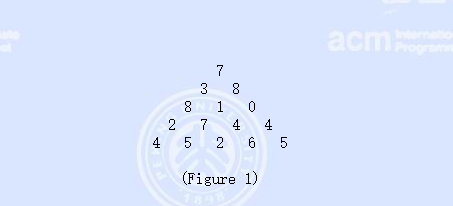
# 实验二：动态规划

**一、实验目的**

理解动态规划的基本思想，理解动态规划算法的两个基本要素最优子结构性质和子问题的重叠性质。熟练掌握典型的动态规划问题。掌握动态规划思想分析问题的一般方法，对较简单的问题能正确分析，设计出动态规划算法，并能快速编程实现。

**二、实验内容**



在上面的数字三角形中寻找一条从顶部到底边的路径，使得路径上所经过的数字之和最大。路径上的每一步都只能往左下或 右下走。只需要求出这个最大和即可，不必给出具体路径。 三角形的行数大于1小于等于100，数字为0~99

输入格式：

5 //表示三角形的行数 接下来输入三角形

7

3 8

8 1 0

2 7 4 4

4 5 2 6 5

要求输出最大和

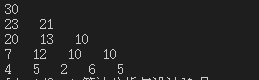
1. **动态规划基本思想**

从下往上计算，从倒数第二层开始，计算于倒数第一层的左右相邻的最大值，然后就把这个最大值加到这个数字本身上去，作为最优子结构为上一层服务。

1. **实验过程**

思路清晰，实验过程流程无问题。

1. **实验结果**



#include<stdio.h>

#include<string.h>

int max(int a,int b){

if(a >b ){

return a;

}else{

return b;

}

}

int main(){

int n;

int num[100][100];

scanf("%d",&n);

memset(num,0,sizeof(num));

for(int i=0;i<n;i++){

for(int j=0;j<=i;j++){

scanf("%d",&num[i][j]);

}

}

for(int i=n-2;i>=0;i--){

for(int j=0 ; j<=i; j++){

num[i][j] += max(num[i+1][j] , num[i+1][j+1]);

}

}

}