Estruturas de dados homogêneas: vetores e matrizes

Tratam-se de estruturas de dados que comportam elementos que possuem o mesmo tipo de dados.

VETORES (ARRAYS UNIDIMENSIONAIS)

Correspondem a posições de memória, identificadas por um único nome, individualizadas por índices, e cujo conteúdo é de um mesmo tipo.

Exemplificando: Numa fila de espera para atendimento num consultório médico, onde há 10 pessoas esperando na fila para serem atendidas, sendo que seus nomes estejam armazenadas em uma variável composta chamada Nome.

Nome									
Ana	Carlos	Bianca	Carolina	Diego	Denise	Pedro	Paulo	José	João

Para fazer referência ao primeiro nome, poderia ser escrito: Nome[0] (“Ana”).
(Observação: em Linguagem C a primeira posição de um vetor é zero.)
Se desejarmos obter o segundo nome, escrevemos: Nome[1] (“Carlos”).
Para fazer referência ao último nome, ou seja o décimo elemento da fila, poderia ser escrito: Nome[9] (“João”).
Utilizando uma variável **indice**, por exemplo, pode-se acessar todos os nomes da fila: Nome[**indice**];
Assim, para mostrarmos os 10 nomes contidos neste vetor, poderíamos escrever o seguinte trecho de programa em Linguagem C:

```
for  ( indice=0 ;  índice<10   ;  índice++ )
     printf  (  “ Nome do %d  da fila  =  %s  “,  índice,  Nome[indice] );
```

Declaração de Arrays em Linguagem C:

Tipo de dados	Nome do tipo [quantidade_de_elementos] ;

Exemplificando:

a) Declaração de um vetor chamado **SALARIOS** que contenha **5** elementos que sejam do tipo **float**:
float SALARIOS[5];

SALARIOS				
1000,00	1500,00	769,98	2200,00	3000,00

b) Declaração de um vetor chamado **LETRAS** que contenha **26** elementos que sejam do tipo **char**:
char LETRAS[26];

LETRAS																									
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

c) Declaração de um vetor chamado **numeros** que contenha **6** elementos que sejam do tipo **int**:
int numeros[10];

numeros					
21	53	32	12	45	19

Programas-exemplo em Linguagem C:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

int   vet[5]; /* Declara um vetor de 5 posições inteiras */
int   i;      /* índice para acessar cada elemento do vetor */
/* Observação: na Linguagem C, o primeiro elemento de um vetor está no índice zero. */

main ()
{
   system ("cls");
   for  ( i=0 ; i<5 ; i++ )
   {
      printf ("\nDigite o %d numero: " , i+1 );
      fflush (stdin);      scanf ("%d" , &vet[i] );
```

```
   }

   system ("cls" );
   printf ("\nNúmeros digitados:\n " );
   for  ( i=0 ; i<5 ; i++ )
      printf (" %d  " , vet[i] );

   getch();
}
```

```
#include <stdio.h>

char  pessoa[256];
int   ind, tamanho;

main ()
{
     system ("cls");               /* limpa a tela                                          */

     printf (" Digite um nome: "); /* solicita a digitação do nome                          */
     fflush (stdin);               /* limpa o buffer do teclado                             */
     scanf  ("%s" , &pessoa);      /* captura o conteúdo digitado (com formato string - %s) */
                                   /* na variável pessoa                                    */
     tamanho = 0; /* zera a variável tamanho, pois ainda não foi contabilizada letra alguma */
     ind     = 0; /* zera a variável ind, pois, em Linguagem C, a primeira posição é ZERO   */
     while ( pessoa[ind] != '\0')  /* enquanto o caractere identificado em pessoa[ind] for  */
     {                             /* diferente de "null terminator" ('\0'):                */
         tamanho++;                /* incrementa tamanho de 1 unidade (mais um caractere)   */
         ind++;                    /* vai para a próxima posição do vetor                   */
     }

     printf("O nome digitado possui %d letras." , tamanho); /* mostra resultado */
     getch();
}
```

Particularidade da função scanf:

No programa anterior, se for digitado o nome MARIA, será exibida na tela a mensagem:
O nome digitado possui 5 letras.
Contudo, se for digitado o nome MARIA DE ALMEIDA, o programa continuará exibindo a mensagem:
O nome digitado possui 5 letras.
Aparentemente, parece que a função scanf considera que o texto digitado termina quando é encontrado o primeiro espaço em branco. Na verdade o que ocorre é que, ao digitarmos várias strings para serem capturadas com scanf (MARIA DE ALMEIDA são 3 strings), ao final de cada uma delas é incorporado o caractere que identifica o final da string ("null terminator" - '\0').
Assim, após a string MARIA ser digitada e acrescentado o espaço em branco o conteúdo da variável pessoa passa a ser apenas MARIA, sendo o restante das strings desprezadas.
Para contornar este tipo de problema recomenda-se o uso da função gets, uma vez que, para esta função, o final da string só é considerado após a digitação do enter. Assim, quando se digita MARIA DE ALMEIDA e depois o enter, aí sim, a string é finalizada.
Para testar isto no programa anterior, apenas digite troque a instrução **`scanf("%s" , &pessoa);`** por **`gets(pessoa);`**

Exercício:

O programa contabiliza todos os caracteres presentes no nome digitado.

Faça a alteração que garanta que apenas sejam contabilizadas as **letras** presentes no nome digitado.