Exercícios sobre Cálculo com Recursividade

1-Faça um algoritmo que peça 5 valores e os armazene em um vetor global de inteiros (pode populacionar o vetor, se quiser). E crie uma função recursiva que calcule quantos valores fora do intervalor de 3 e 8 foram digitados. Então, se o vetor tem os valores 1, 3, 4, 7, 10, a função recursiva deve retornar o número 2, porque há 2 números fora do intervalor entre 3 e 8. Faça o desenho da execução.

#include <iostream>

const int tamanhoVetor = 5;

int vetorGlobal[tamanhoVetor];

int contarForaIntervalo(int indice) {

// Caso base: fim do vetor

if (indice == tamanhoVetor) {

return 0;

}

// Verifica se o valor está fora do intervalo

if (vetorGlobal[indice] < 3 || vetorGlobal[indice] > 8) {

// Se estiver fora, soma 1 e continua com o próximo índice

return 1 + contarForaIntervalo(indice + 1);

} else {

// Se estiver dentro, continua com o próximo índice

return contarForaIntervalo(indice + 1);

}

}

int main() {

// Popula o vetor (pode ser alterado)

for (int i = 0; i < tamanhoVetor; ++i) {

std::cout << "Digite o valor " << i + 1 << ": ";

std::cin >> vetorGlobal[i];

}

// Chama a função recursiva e exibe o resultado

int resultado = contarForaIntervalo(0);

std::cout << "Quantidade de valores fora do intervalo: " << resultado << std::endl;

return 0;

}

Contagem de valores fora do intervalo (3 a 8)

Vetor de entrada: 1, 3, 4, 7, 10

contarForaIntervalo(0)

|

|-- contarForaIntervalo(1)

| |

| |-- contarForaIntervalo(2)

| | |

| | |-- contarForaIntervalo(3)

| | | |

| | | |-- contarForaIntervalo(4)

| | | | |

| | | | |-- Base case: retorna 1 (10 é fora do intervalo)

| | | |

| | | |-- Retorna 1 + 0 = 1

| | |

| | |-- Retorna 1 + 0 = 1

| |

| |-- Retorna 1 + 0 = 1

|

|-- Retorna 1 + 1 + 0 + 0 + 0 = 2

2- Faça um algoritmo que peça 5 valores e os armazene em um vetor global de inteiros (pode populacionar o vetor, se quiser). E crie uma função recursiva que calcule quantos valores são ímpares. Faça o desenho da execução.

#include <iostream>

const int tamanhoVetor = 5;

int vetorGlobal[tamanhoVetor];

int contarImpares(int indice) {

// Caso base: fim do vetor

if (indice == tamanhoVetor) {

return 0;

}

// Verifica se o valor é ímpar

if (vetorGlobal[indice] % 2 != 0) {

// Se for ímpar, soma 1 e continua com o próximo índice

return 1 + contarImpares(indice + 1);

} else {

// Se for par, continua com o próximo índice

return contarImpares(indice + 1);

}

}

int main() {

// Popula o vetor (pode ser alterado)

for (int i = 0; i < tamanhoVetor; ++i) {

std::cout << "Digite o valor " << i + 1 << ": ";

std::cin >> vetorGlobal[i];

}

// Chama a função recursiva e exibe o resultado

int resultado = contarImpares(0);

std::cout << "Quantidade de valores ímpares: " << resultado << std::endl;

return 0;

}

Contagem de valores ímpares

Vetor de entrada: 1, 3, 4, 7, 10

contarImpares(0)

|

|-- contarImpares(1)

| |

| |-- contarImpares(2)

| | |

| | |-- contarImpares(3)

| | | |

| | | |-- contarImpares(4)

| | | | |

| | | | |-- Base case: retorna 1 (3 é ímpar)

| | | |

| | | |-- Retorna 1 + 0 = 1

| | |

| | |-- Retorna 1 + 0 = 1

| |

| |-- Retorna 1 + 1 + 0 + 1 + 0 = 3

|

|-- Retorna 1 + 3 + 0 + 1 + 0 = 5

3- Faça um algoritmo que peça 5 valores e os armazene em um vetor global de inteiros (pode populacionar o vetor, se quiser). E crie uma função recursiva que retorne o menor valor digitado. Faça o desenho da execução.

#include <iostream>

#include <climits>

const int tamanhoVetor = 5;

int vetorGlobal[tamanhoVetor];

int encontrarMenor(int indice, int menorAtual) {

// Caso base: fim do vetor

if (indice == tamanhoVetor) {

return menorAtual;

}

// Atualiza o menor valor, se necessário

if (vetorGlobal[indice] < menorAtual) {

menorAtual = vetorGlobal[indice];

}

// Continua com o próximo índice

return encontrarMenor(indice + 1, menorAtual);

}

int main() {

// Popula o vetor (pode ser alterado)

for (int i = 0; i < tamanhoVetor; ++i) {

std::cout << "Digite o valor " << i + 1 << ": ";

std::cin >> vetorGlobal[i];

}

// Chama a função recursiva e exibe o resultado

int resultado = encontrarMenor(0, INT\_MAX);

std::cout << "Menor valor digitado: " << resultado << std::endl;

return 0;

}

Encontrar o menor valor

Vetor de entrada: 8, 3, 5, 2, 7

encontrarMenor(0, INT\_MAX)

|

|-- encontrarMenor(1, 8)

| |

| |-- encontrarMenor(2, 3)

| | |

| | |-- encontrarMenor(3, 3)

| | | |

| | | |-- encontrarMenor(4, 2)

| | | | |

| | | | |-- Base case: retorna 2

| | | |

| | | |-- Retorna 2

| | |

| | |-- Retorna 2

| |

| |-- Retorna 2

|

|-- Retorna 2