

Parte 1 – Vetores de Caracteres (também conhecido como Arrays de Caracteres)

Vetores são parte essencial de uma linguagem de programação. Como definição, um vetor pode ser compreendido como um ou mais itens (de mesmo tipo – ou tipos diferentes). Nesse caso, entende-se por **Tipo Homogêneo** aquele vetor que contém um mesmo tipo de dado (i.e., apenas Inteiros e/ou Caracteres). Entende-se por **Tipo Heterogêneo** aquele vetor que contém tipos de dados diferentes (i.e., um vetor de Ponteiros para registros e/ou outras partes de células de memória não semelhantes). Geralmente os tipos de dados heterogêneos são usados em programação dinâmica via alocação de memória dinâmica. Por outro lado, dados do tipo homogêneo são dados estáticos previamente alocados em memória. Vale destacar, que o uso de tipos heterogêneos são comumente empregados em programação orientada a objetos através da técnica de Polimorfismo – quando um tipo de variável ou vetor pode ser instanciado (criado) em memória a partir de tipos diferentes.

Nessa aula vamos explorar um **Tipo de Dado Homogêneo** muito usado em programação e que serve referência para o entendimento de vários conceitos – **Vetores de Caracteres**.

Vamos iniciar a discussão a partir do exemplo abaixo, que ilustra em Pseudocódigo (Algoritmo) a ideia básica para a leitura de uma linha a partir de um arquivo e que ao final imprime a maior linha encontrada.

```
ENQUANTO ( EXISTIR OUTRA LINHA )
  SE ( A LINHA LIDA FOR MAIOR QUE ANTERIOR ) ENTÃO
    SALVE A LINHA ATUAL;
    SALVE O SEU TAMANHO;
  FIM_SE
FIM_ENQUANTO

IMPRIMIR A MAIOR LINHA
```

Brainstorn da solução

Para o desenvolvimento do algoritmo acima em Linguagem C precisaremos de algumas peças e partes importantes no código. Dessa forma, sugere-se a criação de uma função **getline()** para fazer a leitura da linha atual de entrada (arquivo por exemplo). Como retorno espera-se a função retorne o tamanho da linha atual na forma. Uma outra função conveniente seria a função **copy()** que copia uma cadeia de caracteres (string) de uma variável para outra. Em seguida, podemos unir tudo em uma rotina principal **main()**

Parte 2 – Exemplo Vetore de Caracteres - maxline.c

```
#include <stdio.h>
#define MAX_LINE 1000    // define o tamanho máximo do vetores de caracteres a serem
lido via I/O

/* declaração das rotinas a serem usadas*/
int getline(char line[], int maxline);
void copy(char to[], char from[]);

/* programa principal */
int main(){
    int len;
    int max;
    char line[MAX_LINE];
    char longest[MAX_LINE];

    max = 0;
    while( (len = getline(line, MAXLINE)) > 0 )
        if (len > max ){
            max = len;
            copy(longest, line);
        }
    if( max > 0)
        printf("%s", longest);

    return 0;
}

int getline(char s[], int lim){
    int c, i;
    for(i=0; i<lim-1 && (c=getchar())!=EOF && c!='\n'; i++)
        s[i]=c;
    if (c == '\n'){
        s[i]=c;
        i++;
    }
    s[i]='\0';
    return i;
}

void copy(char to[], char from[]){
    int i;
    i = 0;
    while( (to[i]=from[i]) !='\0' ) i++
}
```

Parte 3 – Compilação, Execução e Teste do Programa (maxline.c)

Na aula 6 no ambiente do Git você encontrará a seguinte listagem:

- **caso-teste-01-esperado.txt**
- **caso-teste-01.txt**
- **caso-teste-02-esperado.txt**
- **caso-teste-02.txt**
- **maxline.c**
- **tester.sh**

Para compilar o programa use a seguinte sintaxe:

```
gcc maxline.c -o maxline
```

Para executar e use a seguinte sintaxe (use um dos exemplos de caso de teste disponíveis)

```
./maxline < caso-teste-01.txt
```

Para executar e validar o programa compilado use o script **tester.sh**

```
└─ $ ▶ ./tester.sh
SCRIPT PARA TESTAR PROGRAMA EM C ANSI - LINUX

CASO DE TESTE 1 ..... PASSOU
CASO DE TESTE 1 ..... PASSOU
```

Parte 4 – Exercícios

1) Considere a seguinte imagem:

"hello\n"

| | | | | | | |
|---|---|---|---|---|----|----|
| h | e | l | l | o | \n | \0 |
|---|---|---|---|---|----|----|

- Explique o seu entendimento sobre o a imagem acima no contexto apresentado da aula.
- O que significa os marcadores especiais **\n** e **\0**
- Seria possível fazer um programa em C que use uma variável para armazenar as respectivas sequências de caracteres especiais **\n** e **\0** (sugestão use constantes como **#define**)
- Faça um programa em C que armazene e imprima a sequência de strings (vetores de caracteres acima)

2) Modifique o exemplo o exemplo acima (maxline.c) para que seja impresso também o tamanho da da maior linha de string lida do arquivo

- 3) Modifique o programa apresentado em aula (maxline.c) para usar constantes com o comando de pre-directiva **#define**
- 4) Faça um programa em C para imprimir todas as linhas que sejam maior que 50 caracteres.
- 5) Faça um programa em C para remover as linhas em branco de um arquivo de entrada
- 6) Faça uma função em C – **reverse()** que recebe uma string s de tamanho limitado a MAX e inverta essa string.
- 7) Para os exemplos apresentados nessa aula – use o script **valida.sh** para adicionar casos de teste para seu programa que está sendo testado.