

**Atividades ADNP**  
**Semana 09 :: Vetores de Strings e Structs**

Instruções Gerais

- Faça cada exercício em uma função distinta.
- Utilize a extensão .c e o compilador de gcc.
- Utilize o editor de sua preferência: Code Blocks, VS Code, Dev C++, etc.

1. Escreva uma função que imprime um vetor de **n** strings de tamanho **len**.  
**void printStringVector(int n, int len, char list[n][len]).**

```
Ex: char v[6][20] = {"john", "mary", "jake", "finn", "ada", "michael"};
printStringVector(6, 20, v);
```

2. Escreva uma função que imprime um vetor de **n** strings de tamanho **len**, todas em letras maiúsculas.  
**void printStringVector(int n, int len, char list[n][len]).**

```
Ex: char v[6][30] = {"Rockman", "Mad Stalker", "Valis III", "Vectorman 2",
                    "Assault Suits Valken", "Dracula X"};
printStringVector(6, 30, v);
```

Exemplo de saída:

ROCKMAN - MAD STALKER - VALIS III - VECTORMAN 2 - ASSAULT SUITS VALKEN - DRACULA X

3. Escreva uma função que imprime uma matriz de strings.  
**void printStringMatrix(int rows, int cols, int len, char m[rows][cols][len]).**

```
Ex: char v[4][3][20] = {
    {"John", "Lara", "Emma"},
    {"Paul", "Pamela", "Joanne"},
    {"Jones", "Anne", "Fin"},
    {"Jane", "Jake", "Jayne"}
}
printStringMatrix(4, 3, 20);
```

4. Escreva uma função que imprime uma matriz de strings, sendo cada string impressa ao contrário.  
**void printRevStringMatrix(int rows, int cols, int len, char m[rows][cols][len]).**

```
Ex: char v[2][3][20] = {
    {"John", "Lara", "Emma"},
    {"Paul", "Pamela", "Joanne"}
}
printRevStringMatrix(2, 3, 20);
```

Exemplo de saída:

```
nhoJ    araL    ammE
luaP    alemaP  ennaoJ
```

5. Escreva uma função que recebe os dados de uma pessoa (struct Person) e imprime seu nome, a quantidade de palavras e sua idade (com base na data de nascimento e na data atual). Como data atual, considere dia, mês e ano do momento em que desenvolver o exercício. Ex: ano = 2020, mes = 10, dia = 02.

```
void printPerson(struct Person p);
```

```
struct person {
    char name[50];
    int year; // dia nasc
    int month; // mes nasc
    int day; // ano nasc
};
```

Exemplo de uso na main():

```
struct Person p = {"Joao da Silva Sauro Júnior", 1999, 5, 12};
printPerson(p);
```

Saída:

```
Nome: Joao da Silva Sauro Júnior
Palavras: 5
Idade: 21
```

6. Considerando o exercício anterior, escreva uma função que recebe um vetor de pessoas e seu tamanho. A função deve imprimir nome, quantidade de palavras e idade de cada pessoa.

```
void printPersonVector(int n, struct Person v[n]);
```

7. Escreva uma função que recebe um vetor de empregados (struct Employee) e seu tamanho. A função deve calcular e imprimir: o total dos salários, a média dos salários e as médias dos salários por tipo de empregado: (D)eveloper, Desi(g)ner, (M)anager ou (S)upport.

```
void printReport(int n, struct Employee v[n])
```

```
struct Employee {
    char name[50];
    float salary;
    char type;
    // 'D' - Developer, 'G' - Designer, 'M' - Manager ou 'S' - Support
};
```

8. Escreva uma função que imprime os dados de um vetor de contatos (nome, e-mail e telefones).

```
void printContactVector(int n, struct Contact v[n])
```

```
struct Phone {
    char ddd[3];
    char number[10];
    char type[20]; // "home", "work" ou "mobile"
};
```

```
struct Contact {
    char name[50];
    char email[70];
    struct Phone phone1;
    struct Phone phone2;
};
```

Exemplo de uso na main():

```
struct Contact v[3];
```

```
struct Contact c1 = {"Marta Alias", "marta@gmail.com",
                    {"44", "20456772", "work"},
                    {"44", "995769634", "mobile"}
};
```

```
struct Contact c2 = {"Carlos Tunes", "joao@gmail.com",
                    {"44", "40456482", "home"},
                    {"44", "48769084", "work"}
};
```

```
struct Contact c3 = {"Joao Short", "joao@gmail.com",
                    {"44", "30556702", "home"},
                    {"44", "900869034", "mobile"}
};
```

```
v[0] = c1;
v[1] = c2;
v[2] = c3;
```

```
printContactVector(3, v);
```