**《算法设计与分析》实验报告**

# 实验一 递归与分治策略

|  |  |  |  |
| --- | --- | --- | --- |
| **报告书** | | | |
| 姓名 | 吴宇敖 | 指导教师 |  |
| 学号 | 176001752 | 日 期 | 2020.3.15 |
| 班级 | 17计算机软件一班 |  |  |
| **实验内容** | | | |
| 标有OJ的题，请先在OJ上提交   1. 在指定时间完成OJ的练习：此处仅需粘贴源码 2. 未及时完成的OJ练习：此处粘贴源码+错误里类型(如PE)+问题说明及修改通过后源码 | | | |
| **实验目的** | | | |
| 堂练目的：考察学生对本阶段知识点及编程掌握的熟练程度 | | | |
| **实验过程和步骤** | | | |
| **【实验题目】**  有排成一行的n个方格，用红、粉、绿三色涂每个格子，要求：  （1）任何相邻的方格不能同色；  （2）且首尾两格也不同色。  求n个格子满足要求的涂法数。  输入要求：  输入多个整数n，每个整数表示有多少个方格，n<=60  输出要求：  输出多个整数，一行一个，表示每个对应输入的涂法数  **【错误类型】**  WA两次。  **【错误原因分析】**  第一次错误：  我是以为要进行全部输入后在进行全部输出，之后随意尝试了一下，就出现WA错误。  第二次错误：  是定义变量使使用的是int类型，导致最后一些数，超过了int类型的范围  **【正确程序代码及结果】**  #include<iostream>  using namespace std;  int main()  {  long long num[61];//定义最多一排60个方格  num[1]=3;  num[2]=6;  num[3]=6;  int n;  for(int j=4;j<61;j++)//使用递推的方法来进行计算  {  num[j]=num[j-1]+num[j-2]\*2;//第N个方格是由第N-1个方格的涂法数加上第N-2个方格的涂法数的两倍  }  while(cin >> n)  cout << num[n] << endl;  //system("pause");  return 0;  }    **【实验题目】**  王老师爬楼梯，他可以每次走1级或者2级或者3级楼梯，输入楼梯的级数，求不同的走法数。（要求递推求解）如果N很大，需要高精度计算。  输入要求：  一个整数N，N<=1000。  输出要求：  共有多少种走法  **【错误类型】**  无  **【错误原因分析】**  在VS上编写代码时，使用递推写法，但是对于题目的分析不准确，导致出现最后结果输出的错误，最后经过仔细分析解决了问题。  **【正确程序代码及结果】**  #include<iostream>  using namespace std;  int num[1001][1000];  int main()  {  int n;  for(int i=0;i<1001;i++)//初始化ˉ  for(int j=0;j<1000;j++)  num[i][j]=0;  num[1][1]=1;  num[2][1]=2;  num[3][1]=4;  int k=1;//表示有几位，方便后面进行输出  cin>>n;//输入有几个楼梯  for(int i=4;i<=n;i++){  for(int j=1;j<=k;j++)  num[i][j]=num[i-1][j]+num[i-2][j]+num[i-3][j];//计算下一台阶的某一位  for(int j=1;j<=k;j++){  num[i][j+1]+=num[i][j]/10;//进位  if(num[i][j]>=10&&j==k)//增加一位  k++;  num[i][j]%=10;  }  }  for(int i=k;i>=1;i--)//输出  {  cout<<num[n][i];  }  cout<<endl;  //system("pause");  return 0;  }      **【实验题目】**    输入要求：  输入一个整数k，k<=5；  输入特殊格子的坐标x，y。  输出要求：  输出一个由数值表示的二维矩阵。填充规则如下：  （1）用数值填充方格；  （2）特殊方格数值为0；  （3）从中心点开始；然后左上、右上、左下、右下的计数顺序填数；同一块用相同数值表示；  （4）每个数值占4个位置空间；右对齐，左补空格。  **【错误类型】**无  **【错误原因分析】**  在VS上运行时，在find函数中没有定义一个新的变量res1，导致我的全局变量在同一个函数中进行不断的改变。之后定义了一个res1，使每个函数中都有它自己的res1。  **【正确程序代码及结果】**  #include<iostream>  using namespace std;  int num[33][33];//定义一个棋盘  int res=1;  void find(int x1,int y1,int x2,int y2,int size)  {  if(size==1)  return;  int res1=res++;  int m=size/2;  if(x2<x1+m&&y2<y1+m)//当特殊点在左上时  find(x1,y1,x2,y2,m);  else  {  num[x1+m-1][y1+m-1]=res1;  find(x1,y1,x1+m-1,y1+m-1,m);  }  if(x2<x1+m&&y2>=y1+m)//当特殊点在右上  find(x1,y1+m,x2,y2,m);  else  {  num[x1+m-1][y1+m]=res1;  find(x1,y1+m,x1+m-1,y1+m,m);  }  if(x2>=x1+m&&y2<y1+m)//当特殊点在左下  find(x1+m,y1,x2,y2,m);  else  {  num[x1+m][y1+m-1]=res1;  find(x1+m,y1,x1+m,y1+m-1,m);  }  if(x2>=x1+m&&y2>=y1+m)//当特殊点在右下  find(x1+m,y1+m,x2,y2,m);  else  {  num[x1+m][y1+m]=res1;  find(x1+m,y1+m,x1+m,y1+m,m);  }  }  int main()  {  int n,x,y;  int size =1;  cin>>n>>x>>y;  for(int i=0;i<n;i++)  size\*=2;  //num[x][y]=0;  find(0,0,x,y,size);  for(int i=0;i<size;i++)  {  for(int j=0;j<size;j++)  printf("%4d",num[i][j]);  cout<<endl;  }  //system("pause");  return 0;  } | | | |