

Parte 1 – Vetores, Matrizes e Funções

A listagem de código a seguir (vide Listagem 1 abaixo) apresenta um programa em C com uma série de conceitos importantes. Em especial, são definidas Funções (primeira parte do código) e em seguida variáveis globais e locais usadas como variáveis unidimensionais e n-dim (vetores).

```
#include <stdio.h>

int dec2digit(int c){
    return -1;
}

int is_digit(int c){
    return 0;
}

int main(){
    int c,i, nwhite, nother;
    int ndigit[10];
    int digit;
    digit = nwhite = nother = 0;
    for (i = 0; i < 10; ++i)
        ndigit[i] = 0;

    while ((c = getchar()) != EOF){
        digit = c;

        if (is_digit(digit)){
            ++ndigit[dec2digit(digit)];
        }
        else if (c == '\0' || c == '\n' || c == '\t')
            ++nwhite;
        else
            ++nother;
    }
    printf("digits =");
    for (i = 0; i < 10; ++i)
        printf(" %d", ndigit[i]);

    printf(", white space = %d, other = %d\n", nwhite, nother);

    return 0;
}
```

Parte 2 – Execução

A partir do código acima crie em seu diretório de trabalho um arquivo com o nome **vet.c** e procure compilar e tentar identificar o que faz o programa.

Preencha o código com comandos de comentário múltiplos procurando descrever o comportamento geral do programa.

```
/*O programa é responsável por ...*/
```

Como pode-se perceber o programa executa em Loop e em seguida fica esperando uma entrada do usuário. A entrada pode ser qualquer sequência de caracteres ou até mesmo um arquivo.

Tente executar o comando a seguir (apresentado em negrito)

```
erik @ einstein ~/Workspace/Git/progc/aula4 (master)
└─ $ ▶ cat /proc/cpuinfo | ./vet
digits = 32 38 31 28 38 17 37 13 26 12, white space = 240, other = 3552
```

Outra opção é executar o comando compilado com a seguinte entrada e em seguida pressionar **CTRL+D**

```
erik @ einstein ~/Workspace/Git/progc/aula4 (master)
└─ $ ▶ ./vet
3
3 3 3 3 4
digits = 0 0 0 0 0 0 0 0 0 0, white space = 2, other = 4
```

Parte 3 – Questões

A partir do código acima pede-se:

1. Modifique a rotina **dec2digit** para que ela retorne o respectivo dígito a partir da codificação da tabela ASCII (use `man ASCII` na linha de comando para saber os relacionamentos).
2. Modifique a rotina **is_digit** para que ela retorne se o caractere informado é um dígito ou não (nesse caso, 1=TRUE e 0=FALSE).
3. Crie uma outra rotina para retornar 1 se for um caractere de nova linha, tabulação ou um caractere vazio. Troque a respectiva linha do código por essa chamada.