

```

#include <stdio.h>

/**
 * Função para zerar o vetor
 * \param inicio Ponteiro para a primeira posição do vetor
 * \param fim Ponteiro para a última posição + 1 do vetor
 */
void zeraVetor(int *inicio, int *fim)
{
    while (inicio < fim)
    {
        *(inicio++) = 0;
    }
}

/**
 * Função para imprimir o vetor
 * \param vet O vetor de números inteiros
 * \param tam O tamanho do vetor
 */
void imprimeVetor(int vet[], int tam)
{
    for (int i = 0; i < tam; i++) {
        printf("%d ", vet[i]);
    }
    printf("\n");
}

/**
 * Gerador de números pseudo-aleatórios por congruência linear
 * \return Um número pseudo-aleatório
 */
int valorAleatorio(int a, int b, int c, int d, int e)
{
    return (a * b + c) % d - e;
}

/**
 * Função recursiva que inicializa o vetor com valores pseudo-aleatórios
 * \param vetor O vetor de números inteiros
 * \param tamanho O tamanho do vetor
 * \param ultimoValor O último valor aleatório utilizado na inicialização
 * \return A soma de todos os valores inseridos no vetor
 */
int inicializaVetor(int vetor[], int tamanho, int ultimoValor)

```

```

{
    // Caso base: não há vetor!
    if (tamanho <= 0) {
        return 0;
    }

    // Passo recursivo: insere valor aleatório na última posição do vetor
    // e chama recursivamente a função para o vetor com tamanho - 1.
    int novoValor = valorAleatorio(ultimoValor, 47, 97, 337, 3);
    vetor[tamanho - 1] = novoValor;
    return novoValor + inicializaVetor(vetor, tamanho - 1, novoValor);
}

/**
 * Função que troca os valores entre duas posições do vetor
 * \param a Ponteiro para a primeira posição
 * \param b Ponteiro para a segunda posição
 */
void troca(int *a, int *b)
{
    if (a != b) {
        int aux = *a;
        *a = *b;
        *b = aux;
    }
}

/**
 * Função que ordena os elementos do vetor (SelectionSort)
 * \param vet O vetor de números inteiros
 * \param n O tamanho do vetor
 */
void ordenaVetor(int vet[], int n)
{
    int i, j, min_idx;
    for (i = 0; i < n - 1; i++) {
        min_idx = i;
        for (j = i + 1; j < n; j++) {
            if (vet[j] < vet[min_idx]) {
                min_idx = j;
            }
        }
        if (min_idx != i) {
            troca(&vet[min_idx], &vet[i]);
        }
    }
}

```

```
}
```

```
#define SIZE 20
```

```
int main()
```

```
{
```

```
    int vet[SIZE];
```

```
    int soma = inicializaVetor(vet, SIZE, 71);
```

```
    imprimeVetor(vet, SIZE);
```

```
    ordenaVetor(vet, SIZE);
```

```
    imprimeVetor(vet, SIZE);
```

```
    zeraVetor(&vet[0], &vet[SIZE]);
```

```
    imprimeVetor(vet, SIZE);
```

```
    printf("Soma: %d\n", soma);
```

```
    return 0;
```

```
}
```