CÔNG DỤNG, CODE MINH HỌA CÁC HÀM THƯỜNG SỬ DỤNG TRONG CHUỖI

GV.Ths: Nguyễn Thái Sơn

Chú ý khi sử dụng các hàm này ta phải khai báo thư viện #include<string.h>

1.Hàm strcpy:

- Công dụng: sao chép chuỗi nguồn vào chuỗi đích.
- Cấu trúc:

  char*strcpy**(**char** *dich, **char** *nguon);

Có nghĩa là khi ta nhập vào một dãy các kí tự ở chuỗi nguồn thì nó sẽ sao chép tất cả các kí tự vừa nhập vào cái chuỗi đích.

- ví dụ như sau:

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

int main()
{
   char A[255],B[255];
   printf("Nhap chuoi: ");
   gets(A);
   strcpy(B,A);
   printf("Chuoi dich: ");
   puts(B);
   getch();
   return 0;
}
```

Chương trình trên khi ta nhập vào mảng A một dãy các kí tự là "abc" thì khi gap hàm strcpy(B,A); thì nó sẽ copy 3 kí tự "abc" từ mảng A vào mảng B.

Nếu chúng ta muốn copy n kí tự từ chuỗi nguồn vào chuỗi đích ta dùng hàm sau:

2.Hàm strncpy:

- Công dụng: sao chép n kí tự đầu tiên của chuỗi nguồn vào chuỗi đích.

- Cấu trúc:

char ***strncpy**(**char** *dich, **char** *nguon,**int** n);

3.Hàm strlen:

- Công dụng :cho biết độ dài của chuỗi s
- Cấu trúc:

int **strlen**(**char** *s)

- Ví dụ: Sử dụng hàm strlen xác định độ dài một chuỗi nhập từ bàn phím.

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

int main()
{
   char Chuoi[255];
   int Dodai;
   printf("Nhap chuoi: ");
   gets(Chuoi);
   Dodai = strlen(Chuoi);
   printf("Chuoi vua nhap:");
   puts(Chuoi);
   printf("Co do_dai
         %d",Dodai);
   getch();
   return 0;
}
```

4.Hàm strcat:

- Công dụng: ghép chuỗi nguồn vào sau chuỗi đích.
- Cấu trúc:

char *strcat(**char** *dich,**char** *nguon)

- Ví dụ: Nhập vào họ lót và tên của một người, sau đó in cả họ và tên của họ lên màn hình.

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

```
#include<string.h>

int main()
{
   char HoLot[30], Ten[12];
   printf("Nhap Ho Lot: ");
   gets(HoLot);
   printf("Nhap Ten: ");
   gets(Ten);
   strcat(HoLot,Ten); /* Ghep Ten vao HoLot*/
   printf("Ho ten la: ");
   puts(HoLot);
   getch();
   return 0;
}
```

5.Hàm strncat:

- Công dụng: ghép n kí tự đầu tiên của chuỗi vào sau chuỗi đích
- Cấu trúc:

```
char *strncat(char *dich,char *nguon,int n);
```

6.Hàm strcmp:

- Công dụng: so sánh 2 chuỗi s1 và s2
- Cấu trúc:

```
int strcmp(char *s1,char *s2);
```

Hàm sẽ trả về 1 trong các giá trị sau:

- Giá trị âm
  nếu chuỗi s1 nhỏ hơn chuỗi s2
- Giá trị 0 nếu
  hai chuỗi bằng nhau
- Giá trị dương nếu chuỗi s1 lớn hơn chuỗi s2

Ví dụ:

```
char *chu1 = "aaa", *chu2= "bbb", *chu3 = "aaa";
strcmp(chu1, chu2); //kết quả trả về - 1
strcmp(chu1, chu3); //kết quả trả về 0
strcmp(chu2, chu3); //kết quả trả về 1
```

ví dụ minh họa đây:

```
/*
Nhap danh sach ten va sap xep theo thu tu tang dan*/

#include <stdio.h>

#include <conio.h>

#include <string.h>

#define MAXNUM 5

#define MAXLEN 10

int main(void)

{

char ten[MAXNUM][MAXLEN]; //mang chuoi

char *c[MAXNUM]; //mang con tro tro den chuoi

char *ct;

int i, j, n = 0;

//nhap danh sach ten

while (n < MAXNUM)

{

printf("Nhap vao ten nguoi thu %d: ",n + 1);

gets(ten[n]);

c[n++] = ten[n]; //con tro den ten

}

//sap xep danh sach theo thu tu tang dan

for (i = 0; i < n - 1; i ++)
```

```
for (j = i + 1; j < n; j ++)

if (strcmp(c[i], c[j]) > 0)

{

ct = c[i];

c[i] = c[j];

c[j] = ct;

}

//In danh sach da sap xep

printf("Danh sach sau khi sap xep:\n");

for (i = 0; i < n; i ++)

printf("Ten nguoi thu %d : %s\n", i + 1, c[i]);

getch();

}
```

7.Hàm strlwr:

- Công dụng: chuyển tất cả các kí tự chuỗi về chữ thường
- Cấu trúc:

char ***strlwr**(**char** *s);

8.Hàm strupr :

- Công dụng: chuyển tất cả các kí tự chuỗi thường về chữ hoa
- Cấu trúc:

char *strupr(**char** *s)

- Ví dụ: Viết chương trình nhập vào một chuỗi ký tự từ bàn phím. Sau đó sử dụng hàm strupr(); để chuyển đổi chúng thành chuỗi chữ hoa.

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

int main()
{
   char Chuoi[255],*s;
   printf("Nhap chuoi: ");
   gets(Chuoi);
   s=strupr(Chuoi) ;
   printf("Chuoi chu hoa:");
   puts(s);
   getch();
   return 0;
}
```

9.Hàm strrev :

- Công dụng: đảo ngược chuỗi kí tự
- Cấu trúc:

  char ***strrev**(**char** *s);

10.Hàm strchr:

- Công dụng: trả về địa chỉ vị trí xuất hiện đầu tiên của kí tự ch trong chữ s và sẽ trả về giá trị
  NULL trong trường hợp không tìm thấy.
- Cấu trúc:

char ***strchr**(**char** *s,**int** ch);

11.Hàm strrchr:

- Cấu trúc:

char ***strrchr**(**char** *s,**char** ch);

- Công dụng: trả về địachỉ vị trí xuất hiện cuối cùng của kí tự ch trong chuỗi s. Nếu không tìm thấy hàm sẽ trả về giá trị NULL

12.Hàm strstr:

- Công dụng: trả về địa chỉ vị trí xuất hiện đầu tiên của chuỗi s1 trong chuỗi s và sẽ trả về giá trị NULL trong trường hợp không tìm thấy.
- Cấu trúc:

char ***strstr**(**char** *s, **char** *s1);

Ví dụ: Viết chương trình sử dụng hàm strstr() để lấy ra một phần của chuỗi gốc bắt đầu từ chuỗi "hoc".

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

int main()
{
   char Chuoi[255],*s;
   printf("Nhap chuoi: ");
   gets(Chuoi);
   s=strstr(Chuoi,"hoc");
   printf("Chuoi trich ra:");
   puts(s);
   getch();
   return 0;
}
```

13. Hàm memset

- Công dụng: Set num byte nhớ từ vị trí được trỏ tới bằng giá trị value
- Cấu trúc:

  void *memset (**void** *ptr, **int value**, size_t num);

14. Hàm memcpy

- Công dụng: Chép num byte từ vị trí mà source trỏ tới đến vị trí mà destination trỏ tới
- Cấu trúc:

  void ***memcpy** (**void** *destination, **const void** *source, size_t num);

15. Hàm memcmp

- Công dụng: So sánh giá trị các vùng nhớ mà ptr1 và ptr2 trỏ tới theo từng byte, sẽ dừng lại khi so sánh đủ num byte. Trả về -1 khi byte đầu tiên mà không trùng nhau

của 2 vùng so sánh của ptr1 nhỏ hơn ptr2, trả về 0 khi 2 vùng nhớ bằng nhau, trả về 1 khi byte đầu tiên mà không trùng nhau của 2 vùng so sánh của ptr1 lớn hơn ptr2

- Cấu trúc:

```
c
int memcmp(const void *ptr1, const void *ptr2, size_t num);
```

16.Hàm stricmp:

- Công dụng : So sánh 2 chuỗi không phân biệt chữ hoa chữ thường , hàm trả về tương tự strcmp.
- Cấu trúc :

```
c
int stricmp (const char * string1, const char * string2);
```

- Ví dụ:
- Ví dụ này sử dụng stricmp () để so sánh hai chuỗi.

```
#include <stdio.h>
#include <string.h>
int main (void)
{
  / * So sánh hai chuỗi như là chữ thường * /
  if (0 == stricmp ( "hello", "Hello"))
   if (0 == stricmp("hello", "HELLO"))
     printf("The strings are equivalent.\n");
  else
     printf("The strings are not equivalent.\n");
  return 0;
}
```

```
The output:
" The strings are equivalent."
```