar a somar mais numeros?(s/n)\n"); scanf("%s", &resp) ; system("cls") ; } printf("\nO programa sera encerrado!!\n"); system("pause"); return 0; }
- reservado para a questão acima -	
- reservado para a questão acima -	