函数名：strcpy 功能：拷贝一个字符串到另一个

```
char string[10];
char *str1 = "abcdefghi";
strcpy(string, str1);
strcpy(des, src+4, strlen(src+4));
-------------------------------------------------------
```

函数名：strcat 功 能：字符串拼接函数

```
char destination[25];
char *Borland = "Borland";
strcpy(destination, Borland);

-------------------------------------------------------
```

函数名：strchr

功 能：在一个串中查找给定字符的第一个匹配之处\

```
char a[15]="abc";
char *b;
b=strchr(a,'b');


-------------------------------------------------------
```

函数名：strcmp 功 能：串比较

```
char *a = "aaa";
char *b = "bbb";
int c;
c = strcmp(a, b);

-------------------------------------------------------
```

函数名：strncmpi

功 能：将一个串中的一部分与另一个串比较，不管大小写

用 法：int strncmpi(char *str1, char *str2, unsigned maxlen);

程序例：

```
#include <string.h>

#include <stdio.h>

int main(void)
```

```
{
   char *buf1 = "BBB", *buf2 = "bbb";
   int ptr;
   ptr = strcmpi(buf2, buf1);
   if (ptr > 0)
      printf("buffer 2 is greater than buffer 1\n");
   if (ptr < 0)
      printf("buffer 2 is less than buffer 1\n");
   if (ptr == 0)
      printf("buffer 2 equals buffer 1\n");
   return 0;
}
```

函数名: strcpy

功　能: 串拷贝

用　法: char *strcpy(char *str1, char *str2);

程序例:

```
#include <stdio.h>
#include <string.h>
int main(void)
 {
    char string[10];
    char *str1 = "abcdefghi";
    strcpy(string, str1);
    printf("%s\n", string);
    return 0;
 }
```

函数名: strcspn

功　能: 在串中查找第一个给定字符集内容的段

用　法: int strcspn(char *str1, char *str2);

程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>
int main(void)
 {
    char *string1 = "1234567890";
    char *string2 = "747DC8";
    int length;
    length = strcspn(string1, string2);
    printf("Character where strings intersect is at position %d\n", length);
    return 0;
 }
```

函数名: strdup

功　能: 将串拷贝到新建的位置处

用　法: char *strdup(char *str);

程序例:

```
#include <stdio.h>
#include <string.h>
#include <alloc.h>
int main(void)
 {
    char *dup_str, *string = "abcde";
    dup_str = strdup(string);
    printf("%s\n", dup_str);
    free(dup_str);
    return 0;
 }
```

函数名: stricmp

功　能: 以大小写不敏感方式比较两个串

用　法: int stricmp(char *str1, char *str2);

程序例:

```
#include <string.h>

#include <stdio.h>

int main(void)

{

   char *buf1 = "BBB", *buf2 = "bbb";

   int ptr;

   ptr = stricmp(buf2, buf1);

   if (ptr > 0)

      printf("buffer 2 is greater than buffer 1\n");

   if (ptr < 0)

      printf("buffer 2 is less than buffer 1\n");

   if (ptr == 0)

      printf("buffer 2 equals buffer 1\n");

   return 0;

}
```

函数名: strerror

功　能: 返回指向错误信息字符串的指针

用　法: char *strerror(int errnum);

程序例:

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

```
#include <errno.h>

int main(void)

{

   char *buffer;

   buffer = strerror(errno);

   printf("Error: %s\n", buffer);

   return 0;

}
```

函数名: strcmpi

功　能: 将一个串与另一个比较, 不管大小写

用　法: int strcmpi(char *str1, char *str2);

程序例:

```
#include <string.h>

#include <stdio.h>

int main(void)

{

   char *buf1 = "BBB", *buf2 = "bbb";

   int ptr;

   ptr = strcmpi(buf2, buf1);

   if (ptr > 0)

      printf("buffer 2 is greater than buffer 1\n");
   if (ptr < 0)
      printf("buffer 2 is less than buffer 1\n");
   if (ptr == 0)
      printf("buffer 2 equals buffer 1\n");
   return 0;
}
```

函数名: strncmp

功　能: 串比较

用　法: int strncmp(char *str1, char *str2, int maxlen);

程序例:

```
#include <string.h>
#include <stdio.h>
int  main(void)
{
   char *buf1 = "aaabbb", *buf2 = "bbbccc", *buf3 = "ccc";
   int ptr;
   ptr = strncmp(buf2,buf1,3);
   if (ptr > 0)
      printf("buffer 2 is greater than buffer 1\n");
   else
      printf("buffer 2 is less than buffer 1\n");
   ptr = strncmp(buf2,buf3,3);

   if (ptr > 0)

      printf("buffer 2 is greater than buffer 3\n");

   else

      printf("buffer 2 is less than buffer 3\n");

   return(0);

}
```

函数名: strncmpi

功　能: 把串中的一部分与另一串中的一部分比较, 不管大小写

用　法: int strncmpi(char *str1, char *str2);

程序例:

```
#include <string.h>

#include <stdio.h>

int main(void)

{

   char *buf1 = "BBBccc", *buf2 = "bbbccc";

   int ptr;

   ptr = strncmpi(buf2,buf1,3);

   if (ptr > 0)

      printf("buffer 2 is greater than buffer 1\n");

   if (ptr < 0)

      printf("buffer 2 is less than buffer 1\n");

   if (ptr == 0)
```

```
      printf("buffer 2 equals buffer 1\n");

   return 0;

}
```

函数名: strncpy

功　能: 串拷贝

用　法: char *strncpy(char *destin, char *source, int maxlen);

程序例:

```
#include <stdio.h>

#include <string.h>

int main(void)

{

   char string[10];

   char *str1 = "abcdefghi";

   strncpy(string, str1, 3);

   string[3] = '\0';

   printf("%s\n", string);

   return 0;

}
```

函数名: strnicmp

功　能: 不注重大小写地比较两个串

用　法: int strnicmp(char *str1, char *str2, unsigned maxlen);

程序例:

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

```
#include <stdio.h>
int main(void)
{
   char *buf1 = "BBBccc", *buf2 = "bbbccc";
   int ptr;
   ptr = strnicmp(buf2, buf1, 3);
   if (ptr > 0)
      printf("buffer 2 is greater than buffer 1\n");
   if (ptr < 0)
      printf("buffer 2 is less than buffer 1\n");
   if (ptr == 0)
      printf("buffer 2 equals buffer 1\n");
   return 0;
}
```

函数名: strnset

功　能: 将一个串中的所有字符都设为指定字符

用　法: char *strnset(char *str, char ch, unsigned n);

程序例:

```
#include <stdio.h>
#include <string.h>
int main(void)
{
   char *string = "abcdefghijklmnopqrstuvwxyz";
   char letter = 'x';
   printf("string before strnset: %s\n", string);
   strnset(string, letter, 13);
   printf("string after  strnset: %s\n", string);
   return 0;
}
```

函数名: strpbrk

功　能: 在串中查找给定字符集中的字符

用　法: char *strpbrk(char *str1, char *str2);

程序例:

```
#include <stdio.h>
#include <string.h>
int main(void)
{
   char *string1 = "abcdefghijklmnopqrstuvwxyz";
   char *string2 = "onm";
   char *ptr;
   ptr = strpbrk(string1, string2);
   if (ptr)
      printf("strpbrk found first character: %c\n", *ptr);
   else
```

```
      printf("strpbrk didn't find character in set\n");

   return 0;

}
```

函数名: strrchr

功　能: 在串中查找指定字符的最后一个出现

用　法: char *strrchr(char *str, char c);

程序例:

```
#include <string.h>

#include <stdio.h>

int main(void)

{

   char string[15];

   char *ptr, c = 'r';

   strcpy(string, "This is a string");

   ptr = strrchr(string, c);

   if (ptr)

      printf("The character %c is at position: %d\n", c, ptr-string);

   else

      printf("The character was not found\n");

   return 0;

}
```

函数名: strrev

功　能: 串倒转

用　法: char *strrev(char *str);

程序例:

```
#include <string.h>

#include <stdio.h>

int main(void)

{

   char *forward = "string";

   printf("Before strrev(): %s\n", forward);

   strrev(forward);

   printf("After strrev():  %s\n", forward);

   return 0;

}
```

函数名: strset

功　能: 将一个串中的所有字符都设为指定字符

用　法: char *strset(char *str, char c);

程序例:

```
#include <stdio.h>

#include <string.h>

int main(void)

{

   char string[10] = "123456789";
   char symbol = 'c';

   printf("Before strset(): %s\n", string);

   strset(string, symbol);

   printf("After strset():  %s\n", string);

   return 0;

}
```

函数名: strspn

功　能: 在串中查找指定字符集的子集的第一次出现

用　法: int strspn(char *str1, char *str2);

程序例:

```
#include <stdio.h>

#include <string.h>

#include <alloc.h>

int main(void)

{

   char *string1 = "1234567890";

   char *string2 = "123DC8";

   int length;

   length = strspn(string1, string2);

   printf("Character where strings differ is at position %d\n", length);

   return 0;

}
```

函数名: strstr

功　能: 在串中查找指定字符串的第一次出现

用　法: char *strstr(char *str1, char *str2);

程序例:

```
#include <stdio.h>

#include <string.h>

int main(void)

{

   char *str1 = "Borland International", *str2 = "nation", *ptr;

   ptr = strstr(str1, str2);

   printf("The substring is: %s\n", ptr);

   return 0;

}
```

函数名: strtod

功　能: 将字符串转换为double型值

用　法: double strtod(char *str, char **endptr);

程序例:

```
#include <stdio.h>

#include <stdlib.h>

int main(void)

{
   char input[80], *endptr;

   double value;

   printf("Enter a floating point number:");

   gets(input);

   value = strtod(input, &endptr);

   printf("The string is %s the number is %lf\n", input, value);

   return 0;

}
```

函数名: strtok

功　能: 查找由在第二个串中指定的分界符分隔开的单词

用　法: char *strtok(char *str1, char *str2);

程序例:

```
#include <string.h>

#include <stdio.h>

int main(void)

{

   char input[16] = "abc,d";

   char *p;

   /* strtok places a NULL terminator

   in front of the token, if found */

   p = strtok(input, ",");
   if (p)   printf("%s\n", p);

   /* A second call to strtok using a NULL

   as the first parameter returns a pointer

   to the character following the token  */

   p = strtok(NULL, ",");

   if (p)   printf("%s\n", p);

   return 0;

}
```

函数名: strtol

功　能: 将串转换为长整数

用　法: long strtol(char *str, char **endptr, int base);

程序例:

```
#include <stdlib.h>

#include <stdio.h>

int main(void)

{

   char *string = "87654321", *endptr;

   long lnumber;

   /* strtol converts string to long integer  */

   lnumber = strtol(string, &endptr, 10);

   printf("string = %s  long = %ld\n", string, lnumber);

   return 0;
}
```

函数名: strupr

功　能: 将串中的小写字母转换为大写字母

用　法: char *strupr(char *str);

程序例:

```
#include <stdio.h>

#include <string.h>

int main(void)

{

   char *string = "abcdefghijklmnopqrstuvwxyz", *ptr;

   /* converts string to upper case characters */

   ptr = strupr(string);

   printf("%s\n", ptr);

   return 0;

}
```

函数名: swab

功　能: 交换字节

用　法: void swab (char *from, char *to, int nbytes);

程序例:

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

```
#include <stdio.h>

#include <string.h>

char source[15] = "rFna koBlrna d";

char target[15];

int main(void)

{

   swab(source, target, strlen(source));

   printf("This is target: %s\n", target);

   return 0;

}
```