苏州大学实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学院 | 计算机科学与技术学院 | 专业 | 计算机科学与技术 |
| 学号 | 2427405039 | 姓名 | 刘志轩 |
| 课程名称 | 面向对象与C++程序设计 | 指导老师 | 陈伟 |
| **综合实验一** | | | |
| **一．实验目的**   1. 掌握分支、循环、函数、向量等使用方法。 2. 掌握通过多个步骤，完成较复杂的处理过程。   **二．实验内容**  给定两个(10, 10000)开区间的正整数m与n，其中m小于n：   1. 编写函数func1，以向量形式返回闭区间[m, n]范围内各位数字依次递增（从左至右、相邻数字后一位比前一位刚好大1）的所有正整数，如数字23、345、1234等。 2. 编写函数func2，以第1题返回的向量作为输入；依次统计向量中每个数字素因子的个数，并将个数存入新的向量，函数最终返回这个新向量。举例：数字34的所有因子为1、2、17、34，其中2和17是素数，即有两个素因子。因此，需要将与34对应的个数2存入新向量。 3. 编写函数func3，以第1题、第2题返回的向量依次作为func3的两个参数。假设这两个参数是vec1与vec2，根据下标依次遍历两个向量中的元素vec1[i]与vec2[i]，并使用如下规则，从0开始根据vec2[i]的值向vec1[i]跳跃前进：    1. 假设当前vec1[i]的值为a, vec2[i]的值为b;    2. 从0开始，沿X轴第一次前进b并退1，第二次前进b并退2，第三次前进b并退3，以此类推，直至前进b并退b；接下来继续前进b并退1，直至前进b并退b；如此循环跳跃；   请问：需要几次跳跃可以首次到达或跃过vec1[i] (此时跳跃直接结束，不需要退回)，将该结果存入一个新向量，且函数最终返回这个向量。注释：如果vec2[i]的值为1，根据上述规则是无法前进的，此时直接将-1存入新向量即可。  **答题提示：**   1. 某些需要复用的功能，可以单独编写函数供其他函数调用，如：判断某个数字是否为素数； 2. 假设输入的m=100，n=500，其结果如下图所示：     **三、其他说明**   1. 在给定的sample.cpp的基础上完善程序; 2. 程序主体部分不能修改。   **四、实验程序（含必要的注释）及测试结果**  #include<iostream>  using namespace std;  #include<vector>  vector<int> func1(int m, int n) //每一位分别计算，降低时间复杂度  {      vector<int> vec;      for (int len=2;len<5;len++){          for (int s=1;s<=10-len;s++){              int num=0;              int d=s;              for (int i=0;i<len;i++){                  num=num\*10+d;                  d+=1;}                  if (num>=m&&num<=n){                      vec.push\_back(num);                  }          }      }      return vec;  }  vector<int> func2(vector<int> vec)   //试除法&开方降低时间复杂度  {      vector<int> res;      for (unsigned int i = 0; i < vec.size(); i++){          int ans=0;int temp=vec[i];          for (int i=2;i\*i<=temp;i++){              if (temp%i==0){                  ans+=1;                  while (temp%i==0){//除去所有因子                      temp/=i;                  }              }          }          if (temp!=1) ans++;//仍有质因子          res.push\_back(ans);      }      return res;  }  vector<int> func3(vector<int> vec1, vector<int> vec2)  {      vector<int> res;      for (int i=0;i<vec1.size();i++){          int a=vec1[i],b=vec2[i],left=0,time=0,step=1;          if (b==1){//无法跳跃              res.push\_back(-1);              continue;          }          while (left<a){              time++;              left=left+b-step;              step++;              if (step>b){                  step=1;              }              if (left+b>=a){//只跳跃不回头                  time++;                  break;              }          }          res.push\_back(time);      }      return res;  }  int main()  {      int m = 100, n = 500;      //func1的调用      vector<int> v1 = func1(m, n);        //func2的调用      vector<int> v2 = func2(v1);      //func3的调用      vector<int> v3 = func3(v1, v2);      for (unsigned int i = 0; i < v3.size(); i++)          cout << v1[i] << " " << v2[i] << " " << v3[i] << " " << endl;  }    **五、实验总结**  在本次《面向对象与 C++ 程序设计》综合实验一中，我围绕复杂数字处理任务展开编程实践，收获颇丰。  实验要求在给定区间生成特定规律数字，统计其素因子个数并模拟跳跃操作。这使我深度运用 C++ 的分支、循环、函数和向量知识。在func1函数中，多层循环构建数字并存储到向量，提升了我对循环和向量操作的熟练度。func2和func3里分支结构的运用，增强了我处理不同条件判断的能力。  复杂的实验任务让我学会将大问题拆解为小问题，分别编写函数解决，提升了复杂问题处理能力。在函数编写上，func1控制数字生成与筛选，func2用试除法统计素因子，func3模拟跳跃过程并处理特殊情况。  实验中也遇到不少问题。func2起初素因子统计出错，经查阅资料分析，修正了逻辑。func3的跳跃逻辑边界条件处理不当，通过打印中间变量排查并解决了问题。  这次实验全面提升了我的编程技能，让我对 C++ 掌握更扎实，学会优化代码，调试能力也显著增强。同时，思维方式从单纯编写代码转变为系统解决问题。 | | | |