※解一道题，设计一个程序都要有step by step分治的解题思想

Top down

直接解决一个大问题有时候会让人觉得很困难，难以下手，这时候就要发扬走一步再走一步的精神。困难的问题其实是简单的问题的迭代

下面是两个解MOOC里题目的感悟：

**//问题的分解，step by step；关键是让为题顺理成章、行云流水的完成**

**//框架（数据的输入(需要的数据类型)，数据的处理，数据的输出）**

**//本题的关键点：**

**在算法竞赛中，对数据处理基本是唯一的关键点（也要考虑输入对数据处理的影响）**

**这就要求我们对困难的数据处理进行再分解！在一次次分解中，问题会变得简单**

**在大工程里，也是要多次分解问题，在大的框架下逐步细化问题，然后再越来越小，越来越简单的小问题中解决问题。**

**当解题解到看到题目就能很快进行问题分解到各项小问题，然后轻松求解时，那就真正掌握题目了。**

**所以做题目不知要做一次，要做到上述的程度，做一道题目约等于做了很多同类型的题目！**

**/\***

**题目内容：**

**我们认为2是第一个素数，3是第二个素数，5是第三个素数，依次类推。**

**现在，给定两个整数n和m，0<n<=m<=200，你的程序要计算第n个素数到第m个素数之间所有的素数的和，包括第n个素数和第m个素数。**

**输入格式:**

**两个整数，第一个表示n，第二个表示m。**

**输出格式：**

**一个整数，表示第n个素数到第m个素数之间所有的素数的和，包括第n个素数和第m个素数。**

**\*/**

**#include<stdio.h>**

**#include<stdbool.h>**

**#include<math.h>**

**#define N 200**

**bool is\_Prime(int number);**

**//问题的分解，step by step；关键是让为题顺理成章、行云流水的完成**

**//框架（数据的输入(需要的数据类型)，数据的处理，数据的输出）**

**//关键点：一、处理数据：素数(需要is\_Prime()) ,素数的标号对应，求和**

**int main(void)**

**{**

**int min, max;**

**scanf("%d %d", &min, &max);**

**int prime[N] = {[1] = 2};//2 3 5 7 11 13 17 19 77**

**int count = 2;**

**for(int i = 3; i < N; i++)**

**{**

**if(is\_Prime(i))**

**{**

**prime[count++] = i;**

**}**

**}**

**int sum = 0;**

**for(int i = min; i <= max; i++)**

**{**

**sum += prime[i];**

**}**

**printf("%d", sum);**

**return 0;**

**}**

**bool is\_Prime(int number)**

**{**

**for(int i = 2; i <= (int)sqrt(number); i++)**

**{**

**if( number % i == 0)**

**{**

**return false;**

**}**

**}**

**return true;**

**}**

**//题目内容：**

**//对数字求特征值是常用的编码算法，奇偶特征是一种简单的特征值。对于一个整数，从个位开始对每一位数字编号，个位是1号，十位是2号，以此类推。这个整数在第n位上的数字记作x，如果x和n的奇偶性相同，则记下一个1，否则记下一个0。按照整数的顺序把对应位的表示奇偶性的0和1都记录下来，就形成了一个二进制数字。比如，对于342315，这个二进制数字就是001101。**

**//你的程序要读入一个非负整数，整数的范围是[0,1000000]，然后按照上述算法计算出表示奇偶性的那个二进制数字，输出它对应的十进制值。**

**//**

**//提示：将整数从右向左分解，数位每次加1，而二进制值每次乘2。**

**//**

**//输入格式:**

**//一个非负整数，整数的范围是[0,1000000]。**

**//**

**//输出格式：**

**//一个整数，表示计算结果。**

**#include <stdio.h>**

**int power (int x, int n);**

**int main (void)**

**{**

**unsigned long t;**

**int n = 1; // 用来记位数**

**int x ; //用来取位**

**int sum =0 ;**

**int a[7];**

**int i;**

**scanf("%lu",&t);**

**while (t != 0) //出口 1、与数制有关的操作建议**

**{**

**x = t % 10;**

**if (x % 2 == n % 2) //2、用 == 对比通过相同计算的结果是否相同**

**{**

**a[n-1]=1;//sum = sum\*2+1;**

**}**

**else**

**{**

**a[n-1]=0;//sum = sum\*2; 问题在于当二进制数头几位都为零时，不该进位**

**}**

**n++;**

**t = t/10;**

**}**

**for (i=0; i<n-1; i++ ) // 3、想想为什么是n—1 ，因为从for循环里出来多了一！！！**

**{**

**sum += power(2,i)\*a[i];**

**}**

**printf("%d",sum);**

**return 0;**

**}**

**int power (int x, int n)**

**{**

**int sum=1;**

**int i=1; // 5、对于累加累乘的值一定要初始化！！**

**if(n==0)**

**sum = 1; //4、一个函数的定义 1、阶乘、指数、斐波那契 首几项要弄好 （即首项定义） 2、递推定义**

**for (i=1; i<=n;i++)**

**{**

**sum \*= x;**

**}**

**return sum;**

**}**

**/\* t n x sum**

**120 1 0 1**

**12 2 2 3**

**1 3 3 8**

**试考虑临界状况**

**\*/**