

## Ficha de revisões Nº 14

Tenha em atenção que quando usamos os comandos de strings como strcpy, strlen, etc. temos de chamar a biblioteca #include<string.h>

1- Digite e compile o seguinte algoritmo. Tendo em conta o comando STRCPY explique por palavras suas o funcionamento do algoritmo e do respectivo comando.

```
#include <stdio.h>
#include <string.h>
main ()
{
char str1[100],str2[100],str3[100];
printf ("Digite uma string: \n");
gets (str1);
strcpy (str2,str1);
strcpy (str3,"de uma string ao vector");

printf("\n String original  --  str1 = %s",str1);
printf ("\n Cópia da string 1 -- str2= %s",str2);

printf("\n Atribuição directa str3= %s",str3);
scanf ("%d");
}
```

2- Digite e compile o seguinte algoritmo. Tendo em conta o comando STRCAT explique por palavras suas a diferença entre estes dois algoritmos.

<pre>#include &lt;stdio.h&gt; #include &lt;string.h&gt; main () { char str1[100],str2[100]; printf ("Digite uma string: "); gets (str1); strcpy (str2,"digitou a string"); <b>strcpy (str2,str1);</b> printf ("\n\n%s",str2); scanf ("%d"); }</pre>	<pre>#include &lt;stdio.h&gt; #include &lt;string.h&gt; main () { char str1[100],str2[100]; printf ("Digite uma string: "); gets (str1); strcpy (str2,"digitou a string"); <b>strcat (str2,str1);</b> printf ("\n\n%s",str2); scanf ("%d"); }</pre>
---	---

3- Digite e compile o seguinte algoritmo. Tendo em conta o comando STRLEN explique por palavras suas o funcionamento do algoritmo e do respectivo comando.

```
#include <stdio.h>
#include <string.h>
main ()
{
int size;
char str[100];
printf ("Digite com uma string: ");
gets (str);
size=strlen (str);
printf ("\n\nA string que digitou tem tamanho %d",size);
}
```

4- Digite e compile o seguinte algoritmo. Tendo em conta o comando STRCMP explique por palavras suas o funcionamento do algoritmo e do respectivo comando.

```
#include <stdio.h>
#include <string.h>
main ()
{
char str1[100],str2[100];
printf ("Digite uma string: ");
gets (str1);
printf ("\n\nDigite outra string: ");
gets (str2);
if (strcmp(str1,str2))
    printf ("\n\nAs duas strings são diferentes.");
else printf ("\n\nAs duas strings são iguais.");
}
```

5- Faça um programa que leia duas strings digitadas pelo utilizador, e armazene cada string num vector. Depois, concatene as strings lidas numa única string. Por fim apresente esta como resultado final do programa, indicando o seu tamanho. Nota as strings devem ter um espaço entre elas.

6- Digite e compile o seguinte algoritmo:

Utilize um vector 5 posições.  
Caso o utilizador atinja o limite máximo de números no vector o programa deve emitir uma mensagem para alertar o utilizador e obriga-lo a sair.  
Este programa deve ainda informar o utilizador quantos números ainda pode inserir à medida que este vai inserindo números.

```
#include <stdio.h>
#include <conio.h>
main ()
{
int num[5]; /* Declara um vector de inteiros de 100 posições */
int count=1;
int count2=0;
int totalnums;
printf("\tpara ver o resultado digite -999\n");
do
{
    if(count<5)
printf (" Digite o %dº numero: ",count);
    else
printf("NAO PODE DIGITAR MAIS \n Digite -999 para ver o resultado ");
scanf ("%d",&num[count]);
count++;
count2=5-count;
if (count2>0)
{
printf(" Atenção só pode digitar mais %d numeros\n ",count2);
}
}
while (num[count-1]!=-999);
totalnums=count-1;
printf ("\n\n\n\t Os numeros que digitou foram:\n\n");

for (count=1;count<totalnums;count++)
{
printf (" %d",num[count]);
}
scanf("%d");
}
```

7- Digite e compile o seguinte algoritmo:

Página [c430.html](#)

Faça um programa que inverta uma string: leia a string com gets e armazene-a invertida numa outra string. Use o comando for para varrer a string até o seu final.

```
#include <stdio.h>
#define MAX 31
int main()
{
char str[MAX], copia[MAX];
int i, compstr;
printf("\n\n Digite uma string (max 30 caracteres): ");
gets(str);
/* Determina o comprimento da string: */
compstr=strlen(str);
/* Inverte a string */
for(i=0; str[i]; i++)
{
copia[i]=str[compstr-i-1]; /* Linha em destaque */
}
copia[i] = '\0'; /* finaliza a string */
printf("\n\nString Invertida: %s\n", copia);
printf("%d",compstr);
scanf("%d");
}
```

Neste exercício existem alguns detalhes interessantes. Em primeiro lugar, é necessário determinar o tamanho da string. Tarefa que nos fica facilitada com o uso do strlen.

Lembre-se sempre que se temos uma string de 10 posições, o seu índice pode variar entre 0 a 9, e o '\0' vai estar na posição 9. Logo, devemos inverter a posição 8 com a 0, a 7 com a 1, e assim por diante. É por isso que a expressão fica:

```
copia[i] = str[compstr-i-1]
```

Não podemos também deixar de colocar o '\0' no final da nova string.

Traçagem da linha em destaque

```
for(i=0; str[i]; i++)
{
copia[i]=str[compstr-i-1]; /* Linha em destaque */
}
copia[i] = '\0';
```

	<b>Z</b>	<b>I</b>	<b>T</b>	<b>A</b>	/0
Posição	0	1	2	3	4
compstr	4-0-1=3	4-1-1=2	4-2-1=1	4-3-1=0	/0
Copia	<b>A</b>	<b>T</b>	<b>I</b>	<b>Z</b>	/0

7.1- Digite e compile o seguinte algoritmo:

Refaça o programa da página anterior. Use o comando while para fechar o loop

```
#include <stdio.h>
#define MAX 31
int main()
{
char str[MAX], copia[MAX];
int i = 0, compstr;
printf("\n\n Digite uma palavra (max 30 caracteres): ");
gets(str);
/* Determina o comprimento da string através
de um for sem conteúdo : no final dele, compstr
contem a posicao do '\0' da string
é bem mais facil fazer isto com o strlen*/
for(compstr=0; str[compstr]; compstr++);

while (str[i] != '\0')
{
copia[i] = str[compstr-i-1]; /* Linha em destaque */
i++;
}
copia[i] = '\0';
printf("\n\nString Invertida: %s\n", copia);
scanf("%d");
}
```