C++中常用的库函数

制作人：陈旻杰

algorithm

- 1.max,min
- 2.sort
- 3.swap
- 4.find
- 5.count
- 6.reverse
- 7.lower_bound,upper_bound
- 8.next_permutation,prev_permutation

```cpp
#include <iostream>
#include <algorithm>
using namespace std;
int main()
{
    cout<<max(1,2)<<endl;
    cout<<min(1,2)<<endl;
    int a[3]={2,1,3};
    sort(a,a+3);
    for(int i=0;i<3;i++)
    {
        cout<<a[i]<<" ";
    }
    cout<<endl;
    reverse(a,a+3);
    for(int i=0;i<3;i++)
    {
        cout<<a[i]<<" ";
    }
    cout<<endl;
    int b=1,c=2;
    swap(c,b);
    cout<<"b="<<b<<endl<<"c="<<c<<endl;
    cout<<find(a,a+3,2)-a<<endl;
    cout<<count(a,a+3,2)<<endl;
}
```

```
2
1
1 2 3
3 2 1
b=2
c=1
1
1

--------------------------------

Process exited after 0.04216 seconds
请按任意键继续. . .
```

lower_bound,upper_bound

```cpp
#include <iostream>
#include <algorithm>
using namespace std;
int main()
{
    int a[3]={2,1,3};
    sort(a,a+3);
    cout<<*lower_bound(a,a+3,2)<<endl;
    cout<<*upper_bound(a,a+3,2)<<endl;
}
```

```
2
3

--------------------------------

Process exited after (
请按任意键继续. . .
```

A 输出全排列 (20point(s))

请编写程序输出前n个正整数的全排列（$n < 10$），并通过9个测试用例（即n从1到9）观察n逐步增大时程序的运行时间。

输入格式:

输入给出正整数n（< 10）。

输出格式:

输出1到n的全排列。每种排列占一行，数字间无空格。排列的输出顺序为字典序，即序列$a_1, a_2, \cdots, a_n$排在序列$b_1, b_2, \cdots, b_n$之前，如果存在k使得$a_1 = b_1, \cdots, a_k = b_k$ 并且 $a_{k+1} < b_{k+1}$。

输入样例:

```
3
```

输出样例:

```
123
132
213
231
312
321
```

```cpp
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
int main()
{
    int m;
    int a[11];
    cin>>m;
    for(int i=1;i<=m;i++)
    {
        a[i]=i;
    }
    for(int i=1;i<=m;i++)
    {
        cout<<a[i];
    }
    cout<<"\n";
    while(next_permutation(a+1,a+m+1))
    {
        for(int i=1;i<=m;i++)
        {
            cout<<a[i];
        }
        cout<<"\n";
    }
    return 0;
}
```

```
3
123
132
213
231
312
321

--------------------------------

Process exited after 0.3915
请按任意键继续. . .
```

```cpp
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
int main()
{
    int m;
    int a[11];
    cin>>m;
    for(int i=1;i<=m;i++)
    {
        a[i]=i;
    }
    reverse(a+1,a+1+m);
    for(int i=1;i<=m;i++)
    {
        cout<<a[i];
    }
    cout<<"\n";
    while(prev_permutation(a+1,a+m+1))
    {
        for(int i=1;i<=m;i++)
        {
            cout<<a[i];
        }
        cout<<"\n";
    }
    return 0;
}
```

```
3
321
312
231
213
132
123

--------------------------------

Process exited after
请按任意键继续. . .
```

cmath

- 1.abs,fabs
- 2.fmod
- 3.pow
- 4.sqrt
- 5.rand
- 6.atof,atol

```cpp
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int a=-1;
    double b=-114.514;
    cout<<abs(a)<<" "<<fabs(b)<<endl;
    cout<<sqrt(4)<<endl;
    cout<<pow(10,2)<<endl;
    cout<<fmod(4,3)<<endl;
    cout<<rand()<<endl;//int rand(void) 返回一个范围在 0 到 RAND_MAX (值至少是 32767)之间的伪随机数
    char c[10]={"1.01"};
    cout<<atof(c)<<endl;
    cout<<atol(c)<<endl;
}
```

```
1 114.514
2
100
1
41
1.01
1
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main()
{
    int i, n;
    time_t t;
    n = 5;
    srand((unsigned) time(&t));
    for( i = 0 ; i < n ; i++ )
    {
        printf("%d\n", rand() % 50);
    }
  return(0);
}
```

```
14
2
30
40
3


--------------------

Process ex
请按任意键
```

cstring

- 1.memset
- 2.strcpy,strcat,strcmp,strlen
- 3.strncpy,strncat,strncmp
- 4.strchr,strstr
- 5.memcpy

strcpy,strncpy

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str1[] = "Sample string";
    char str2[40];
    char str3[40];
    strcpy (str2, str1);
    strcpy (str3, "copy successful");
    printf ("str1: %s\nstr2: %s\nstr3: %s\n", str1, str2, str3);

    char str4[] = "To be or not to be";
    char str5[40];
    char str6[40];
    strncpy ( str5, str4, sizeof(str4) );
    strncpy ( str6, str5, 5 );
    str3[5] = '\0';
    puts (str4);
    puts (str5);
    puts (str6);

}
```

```
str1: Sample string
str2: Sample string
str3: copy successful
To be or not to be
To be or not to be
To be

--------------------------------

Process exited after 0.04543 sec
请按任意键继续. . .
```

strcat,strncat

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str[80];
    strcpy (str, "these ");
    strcat (str, "strings ");
    strcat (str, "are ");
    strcat (str, "concatenated.");
    puts (str);

    char str1[20];
    char str2[20];
    strcpy (str1, "To be ");
    strcpy (str2, "or not to be");
    strncat (str1, str2, 6);
    puts (str1);
}
```

```
these strings are concatenated.
To be or not

--------------------------------

Process exited after 0.03413 secon
请按任意键继续. . .
```

strchr,strstr

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str[] = "This is a sample string";
    char * pch;
    printf ("Looking for the 's' character in \"%s\"...\n", str);
    pch = strchr(str, 's');
    while (pch != NULL)
    {
        printf ("found at %d\n", pch - str + 1);
        pch = strchr(pch + 1, 's');
    }
}
```

```
Looking for the ’s’ character in ”This is a sample string”...
found at 4
found at 7
found at 11
found at 18

--------------------------------

Process exited after 0.05185 seconds with return value 0
```

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str[] = "This is a simple string";
    char * pch;
    pch = strstr (str, "simple");
    puts (pch);
}
```

```
simple string

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

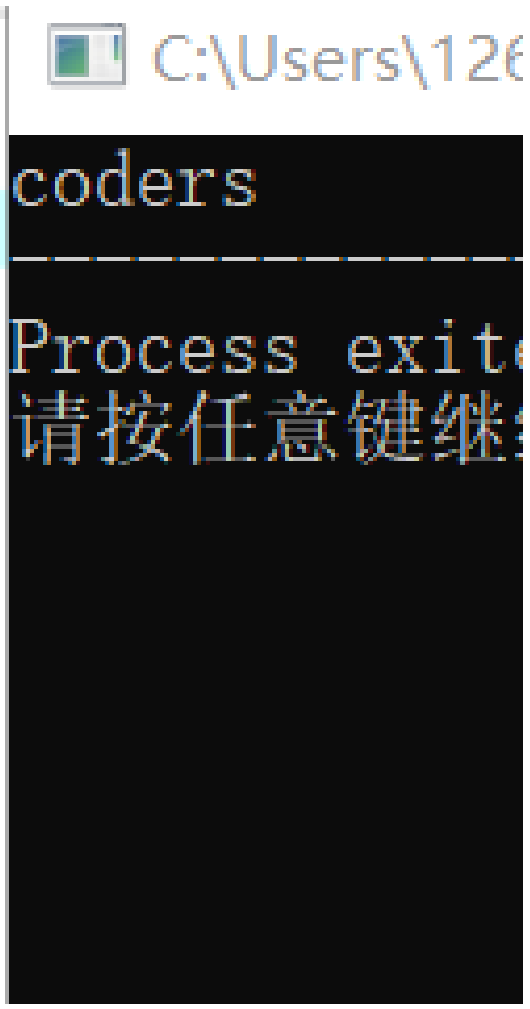
memset

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str[] = "almost every programmer should know memset!";
    memset (str, '-', 6);
    puts (str);
    memset (str, '-', sizeof(str));
    puts (str);
}
```

```
------ every programmer should know memset!
----------------------------------------------
```

memcpy

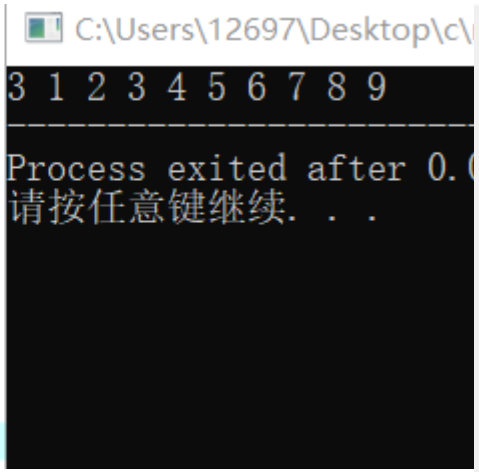
```cpp
#include <cstdio>
#include <cstring>
using namespace std;
int main()
{
   char *s="ncut coders";
   char d[20];
   memcpy(d, s+5, 6);
   d[6]='\0';
   printf("%s", d);
   return 0;
}
```



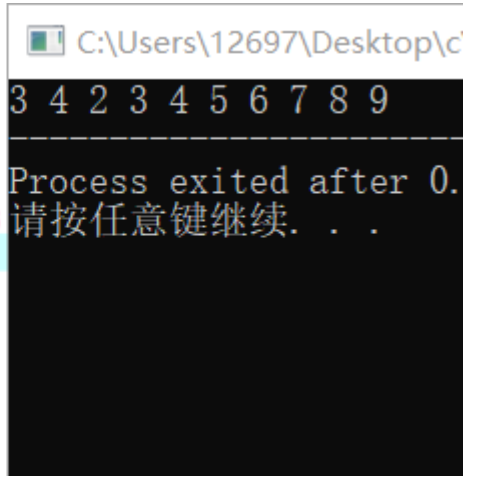
对memcpy的提醒

- memcpy() 函数的声明:
- void *memcpy(void *str1, const void *str2, size_t n)

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    int a[10] = {0,1,2,3,4,5,6,7,8,9};
    memcpy(a, a + 3, 2);
    for(int i=0;i<10;i++)
    {
        cout<<a[i]<<" ";
    }
}
```

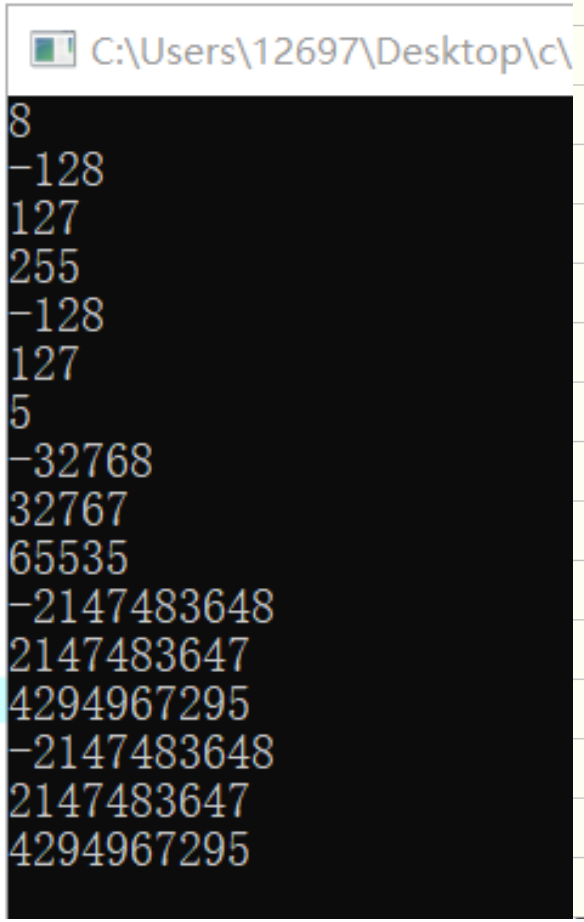



```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    int a[10] = {0,1,2,3,4,5,6,7,8,9};
    memcpy(a, a + 3, 2*sizeof(int));
    for(int i=0;i<10;i++)
    {
        cout<<a[i]<<" ";
    }
}
```



其他（limits，cctype）

```cpp
#include <iostream>
#include <climits>
using namespace std;
int main()
{
    cout<<CHAR_BIT<<endl;
    cout<<SCHAR_MIN<<endl;
    cout<<SCHAR_MAX<<endl;
    cout<<UCHAR_MAX<<endl;
    cout<<CHAR_MIN<<endl;
    cout<<CHAR_MAX<<endl;
    cout<<MB_LEN_MAX<<endl;
    cout<<SHRT_MIN<<endl;
    cout<<SHRT_MAX<<endl;
    cout<<USHRT_MAX<<endl;
    cout<<INT_MIN<<endl;
    cout<<INT_MAX<<endl;
    cout<<UINT_MAX<<endl;
    cout<<LONG_MIN<<endl;
    cout<<LONG_MAX<<endl;
    cout<<ULONG_MAX<<endl;
}
```



name	expresses	min. magnitude*
CHAR_BIT	Number of bits for a charobject (byte)	8
SCHAR_MIN	Minimum value for an object of type signed char	-127
SCHAR_MAX	Maximum value for an object of type signed char	127
UCHAR_MAX	Maximum value for an object of type unsigned char	255
CHAR_MIN	Minimum value for an object of type char	either SCHAR_MIN or 0
CHAR_MAX	Maximum value for an object of type char	either SCHAR_MAX or UCHAR_MAX
MB_LEN_MAX	Maximum number of bytes in a multibyte character, for any locale	1
SHRT_MIN	Minimum value for an object of type short int	-32767
SHRT_MAX	Maximum value for an object of type short int	32767
USHRT_MAX	Maximum value for an object of type unsigned short int	65535
INT_MIN	Minimum value for an object of type int	-32767
INT_MAX	Maximum value for an object of type int	32767
UINT_MAX	Maximum value for an object of type unsigned short int	65535
LONG_MIN	Minimum value for an object of type long int	-2147483647
LONG_MAX	Maximum value for an object of type long int	2147483647
ULONG_MAX	Maximum value for an object of type unsigned int	4294967295

cctype（由于本来函数比较易懂就不写了）

- cctype的函数为符合条件返回非0值，否则返回0，至于非零值为多少，则与本地有关。

<ctype.h>定义的宏如下表所示:

isalnum	是否为字母数字
isalpha	是否为字母
islower	受否为小写字母
isupper	是否为大写字母
isdigit	是否为数字
isxdigit	是否为16进制数字
iscntrl	是否为控制字符
isgraph	是否为图形字符（例如，空格、控制字符都不是）
isspace	是否为空格字符（包括制表符、回车符、换行符等）
isblank	是否为空白字符 (C99/C++11新增)（包括水平制表符）
isprint	是否为可打印字符
ispunct	是否为标点
tolower	转换为小写
toupper	转换为大写

这是cctype的函数实现（掩码）

_Ctype 转换表：

```
/* xctype.c _Ctype 转换表 -- ASCII 版 */
#include <limits.h>
#include <stdio.h>
#include "ctype.h"
#if EOF != -1 || UCHAR_MAX != 255
#error WRONG CTYPE table
#endif
/* 组合位 */
#define XDI (_DI|_XD)
#define XLO (_LO|_XD)
#define XUP (_UP|_XD)

/* 转换表 */
static const short ctype_tab[257] = { 0, /* EOF */
    _BB, _BB, _BB, _BB, _BB, _BB, _BB, _BB,
    _BB, _CN, _CN, _CN, _CN, _CN, _BB, _BB,
    _BB, _BB, _BB, _BB, _BB, _BB, _BB, _BB,
    _BB, _BB, _BB, _BB, _BB, _BB, _BB, _BB,
    _SP, _PU, _PU, _PU, _PU, _PU, _PU, _PU,
    _PU, _PU, _PU, _PU, _PU, _PU, _PU, _PU,
    XDI, XDI, XDI, XDI, XDI, XDI, XDI, XDI,
    XDI, XDI, _PU, _PU, _PU, _PU, _PU, _PU,
    _PU, XUP, XUP, XUP, XUP, XUP, XUP, _UP,
    _UP, _UP, _UP, _UP, _UP, _UP, _UP, _UP,
    _UP, _UP, _UP, _UP, _UP, _UP, _UP, _UP,
    _UP, _UP, _UP, _PU, _PU, _PU, _PU, _PU,
    _PU, XLO, XLO, XLO, XLO, XLO, XLO, _LO,
    _LO, _LO, _LO, _LO, _LO, _LO, _LO, _LO,
    _LO, _LO, _LO, _LO, _LO, _LO, _LO, _LO,
    _LO, _LO, _LO, _PU, _PU, _PU, _PU, _BB,
};
const short *_Ctype = &ctype_tab[1];
```