1. Calcular x elevado a n recursivamente:

#include <iostream>

int potencia(int x, int n) {

if (n == 0) {

return 1;

} else {

return x \* potencia(x, n - 1);

}

}

int main() {

int x, n;

std::cout << "Digite dois números (x e n): ";

std::cin >> x >> n;

int resultado = potencia(x, n);

std::cout << x << " elevado a " << n << " é: " << resultado << std::endl;

return 0;

}

2. Calcular a soma de um vetor recursivamente:

#include <iostream>

int somaVetor(int vetor[], int tamanho) {

if (tamanho == 0) {

return 0;

} else {

return vetor[tamanho - 1] + somaVetor(vetor, tamanho - 1);

}

}

int main() {

int vetor[] = {3, 2, 1, 7, 5, 9};

int tamanho = 6;

int resultado = somaVetor(vetor, tamanho);

std::cout << "A soma do vetor é: " << resultado << std::endl;

return 0;

}

3. Inverter a ordem de um vetor recursivamente:

#include <iostream>

void inverteVetor(int vetor[], int inicio, int fim) {

if (inicio < fim) {

// Trocar elementos nos índices inicio e fim

int temp = vetor[inicio];

vetor[inicio] = vetor[fim];

vetor[fim] = temp;

// Chamada recursiva para os índices restantes

inverteVetor(vetor, inicio + 1, fim - 1);

}

}

int main() {

int vetor[] = {3, 2, 1, 7, 5, 9};

int tamanho = 6;

std::cout << "Vetor original: {";

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

std::cout << vetor[i];

if (i < tamanho - 1) {

std::cout << ", ";

}

}

std::cout << "}\n";

inverteVetor(vetor, 0, tamanho - 1);

std::cout << "Vetor invertido: {";

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

std::cout << vetor[i];

if (i < tamanho - 1) {

std::cout << ", ";

}

}

std::cout << "}\n";

return 0;

}

4. Função fazAlgo:

A função fazAlgo permanece a mesma.

5. Sequência de Pell recursiva:

#include <iostream>

int sequenciaPell(int n) {

if (n == 0) {

return 0;

} else if (n == 1) {

return 1;

} else {

return 2 \* sequenciaPell(n - 1) + sequenciaPell(n - 2);

}

}

int main() {

int posicao;

std::cout << "Digite a posição na Sequência de Pell: ";

std::cin >> posicao;

int resultado = sequenciaPell(posicao);

std::cout << "O valor na posição " << posicao << " da Sequência de Pell é: " << resultado << std::endl;

return 0;

}