Lab10

1. Homework Problem I

part1.

A robot can take steps of 1meter, 2 meters and 3 meters. Write a recursive function to evaluate the number of ways the robot can walk n meters. The program should be able to execute repeatedly until user entering Ctrl-D/Ctrl-Z.

part2.

Write a program that inputs two numbers: x and y (data are all integers) in the main program and passes these two numbers (pass by value) to the recursive function: power that returns the x^y.

If y≧0,

$$\text{power}(x, y) = \begin{cases} 1 & \text{if } y=0 \\ x & \text{if } y=1 \\ x * \text{power}(x, y-1) & \text{if } y>1 \end{cases}$$

If y<0,

$$\text{power}(x, y) = \frac{1}{\text{power}(x,-y)}$$

Print the result in the main program. The program should be able to execute repeatedly until user entering Ctrl-D/Ctrl-Z.

Sample Output:

```
====Part1====
n = 3
4 ways
n = 5
13 ways
n = 7
44 ways
n = ^Z


====Part2====
2 10
power(x, y)= 1024.000000
3 5
power(x, y)= 243.000000
^Z
```

2. Homework Problem II

part 1.

Please use recursive function to compute the following:

(a). $S = \frac{1}{1*2} + \frac{1}{2*3} + \frac{1}{3*4} + \cdots + \frac{1}{n*(n-1)}$

(Think about when n becomes to 2 what the value of this term)

(b). $\pi = 4 * \frac{2}{3} * \frac{4}{3} * \frac{4}{5} * \frac{6}{5} * \frac{6}{7} * \frac{8}{7} * \cdots * \frac{2*n}{2*n+1} * \frac{2*(n+1)}{2*n+1}$

(You may consider ($\frac{2}{3}$ and $\frac{4}{3}$), ($\frac{4}{5}$ and $\frac{6}{5}$), ($\frac{6}{7}$ and $\frac{8}{7}$), ⋯ as one pair and let n=0 as 4)

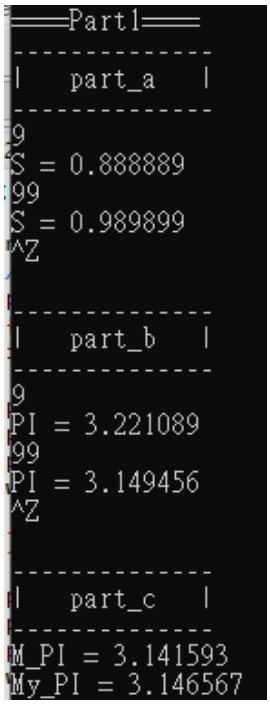
(c). From (b) and using the constant M_PI in the math.h to find the approximated value of π till the error between M_PI and your approximated value π reduces to 0.005. Please print both M_PI and your approximated value π .

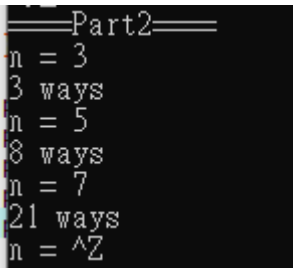
The program (a)&(b) should be able to execute repeatedly until user entering Ctrl-D/Ctrl-Z.

part2.

Suppose that we have a 2 × n rectangular board divided into 2n squares. Write a function that computes the number of ways to cover this board exactly by 1 × 2 dominoes. The program should be able to execute repeatedly until user entering Ctrl-D/Ctrl-Z.

Sample Output:





3.

part1.

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!

part2.

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

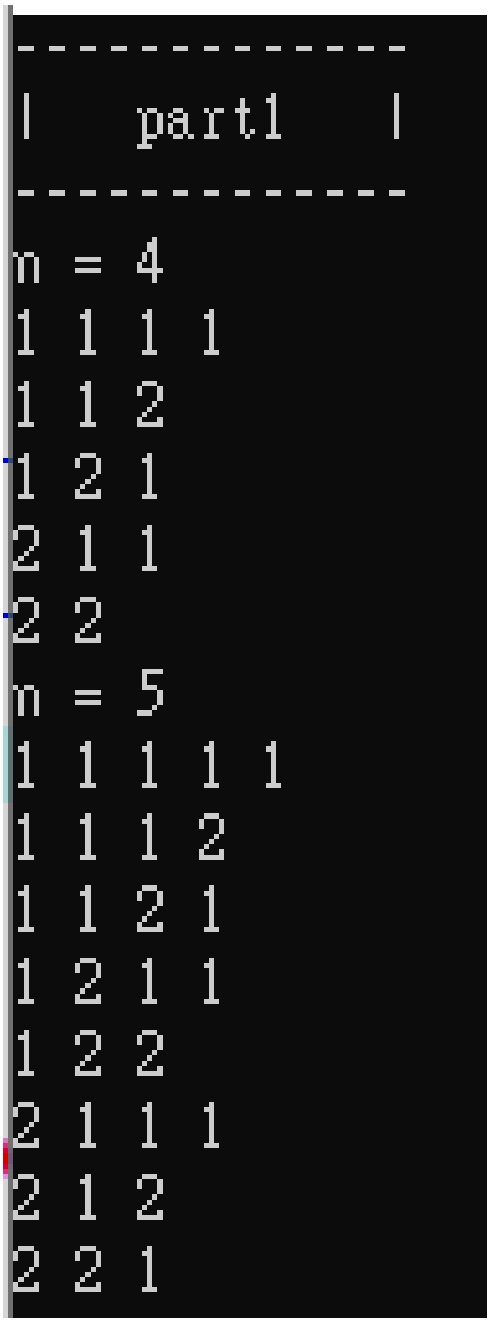
Sample Output:

```
part1
1 1
walk way : 2
3 3
walk way : 14
1 0
walk way : 1
0 3
robot will shut down!
^Z
part2
1
1
20
36
50
243
^Z
```

4.

part1

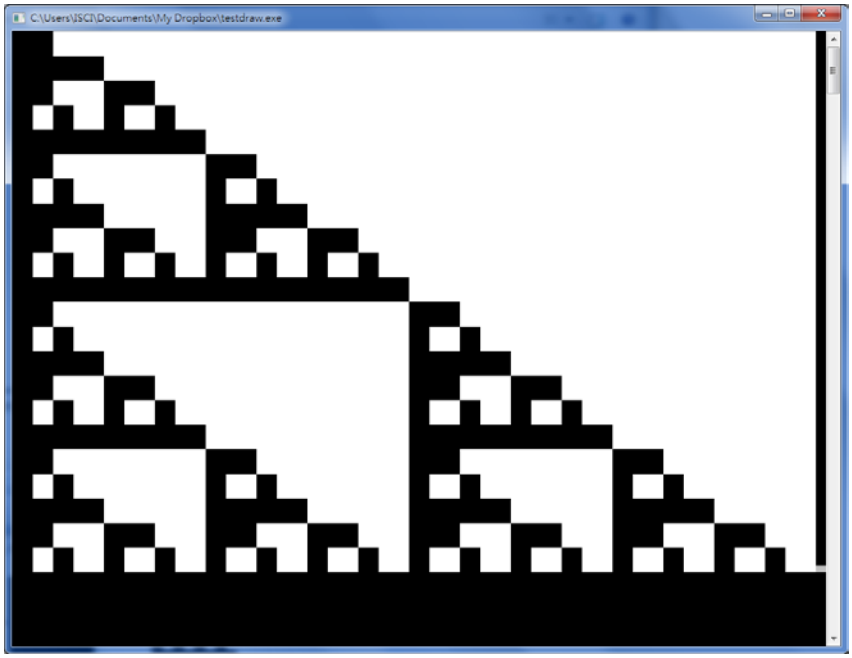
A robot can take steps of 1meter, 2 meters. Write a function that lists all of the ways that the robot can walk n meters. The program should be able to execute repeatedly until user entering Ctrl-D/Ctrl-Z.



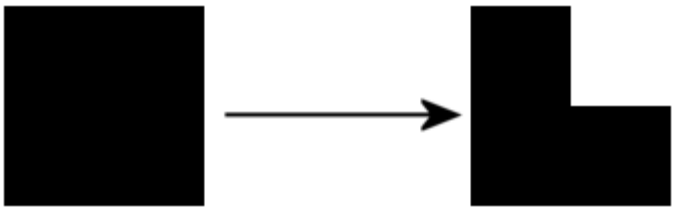
part2

Write a recursive function void drawSierpinski(int n, int x1, int y1, int x2, int y2) to draw a Sierpinski Triangle.

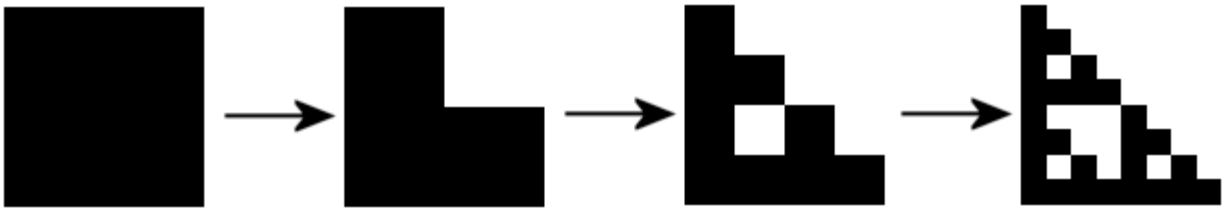
i.e the result of a Sierpinski Traingle after 5 recursions should look like this:



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(1,1,1,120,30);    // try different sizes!
        return 0;
}
```