

```

1  /* Bitmap.c */
2  #include <string.h>
3  #include <assert.h>
4  #include <stdlib.h>
5  #include "Bitmap.h"
6
7  //Bitmap初始化
8  void BitmapNew(BITMAP *map, int capacity)
9  {
10     assert(capacity > 0);
11     map->capacity = capacity;
12     int charCount = (capacity + 7) / 8;
13     map->content = (char *)malloc(charCount);
14     assert(map->content);
15     memset(map->content, 0, charCount);
16 }
17
18 int BitmapCapacity(BITMAP *map)
19 {
20     return map->capacity;
21 }
22
23 //Bitmap销毁
24 void BitmapDispose(BITMAP *map)
25 {
26     free(map->content);
27 }
28
29 //将第i位置为1
30 void BitmapSet(BITMAP *map, int i)
31 {
32     if (i <= 0 || i > BitmapCapacity(map))
33     {
34         return ;
35     }
36     map->content[i >> 3] |= (0x80 >> (i & 0x07));
37 }
38
39 //将第i位置为0
40 void BitmapClear(BITMAP *map, int i)
41 {
42     if (i <= 0 || i > BitmapCapacity(map))
43     {
44         return ;
45     }
46     map->content[i >> 3] &= ~(0x80 >> (i & 0x07));
47 }
48
49 //测试第i位是否为1
50 int BitmapTest(BITMAP *map, int i)
51 {
52     if (i <= 0 || i > BitmapCapacity(map))
53     {
54         return 0;
55     }
56     return map->content[i >> 3] & (0x80 >> (i & 0x07));
57 }
58
59 //Bitmap复制操作
60 int BitmapCopy(BITMAP *mapN, BITMAP *mapO)
61 {
62     int charCountN = (mapN->capacity + 7) / 8;
63     int charCountO = (mapO->capacity + 7) / 8;
64     if (charCountN <= charCountO)
65     {
66         memcpy(mapN->content, mapO->content, charCountN);
67     }
68     else
69     {
70         memcpy(mapN->content, mapO->content, charCountO);
71         memset(mapN->content + charCountO, 0, charCountN - charCountO);
72     }
73 }

```