

LAB 12/18

***Please use recursive function to solve the following problems.**

1. A robot can take 1meter for two directions: right side or down side. However, it can't walk in the same direction more than 3meters continuously or it will shut down. Write a recursive function to evaluate the number of ways the robot can arrive at the goal of the 2D map. Assume that the start position is located at (0,0), and let the user input the goal position. **You should let the user input continuously until CTRL+D or CTRL+Z.**

Ex:

1 1 → 2

3 3 → 14

1 0 → 1

0 3 → robot will shut down!

2. The Hamming sequence S is a sequence of distinct integers in ascending order defined as follows:

(1). $1 \in S$

(2). If $x \in S$, then $2x \in S$, $3x \in S$, and $5x \in S$

(3). Nothing else belongs to S

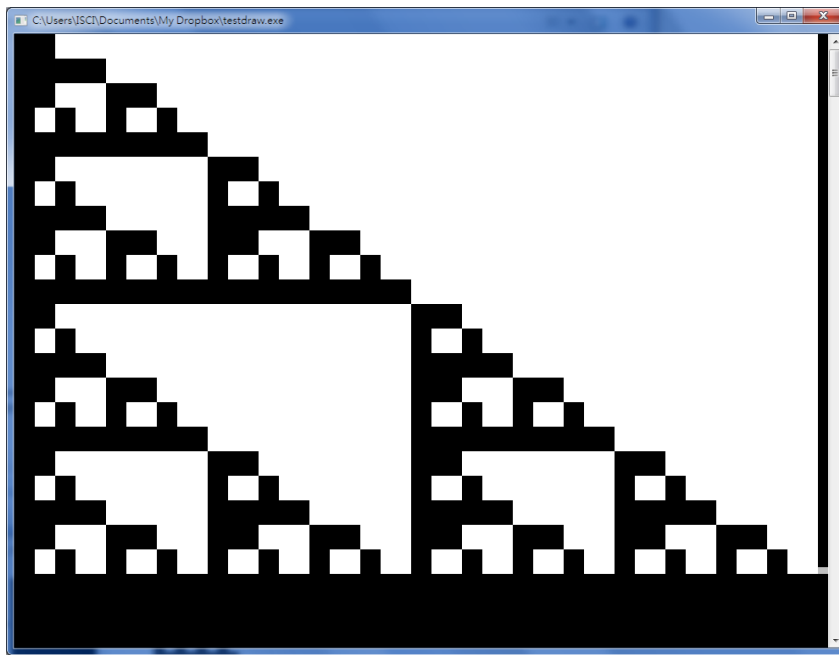
The first 20 elements of the Hamming sequence are

1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,25,27,30,32,36

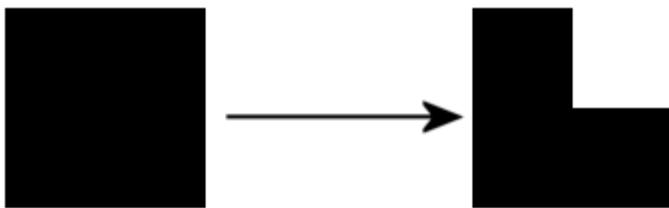
Write a recursive function "bool isHamming(int n)", which can determine whether n belongs to Hamming sequence. In this problem you are required to output the Kth number of Hamming sequence. Please let user input an integer K continuously until inputting 0 for stopping program.

Ex: 1 → 1 20 → 36 50 → 243

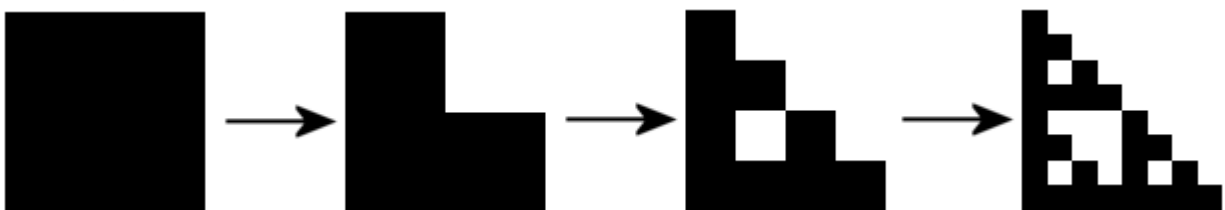
3. Write a recursive function `void drawSierpinski(int n, int x1, int y1, int x2, int y2)` to draw a Sierpinski Triangle.
i.e the result of a Sierpinski Traingle after 5 recursions should look like this:



The algorithm will convert every rectangle into an L-shape. For instance:



Since the L-shape itself consists out of 3 rectangles, which are again converted into an L-shape. For instance:



You only need to complete the following program:

```
#include <stdio.h>
#include <conio2.h>
```

```
#define MAX 5 // define the max recursions, try different numbers!
```

```
void drawRect(int x1, int y1, int x2, int y2)
```

```
{    textbackground(WHITE);
```

```
    int x,y;
```

```
    for (x=x1;x<=x2;x++)
```

```
        for (y=y1;y<=y2;y++)
```

```
        {
```

```
            gotoxy(x,y);
```

```
            printf(" ");
```

```
        }
```

```
}
```

```
void drawSierpinski(int n, int x1, int y1, int x2, int y2)
```

```
{    //draw the white rectangle
```

```
    drawRect((x1+x2)/2,y1,x2-1,(y1+y2)/2-1);
```

```
    if (n < MAX)
```

```
    {
```

```
        //Something missed here...
```

```
    }
```

```
}
```

```
int main()
```

```
{    textbackground(0);
```

```
    clrscr();
```

```
    drawSierpinski(1,1,1,80,24); // try different sizes!
```

```
    return 0;
```

```
}
```