



INSTITUTO FED. DE EDUCAÇÃO, CIÊNC. E TEC. DE PERNAMBUCO
CURSO: TEC. EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: ALGORITMOS E ESTRUTURAS DE DADOS
PROFESSOR: RAMIDE DANTAS
ASSUNTO: REVISÃO – ARRAYS, STRINGS E RECURSÃO

Prática 01

Qualquer IDE pode ser usada desde seja possível fazer depuração (debug). Sugestões:

- [Eclipse com CDT](#) + [MinGW](#) (local; instalação meio complicada).
- [CLion](#) (local; meio pesado, requer licença de estudante).
- [Replit.com](#) (online; requer acesso constante a internet).
- [OnlineGDB](#) (online; meio limitado mas usável).

Parte 1: Cálculos, Arrays e Recursão

Problema 1: Dado um array (ou vector) de inteiros e um número N qualquer ($N \geq 0$), escreva uma função que desloque os elementos do array N posições à direita, de forma que os elementos cuja nova posição extrapolem o tamanho do array sejam colocados ao início, ver exemplos:

Array no início:

3	5	8	10	1	7	9	4
---	---	---	----	---	---	---	---

N (descolamento): 3

Array ao final:

7	9	4	3	5	8	10	1
---	---	---	---	---	---	----	---

Uma implementação boa deve usar apenas um *loop* (*for*, *while*) e resto da divisão (operador %) para calcular as novas posições dos elementos. Pode ser usado um array/vector auxiliar.

Use o código abaixo como base. Adicione mais casos de testes.

```
#include <iostream>
#include <vector>

using namespace std;

void rotate(vector<int> &arr, int n) {
    // seu código aqui
}

int main() {
    vector<int> arr = {3, 5, 8, 10, 1, 7, 9, 4};
    rotate(arr, 3);
    for (auto & v : arr) cout << v << " ";
    return 0;
}
```

Desafio (Opcional): Adapte e submeta sua solução ao problema [189 no LeetCode](#).

Problema 2: Dado um array (ou vector) de números inteiros, escreva uma função **recursiva** que retorna o **maior** número do array.

Array de entrada:

10	1	5	3	12	7	4	6
----	---	---	---	----	---	---	---

Resultado (saída): 12

Use o código abaixo como base. Adicione mais casos de testes.

```
#include <iostream>
#include <vector>

using namespace std;

int maior(const vector<int> &arr) {
    // seu código aqui
    // a recursão pode ocorrer em outra(s) funções que voce adicionar
}

int main() {
    vector<int> arr = {10, 1, 5, 3, 12, 7, 4, 6};
    cout << maior(arr) << endl;
    return 0;
}
```

Parte 2: Strings, Arrays e Recursão

Problema 1: Dado um string com caracteres minúsculos ('a' a 'z'), reescrever essa string de forma que as letras existentes sejam “reordenadas”, isto é, apareçam em ordem alfabética. Se houver repetições no string original, devem ocorrer no string final. Exemplos:

- “casa” → “aacs”
- “escola” → “acelos”
- “ifpe” → “efip”
- “banana” → “aaabnn”

Use o código abaixo como base. Adicione mais casos de testes.

Obs.: resolver apenas com arrays/vector; não usar métodos como `std::sort()`, etc.

```
#include <iostream>
#include <vector>

using namespace std;

string reordenar(string str) {
    // seu código aqui
}

int main() {
    cout << reordenar(string("casa")) << endl;
    cout << reordenar(string("escola")) << endl;
    cout << reordenar(string("ifpe")) << endl;
    cout << reordenar(string("banana")) << endl;
}
```

Desafio (Opcional): Adapte sua solução para resolver o problema [1276 no Beecrowd](#).

Problema 2: Dado um valor N qualquer ($0 \leq N \leq 10$), escreva uma função recursiva que exibe na tela todos os strings com as sequências binárias de N bits. Exemplos:

Com N = 2, exibir: 00, 01, 10, 11

Com N = 3, exibir: 000, 001, 010, 011, 100, 101, 110, 111.

Use o código abaixo como base. Adicione mais casos de testes.

Dica: use o método `push_back()` para adicionar um elemento ao final de um `vector`.

```
#include <iostream>
#include <vector>

using namespace std;

void gerar(vector<string> &seqs, int n, string tmp) {
    // seu código aqui
}

vector<string> gerar(int n) {
    vector<string> seqs;
    gerar(seqs, n, string(""));
    return seqs;
}

int main() {
    for (auto &s : gerar(5)) {
        cout << s << endl;
    }
}
```

Desafio (Opcional): Adapte a solução e submeta para o problema [3211 no LeetCode](#).