

## C++ Code

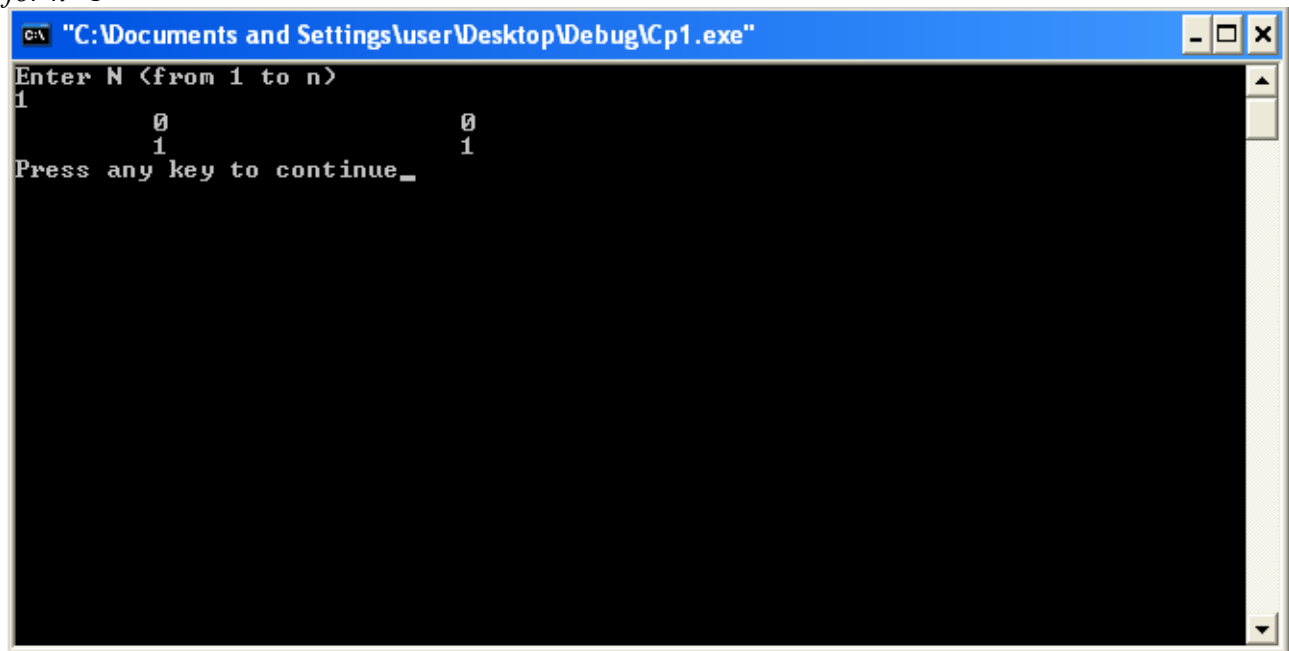
```
# include <iostream>
# include <iomanip>
# include <string>
# include <cmath>
using namespace std;

int main () {
    int N;
    cout<<"Enter N (from 1 to n)"<<endl;
    cin>>N;
    string * p;
    p= new string [pow(2,N)];
    p[0]="0";
    p[1]="1";
    for (int i=1;i<N;i++)
        for (int j=0; j<pow(2,i);j++)
            {
                int half=pow(2,i);
                p[half+j]=p[half-1-j];
                p[half+j]="1"+p[half+j];
                p[half-1-j]= "0"+p[half-1-j];
            }
    for (i=0;i<pow(2,N);i++)
        cout<<setw(10)<<i<<setw(20)<<p[i]<<endl;

    return 0;}
```

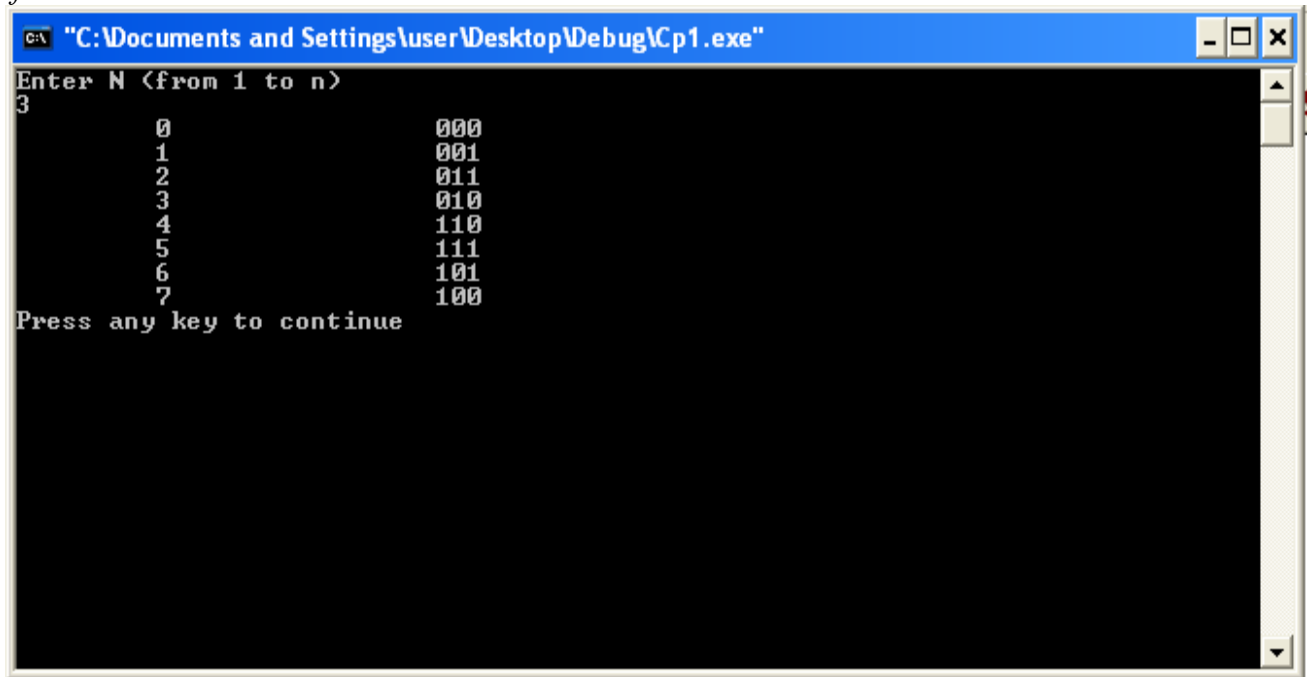
## Output

*for n=1*



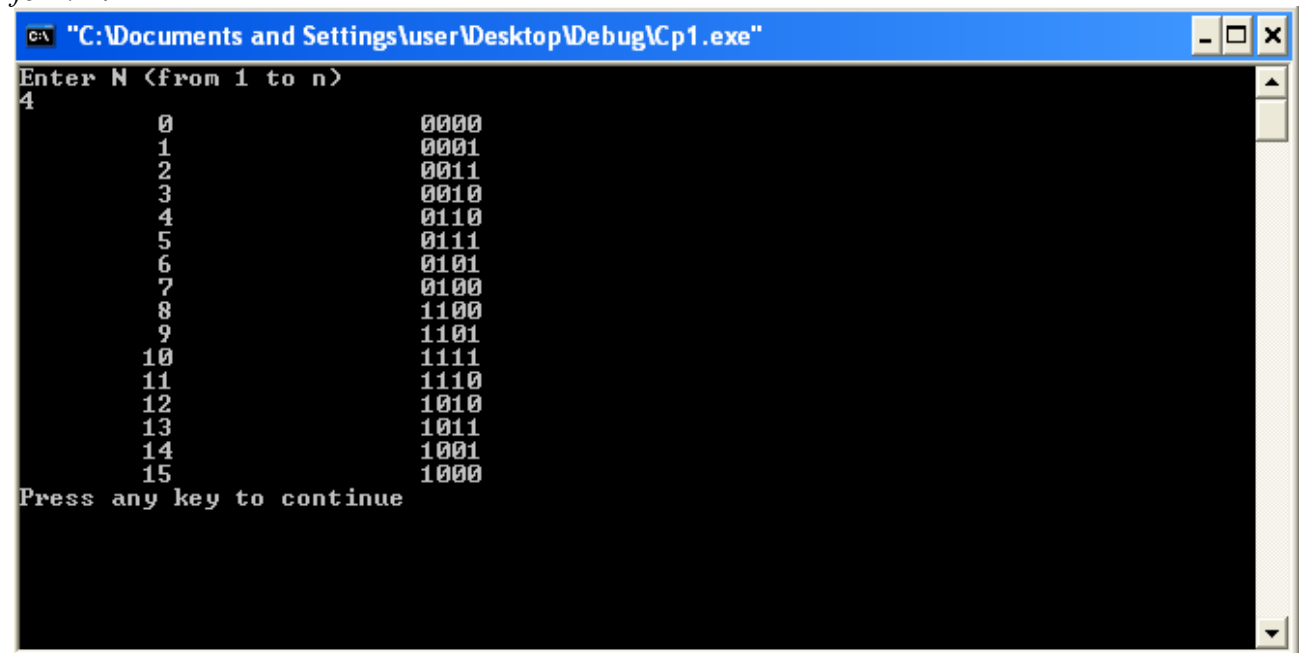
```
C:\Documents and Settings\user\Desktop\Debug\Cp1.exe
Enter N <from 1 to n>
1
0
1
Press any key to continue_
```

for  $n=3$



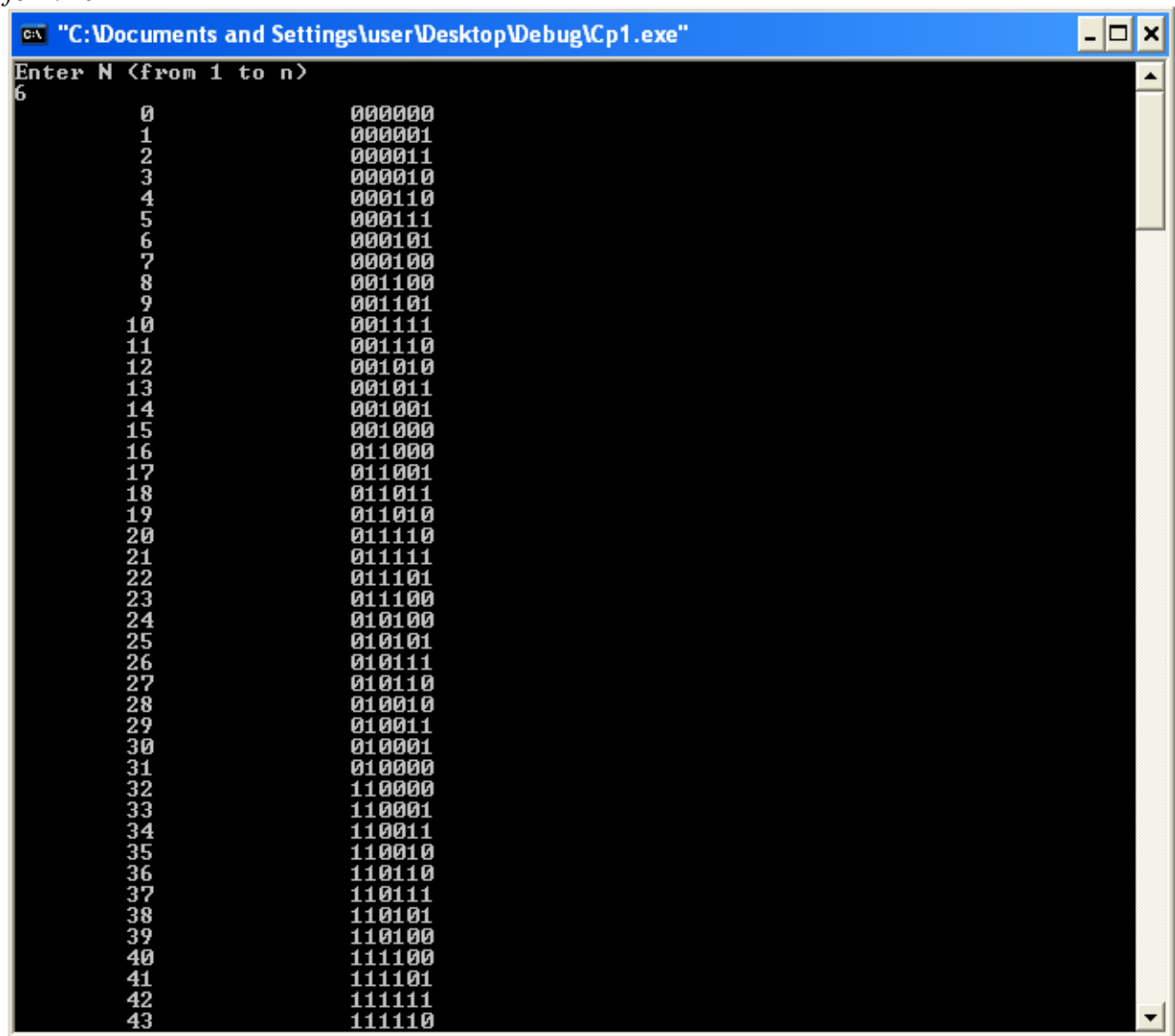
```
C:\Documents and Settings\user\Desktop\Debug\Cp1.exe
Enter N <from 1 to n>
3
    0      000
    1      001
    2      011
    3      010
    4      110
    5      111
    6      101
    7      100
Press any key to continue
```

for  $n=4$



```
C:\Documents and Settings\user\Desktop\Debug\Cp1.exe
Enter N <from 1 to n>
4
    0      0000
    1      0001
    2      0011
    3      0010
    4      0110
    5      0111
    6      0101
    7      0100
    8      1100
    9      1101
   10      1111
   11      1110
   12      1010
   13      1011
   14      1001
   15      1000
Press any key to continue
```

for  $n=6$



```
C:\Documents and Settings\user\Desktop\Debug\Cp1.exe
Enter N (from 1 to n)
6
0      000000
1      000001
2      000011
3      000010
4      000110
5      000111
6      000101
7      000100
8      001100
9      001101
10     001111
11     001110
12     001010
13     001011
14     001001
15     001000
16     011000
17     011001
18     011011
19     011010
20     011110
21     011111
22     011101
23     011100
24     010100
25     010101
26     010111
27     010110
28     010010
29     010011
30     010001
31     010000
32     110000
33     110001
34     110011
35     110010
36     110110
37     110111
38     110101
39     110100
40     111100
41     111101
42     111111
43     111110
```