Na linha 1, temos a declaração da função conforme o enunciado da questão.

*(Não precisa explicar mais nada, pois você já fez isso no power point)*

Na linha 5, temos uma validação do argumento de entrada. Caso num seja menor que 100 ou maior que 500, a função vai exibir uma mensagem de erro pelo comando disp e será finalizada pelo comande return.

Na linha 11, fazemos a inicialização do argumento de saída SP = 0.

Na linha 12, inicializamos uma variável auxiliar naoeprimo = 0. Essa variável só recebe dois valores possíveis: 1 para verdadeiro e 0 para falso.

Na linha 18, temos um for na variável i que inicia em 101, de dois em dois, e vai até num-1. Esse for irá percorrer todos os dividendos do intervalo ABERTO entre 100 e num. Todos os dividendos serão ímpares, pois nenhum dividendo par neste intervalo é um número primo.

Na linha 19, fazemos naoeprimo = 0, pois a cada iteração do for, esse valor inicial deverá ser restaurado. Isto é, assumimos inicialmente que o dividendo i é primo, caso ele falhe numa condição adiante, seu o valor de naoeprimo será alterado para 1.

Na linha 20, temos outro for, na variável j, que percorre os divisores entre 3 e a metade do dividendo de cada iteração do laço anterior. O for inicia em 3 porque não se faz necessário testar a divisão por 1, já que todo número é divisível por 1. Esse for também não contempla números pares, pois todos os dividendos são ímpares e, por isso, não são divisíveis por divisores pares. O for para em i/2 porque para divisores maiores que esse valor, as divisões inteiras nunca terão resto igual a zero, então não se faz necessário checar esses divisores.

Na linha 21, fazemos a checagem se i é ou não um número primo. Se o resto da divisão inteira do dividendo i pelo divisor j for igual a zero, isso significa que i não é um número primo, portanto, na linha 22, a variável naoeprimo é alterada para 1. Caso contrário, o for em j passa para a próxima iteração.

Na linha 22 foi adicionado um comando break para finalizar o laço em j assim que o dividendo for classificado como não primo. Isso evita checagens para divisores maiores e diminui significativamente o tempo de processamento da função.

Na linha 26, se o dividendo i passou por todas as checagens no laço em j sem ter seu valor alterado, isto é, se seu valor ainda for 0, sabemos que i é primo, então naoeprimo é igual a 0, portanto o valor de i é incrementado ao argumento de saída SP. Caso contrário, o for em i passa para a próxima iteração.

Ao final de todas as iterações em i, o valor de SP, contendo a soma de todos os números primos no intervalo testado, será retornado pela função.