

算法与数据结构设计报告

（ 2023 / 2024学年 第 一 学期）

题 目：题目A1：Top-N问题

题目B1：代码分析器

|  |  |
| --- | --- |
| **专 业** | **计算机科学与技术** |
| **学 生 姓 名** | **周才凯** |
| **班 级 学 号** | **B21030129** |
| **指 导 教 师** | **陈小柏** |
| **指 导 单 位** | **计算机科学与技术系** |
| **日 期** | **2023.10.30-2023.11.12** |

|  |  |  |
| --- | --- | --- |
| **支撑指标点** | **评价准则** | **计分（每项10分）** |
| 课程目标1：文献调研与资料收集能力，问题发现、研究、分析与解决能力（20分） | 1、能够掌握算法与数据结构设计的相关基础知识，并能够针对求解的工程问题，收集资料进行合理的分析与设计 |  |
| 2、通过调研，能够选择合适的程序设计语言与编程开发平台，对求解的工程问题进行编程实现 |  |
| 课程目标2：通过课程设计，培养学生综合应用算法和数据结构等知识解决工程问题的实践能力（20分） | 3、能够给出数据结构和算法的设计描述，给出关键算法的流程图或伪代码，并给出各算法之间的结构关系描述 |  |
| 4、具备一定的人机交互设计意识，人机交互设计合理、友好，操作简便 |  |
| 课程目标3：培养解决工程问题的开发工具运用能力，能够利用程序设计软件或系统对问题求解进行模拟和实现，能够设计测试数据验证问题解决方法的正确性，并能够对问题解决方法的性能和效率进行分析（40分） | 5、具备一定的算法与数据结构设计分析能力，能够完成课题要求的各项任务和指标 |  |
| 6、能够结合计算机软硬件资源，合理选用算法、数据结构、数据存储方式等技术手段，对求解的工程问题进行有效建模和求解 |  |
| 7、具备一定自学能力与探索创新意识，能够充分利用教科书及其资源（如网络等）自学新知识与新技能 |  |
| 8、掌握调试方法与工具，对程序开发过程中出现的问题进行分析、跟踪与调试，并能够进行充分测试 |  |
| 课程目标4：选择同类课题的学生能够通过讨论和交流解决课程设计中的难题，能在实验报告中准确阐述课程设计的内容，能够清晰陈述观点和回答问题（20分） | 9、能够正确、完整地回答指导教师关于课题的问询，反映其对课题内容，以及相关的工程基础知识具有较好的理解和掌握 |  |
| 10、具备一定的语言表达能力与文字处理能力，能够结合复杂工程问题撰写报告，报告内容和实验数据详实，格式规范 |  |
| 算法与数据结构设计能力测评总分 | |  |
| **指导教师： 年 月 日** | | |
| **备注：** | | |

**具体课题题目**

**（题目格式：宋体，三号，加粗，居中对齐）**

**一、课题内容和要求（标题格式：宋体，四号，加粗，两端对齐，顶格不缩进，下同）**

本课题目标系统“top\_n问题”的功能框架图如图1所示。

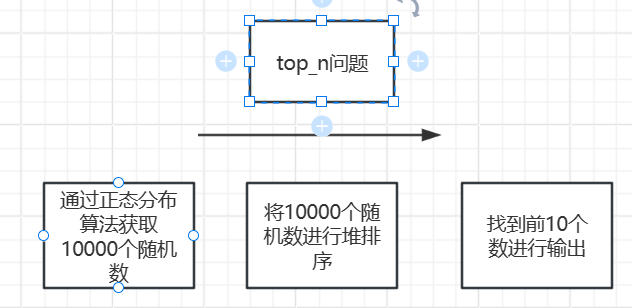


图1 功能框架图

（1）支持对物理竞赛的前十名结果进行模拟研究。

（2）利用正态分布算法实现对物理竞赛的模拟；

（3）支持界面展示；

（4）支持对不同类竞赛的效果评估。

本课题目标系统“代码分析器”的功能框架图如图2所示。

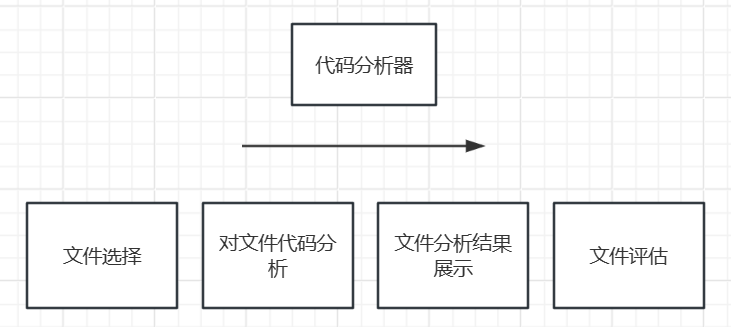


图2 功能框架图

（1）支持对包含代码的文件进行选择。

（2）利用排除法算法实现对函数的判断；

（3）支持界面展示；

（4）支持对单个或者多个代码文件进行分析评估。

**二、数据结构说明**

//函数属性结构

typedef struct {

int length; //每一个函数的长度

int pos; //每一个函数的位置

}Fun;

//统计结构的声明

typedef struct {

int comments; //纯注释的个数

int comment; //混合注释个数

int blank; //空行的个数

Fun fun[MAXSIZE]; //函数的属性

int others; //除去函数中代码外其余的代码个数

int funcount; //函数的个数

}Analy;

**三、算法设计**

在此说明每个部分的算法设计说明（可以是描述算法的流程图），每个程序中使用的存储结构设计说明（如果指定存储结构请写出该存储结构的定义，如果用面向对象的方法，应该给出类中成员变量和成员函数原型声明）。例如：

1堆排序算法 （二级标题格式：宋体，小四号，不加粗，两端对齐，不缩进，下同）

基于堆排序思想，将数列通过推排序变成最大堆，再将堆顶和堆尾交换，再求n-1的最大堆。

2正态分布随机数算法

int Gaussian\_noise(double mean, double stddev)

{

unsigned seed = std::chrono::system\_clock::now().time\_since\_epoch().count();

std::default\_random\_engine generator(seed);

std::normal\_distribution<double> dist(mean, stddev);

return dist(generator);

}

3 函数判断算法

排除法：有分号,if while for的不是函数开头，没有小括号的不是函数开头，在对开头进行排除后，再将int和int后面的空格去掉做判断，如果发现左括号在空格前面，一定不是，如果函数名为空，函数名以括号开头，函数的括号出现在int中，一定不是，后续测试如果继续发现其他情况，将再次写入程序，进行排除。

4注释判断算法

先去空白，找//的位置，在第一个为纯注释，不在第一个为混合注释，没有就不为注释。

**四、详细设计**

//题目A1：Top - N问题

//假设全国有10000人参加物理竞赛，只录取成绩优异的前10名，并将他们从高分到低分输出。(满分100分) 人和分数应该对应，就是说找出前10还包括人名，但是找出前十的分数也可以对应到人名，但是随机数的结果不符合物理竞赛的实际结果，需要改进算法

//而对落选的其他考生，无需排出名次，问此种情况下，试编写尽可能快的算法解决Top - 10问题？

//扩展：如何在10亿数中找出前1000大的数？

#pragma warning(disable:4996) //解决返回值被忽略问题 scanf

#include <iostream> //符号 # 预处理指令 include 文件名 iostream 尖括号 <>

//#include <algorithm> //算法

#include <random>

#include <chrono>

#include"topn.h"

using namespace std;

//开头 using namespace std就是使用一个名字为std的namespace。

//这样命名空间 std 内定义的所有标识符都有效（曝光）

int nums[10];

int main()

{

srand(time(0)); //time(0)是得到当前时间值

int len = sizeof(nums) / sizeof(int);//sizeof(a)/sizeof(int)含义

//sizeof是求字节数的函数，计算小括号里的变量占内存多少单元，计算单位是字节数。

//a如果是int，float，long就是4个字节，short就是2个字节，char就是1个字节，double就是8个字节。

//sizeof（int）就是求int型变量的字节数，也就是等于4。sizeof的优先级比除号 / 的高。

//本段代码的功能是获取数组元素的个数

//printf("%d ", len); //测试代码

for (int i = 0; i < len; ++i)

{

nums[i] = Gaussian\_noise(50, 10) % 100;//防止出现100+的数，但是负数存在出现可能，需要限制11.7 14：21

//nums[i] = rand() % 100; //编程语言中rand()表示产生一个随机整数，%100将随机数限制在[0,100)，应该改进为建立一个gaussrand( )函数，生成正态分布的数据

}

//获取随机数(0，100]，len的长度

for (int i = 0; i < len; ++i)

{

printf("%d ", nums[i]);

}

printf("\n");

int k = 0;

printf("\n请输入需要获取成绩的前n名的个数："); //需要将随机数改为正态分布

scanf("%d", &k);

sort(nums, len);//排序，从小到大

for (int i = len - 1; i > len - 1 - k; --i)

{

printf("%d ", nums[i]);//输出最后的k个数

}

return 0;

}

#include"topn.h"

using namespace std;

//开头 using namespace std就是使用一个名字为std的namespace。

//这样命名空间 std 内定义的所有标识符都有效（曝光）

int Gaussian\_noise(double mean, double stddev)

{

unsigned seed = std::chrono::system\_clock::now().time\_since\_epoch().count();

std::default\_random\_engine generator(seed);

std::normal\_distribution<double> dist(mean, stddev);

return dist(generator);

}

void swap(int nums[], int i, int j) //用于交换num数组中i项和j项的值

{

int temp = nums[i];

nums[i] = nums[j];

nums[j] = temp;

}

void heapify(int nums[], int i, int len)

{

int left = 2 \* i + 1;

int right = 2 \* i + 2;

int newIndex = i;//i = len / 2 - 1

if (left<len && nums[left]>nums[newIndex])

{

newIndex = left;//左子节点大，父节点下标等于左子节点

}

if (right<len && nums[right]>nums[newIndex])

{

newIndex = right;

}

if (newIndex != i) //下标发生变换，交换num数组中i项和newIndex项的值

{

swap(nums, i, newIndex);

heapify(nums, newIndex, len);//继续判断新子节点和其子节点之间的大小关系，直到建立i节点的最大堆

}

}

void buildMaxHeap(int nums[], int len)

{

for (int i = len / 2 - 1; i >= 0; --i) //对第一个非叶子节点及之前的进行堆化

{

heapify(nums, i, len);

}

}

void sort(int nums[], int len)

{

buildMaxHeap(nums, len);//建立最大堆

for (int i = len - 1; i > 0; --i)

{

//swap(nums, 0, len); //感觉是len-1，但是不想改

swap(nums, 0, len - 1);//果然是len-1

heapify(nums, 0, --len);//堆排序

//最大堆：对堆树中任意子树满足：父节点大于等于任一子节点；堆顶元素是堆中最大元素； 最小堆：对堆树中任意子树满足：父节点小于等于任一子节点；堆顶元素是堆中最小元素。

}//将堆顶和末尾交换，对n-1再求最大堆，再交换，直到数组由小到大排列

}

#pragma warning(disable:4996) //解决返回值被忽略问题 scanf

#include <iostream> //符号 # 预处理指令 include 文件名 iostream 尖括号 <>

//#include <algorithm> //算法

#include <random>

#include <chrono>

using namespace std;

//开头 using namespace std就是使用一个名字为std的namespace。

//这样命名空间 std 内定义的所有标识符都有效（曝光）

#pragma once

int Gaussian\_noise(double mean, double stddev);//用于生成正态分布随机数

void swap(int nums[], int i, int j); //用于交换num数组中i项和j项的值

void heapify(int nums[], int i, int len);//堆排序

void buildMaxHeap(int nums[], int len);//形成最大堆

void sort(int nums[], int len);//排序

//题目B1：代码分析器

//（一）课题内容

//读入C源文件，统计文件中的代码、注释和空行的行数以及函数的个数和平均行数，并利用统计信息分析评价该程序的风格。

//（二）课题要求

//(1) 将程序文件按字符顺序读入源程序；

//(2) 边读入程序，边识别统计代码行、注释行和空行，同时还要识别函数的开始和结束，以便统计其个数和平均行数。

//(3) 程序的风格评价分为代码（函数平均长度）、注释（占总行数比例）和空行（占总函数比例）三个方面。每个方面分为A，B，C:和D 四个等级。等级的划分标准如下：

//(4) 测试时先对较小程序进行分析。当你的程序能正确运行时，对你的程序本身进行分析。

//【其他要求】

//(1) 变量、函数命名符合规范。

//(2) 注释详细：每个变量都要求有注释说明用途；函数有注释说明功能，对参数、返回值也要以注释的形式说明用途；关键的语句段要求有注释解释。

//(3) 程序的层次清晰，可读性强。

#pragma warning(disable:4996) //解决返回值被忽略问题 scanf

#include<iostream>

#include<stdio.h>

#include"analy.h"

using namespace std;

#define COUNT 20 // 可以统计的最大的文件个数

#define LEN 20 // 文件名的最大长度

int main()

{

int i=0;

int c;

char filename[COUNT][LEN];

printf("\t-------------------------------------------------------------\n");

printf("\t 代码分析器 \n");

printf("\t-------------------------------------------------------------\n");

while (1) {

printf("\t\t\t\*\*\*\*\*\*\*\*\*\*程序分析\*\*\*\*\*\*\*\*\*\n");

printf("请输入您要分析的源文件: ");

scanf("%s", filename[i]);

fflush(stdin);//使用“fflush（stdin）”清除某些编译器中的输入缓冲区

analy(filename);

cout << "\t----------------------------------------------------------" << endl;

cout << "\t 1.分析其他代码 2.退出 " << endl;

cout << "\t----------------------------------------------------------" << endl;

cout << "请输入序号 ： ";

cin >> c;

if (c == 1) {

system("cls"); //清屏

getchar(); //清除缓存

main();

}

return 0;

}

}

#include"analy.h"

using namespace std;

//串的基本操作ADT

//检测是否为空串

BOOL StrEmpty(char\* s)

{

if (s[0] == '\0')

return TRUE;

return FALSE;

}

//查看S1中是否有值为S2的子串,若有则返回第一个子串的位置，若无则返回-1；

int Find(char\* s1, const char\* s2)

{

int i = 0, j = 0;

if (strlen(s1) < strlen(s2))

return -1;

while (s1[i] != '\0')

{

if (s1[i] == s2[j])

{

i++;

j++;

if (s2[j] == '\0')

return i - j;

continue;

}

i++;

j = 0;

}

return -1;

}

//读取文件中的一行字符

void HaveLine(FILE\* fp, char\* s)

{

while (!feof(fp))

{

\*s = fgetc(fp);

if (\*s == '\n') { //若是一行的结尾则表示取完了一行

\*s = '\0';

return;

}

s++;

}

\*s = '\0';

}

//忽略一行字符开头的空格和tab，返回截断后上的串指针

char\* IgnoreB(char\* s)

{

while (\*s == ' ' || \*s == ' ')

s++;

return s;

}

//判断一行字符是不是注释

int IsCom(char\* s)

{

int posc, pos1, pos2;

s = IgnoreB(s);//去空白

posc = Find(s, "//");

if (posc == 0)//此行仅有注释，无代码；

return 1;

if (posc == -1)

return 0;

pos1 = Find(s, "\"");

pos2 = Find(&s[pos1 + 1], "\"");

if (posc > pos1 && posc < pos2)

return 0;

return 2;

}

//判断一行字符是不是空白

BOOL IsBlank(char\* s)

{

s = IgnoreB(s);

if (\*s == '\0')

return TRUE;

return FALSE;

}

int IsFunB(char\* s)//判断函数

{

int i, j, pos, pos2;

//有分号,if while for的不是函数开头

if (Find(s, ";") != -1 || Find(s, "if") != -1 || Find(s, "for") != -1 || Find(s, "while") != -1 || Find(s, "switch") != -1)

return FALSE;

//没有小括号的不是函数开头

if ((pos = Find(s, "(")) == -1)

return FALSE;

i = Find(s, " ");//开头空格

s = IgnoreB(s);

j = Find(s, " ");//int add间空格

if (i != -1 && j != -1)//都有空格，将i变成int add间空格下标

i = i > j ? j : i;

else if (i == -1 && j == -1)//没空格

return FALSE;

else if (i == -1)//开头没有空格

i = j;

if (i > pos)//如果空格在左括号后面，一定不是

return FALSE;

s = &s[i];//去掉int

s = IgnoreB(s);//去掉空格 只留函数名（）

pos2 = Find(s, "(");

if (\*s == '\0' || \*s == '(' || pos2 > pos)//如果函数名为空，函数名括号开头，函数的括号出现在int中，一定不是

return FALSE;

return TRUE;

}

void printR(int aver, int comc, int blanks)

{

//按代码级别判定标准输出分析结果

int i;

char Grade[4][15] = { "A","B","C","D" };//定义四个级别段

//判定代码的级别

if (aver <= 15 && aver >= 10)

i = 0;

else if ((aver <= 20 && aver >= 16) || aver <= 9 && aver >= 8)

i = 1;

else if ((aver <= 7 && aver >= 5) || aver <= 24 && aver >= 21)

i = 2;

else if ((aver < 5) || (aver > 24))

i = 3;

printf(" 代码级别为 %s \n", Grade[i]);

//判定注释的级别

if (comc <= 25 && comc >= 15)

i = 0;

else if ((comc <= 14 && comc >= 10) || comc <= 30 && comc >= 26)

i = 1;

else if ((comc <= 9 && comc >= 5) || comc <= 35 && comc >= 31)

i = 2;

else if ((comc < 5) || (comc > 35))

i = 3;

printf(" 注释级别为 %s \n", Grade[i]);

//判定空行的级别

if (blanks <= 25 && blanks >= 15)

i = 0;

else if ((blanks <= 14 && blanks >= 10) || blanks <= 30 && blanks >= 26)

i = 1;

else if ((blanks <= 9 && blanks >= 5) || blanks <= 35 && blanks >= 31)

i = 2;

else if ((blanks < 5) || (blanks > 35))

i = 3;

printf(" 空行级别为 %s \n", Grade[i]);

}

//打印输出结果

void print(Analy\* An)

{

int sum = 0, funcode = 0;//总代码 函数代码行数

int i, comc, blanks, aver, code;

for (i = 0; i < An->funcount; i++)//求函数的代码总数

funcode += An->fun[i].length;

//求所有的代码总数

sum += An->blank;

sum += An->comments;

sum += An->comment;

sum += An->others;

sum += funcode;

if (sum == 0)//防止除数sum为0

sum = 1;

if (An->funcount == 0)//防止除数m为0

aver = 0;

else

aver = funcode / An->funcount;

comc = (An->comments + An->comment) \* 100 / sum;//求百分比

blanks = ((An->blank) \* 100) / sum;

code = 100 - comc - blanks; //((funcode + An->others)\*100)/sum;

printf(" 程序源代码分析结果如下所示\n\n");

printf(" 代码行数: %d\n", sum - An->blank - An->comment - An->comments);//代码中包含混合注释的长度

printf(" 注释行数: %d\n", An->comments + An->comment);

printf(" 空行数: %d\n", An->blank);

printf(" 代码 注释 空行\n");

printf(" \*\*\*\* \*\*\*\* \*\*\*\*\n");

printf(" \*\*\*\* \*\*\*\* \*\*\*\*\n");

printf(" %3d%% %3d%% %3d%%\n", code, comc, blanks);//%%表示百分号

printf(" 代码包含函数个数为 %d \n", An->funcount);

printf(" 函数的平均长度为 %d\n", aver);

printf("\n");

//按代码级别判定标准输出分析结果

printR(aver, comc, blanks);

}

void analy(char filename[COUNT][LEN])

{

FILE\* fp;//分析源文件指针

Analy An;//程序统计结构体

char s[2000];//存储每行的文件

BOOL begin = 0, start = 0;//设置函数开始标记

//设置函数开始结束标记 begin为是否出现函数 start为是否出现大括号

int i=0, j = -1, pos = 0;//函数的位置 长度信息 j为函数个数-1

//c检测函数内大括号的匹配,comtype是注释的类型

int c = 0, comtype;//c判断大括号个数 comtye判断一行是非注释 纯注释 混合注释

An.blank = 0;

An.comments = 0;

An.comment = 0;

An.others = 0;

An.funcount = 0;

if ((fp = fopen(filename[i], "r")) == NULL)

{

printf("无法打开文件%s\n", filename[i]); //输出filename[i]

getchar();//清除缓存

exit(0);

}

pos = 0; //函数在新一个文件中的位置初始化

while (!feof(fp))

{

HaveLine(fp, s); //从文件中读取一行数据

pos++; //每个函数在文件中开始的位置

//分类统计文件中的代码个数

comtype = IsCom(s);//判断该行是非注释 纯注释 混注释

if (comtype == 1)

{

An.comment++;

continue;

}

if (comtype == 2)//混注释

An.comments++;

//if (Find(s, "/\*") != -1)

//{

// s=IgnoreB(s);

// while (1)

// {

// if (Find(s, "\*/") == -1)

// {

// HaveLine(fp, s);

// An.comment++;

// }

// }

//}

if (IsFunB(s))

{

j++; //j为函数的个数-1;

c = 0;//大括号个数初始化为0

begin = TRUE;

An.fun[j].pos = pos;//函数位置初始化

An.fun[j].length = 1;

}

else if (IsBlank(s))//判断空白行

An.blank++;

else if (begin) {//要求函数的左大括号必须写在下一行，不能和函数名同一行

An.fun[j].length++;//还在函数内，函数长度自增

if (Find(s, "{") != -1) //检测是否进入到了函数体内

{

c++;

start = TRUE;//扫描到函数的左大括号

}

if (Find(s, "}") != -1)//检测是否一个函数的结束

c--;

if (c == 0 && start)

{

begin = FALSE;

start = FALSE;

}

}

else

An.others++;

}

fclose(fp);//关闭分析文件

An.funcount = j + 1; //把函数的个数保存

print(&An); //打印分析结果

}

#pragma warning(disable:4996)

#include<stdlib.h>

#include<string.h>

#define TRUE 1

#define FALSE 0

#define BOOL int

#define MAXSIZE 5000

#define COUNT 20 // 可以统计的最大的文件个数

#define LEN 20 // 文件名的最大长度

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

//函数属性结构

typedef struct {

int length; //每一个函数的长度

int pos; //每一个函数的位置

}Fun;

//统计结构的声明

typedef struct {

int comments; //纯注释的个数

int comment; //混合注释个数

int blank; //空行的个数

Fun fun[MAXSIZE]; //函数的属性

int others; //除去函数中代码外其余的代码个数

int funcount; //函数的个数

}Analy;

//串的基本操作ADT

//检测是否为空串

BOOL StrEmpty(char\* s);

//查看S1中是否有值为S2的子串,若有则返回第一个子串的位置，若无则返回-1；

int Find(char\* s1, const char\* s2);

//读取文件中的一行字符

void HaveLine(FILE\* fp, char\* s);

//忽略一行字符开头的空格和tab，返回截断后上的串指针

char\* IgnoreB(char\* s);

//判断一行字符是不是注释

int IsCom(char\* s);

//判断一行字符是不是空白

BOOL IsBlank(char\* s);

int IsFunB(char\* s);

void printR(int aver, int comc, int blanks);

//打印输出结果

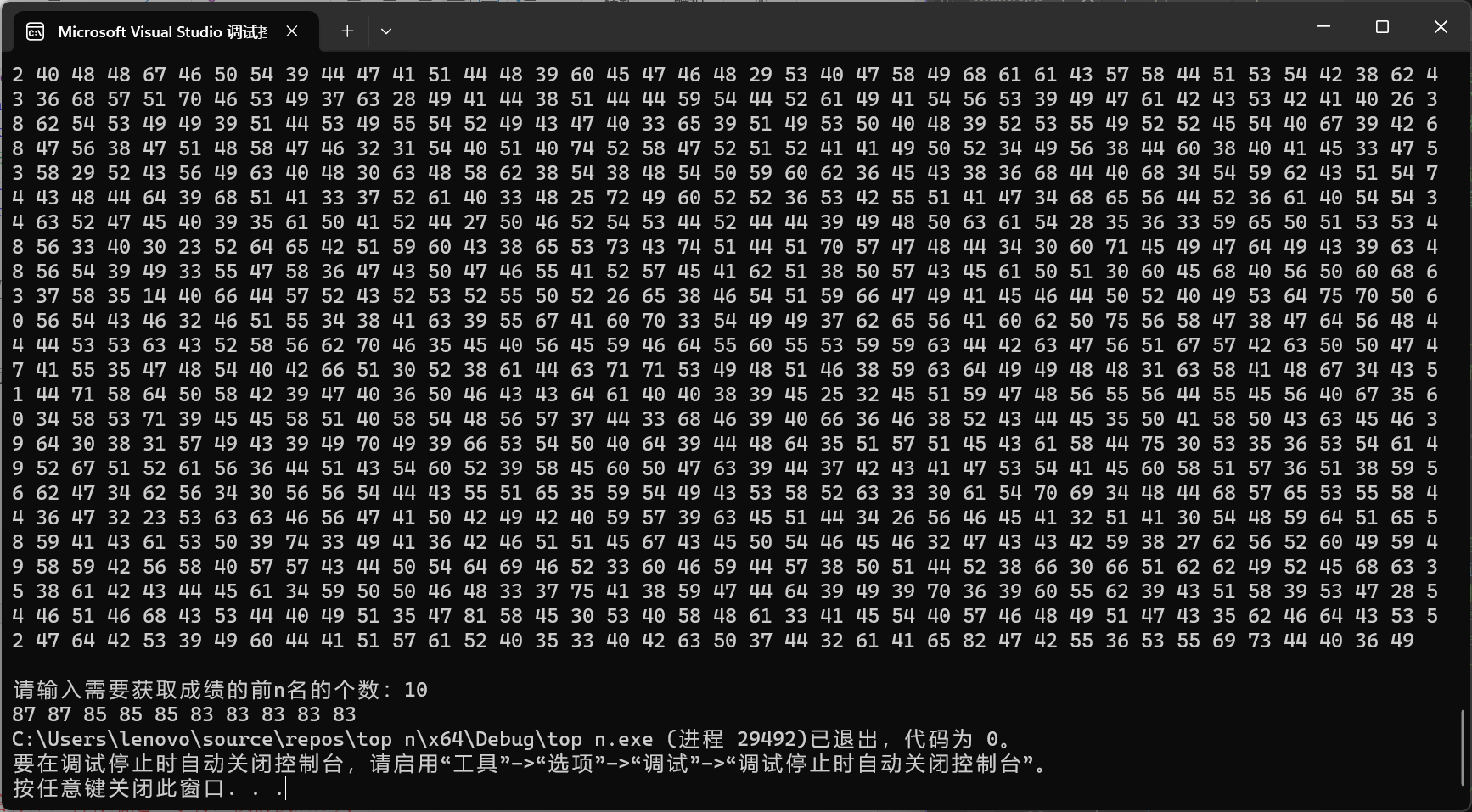
void print(Analy\* An);

void analy(char filename[COUNT][LEN]);

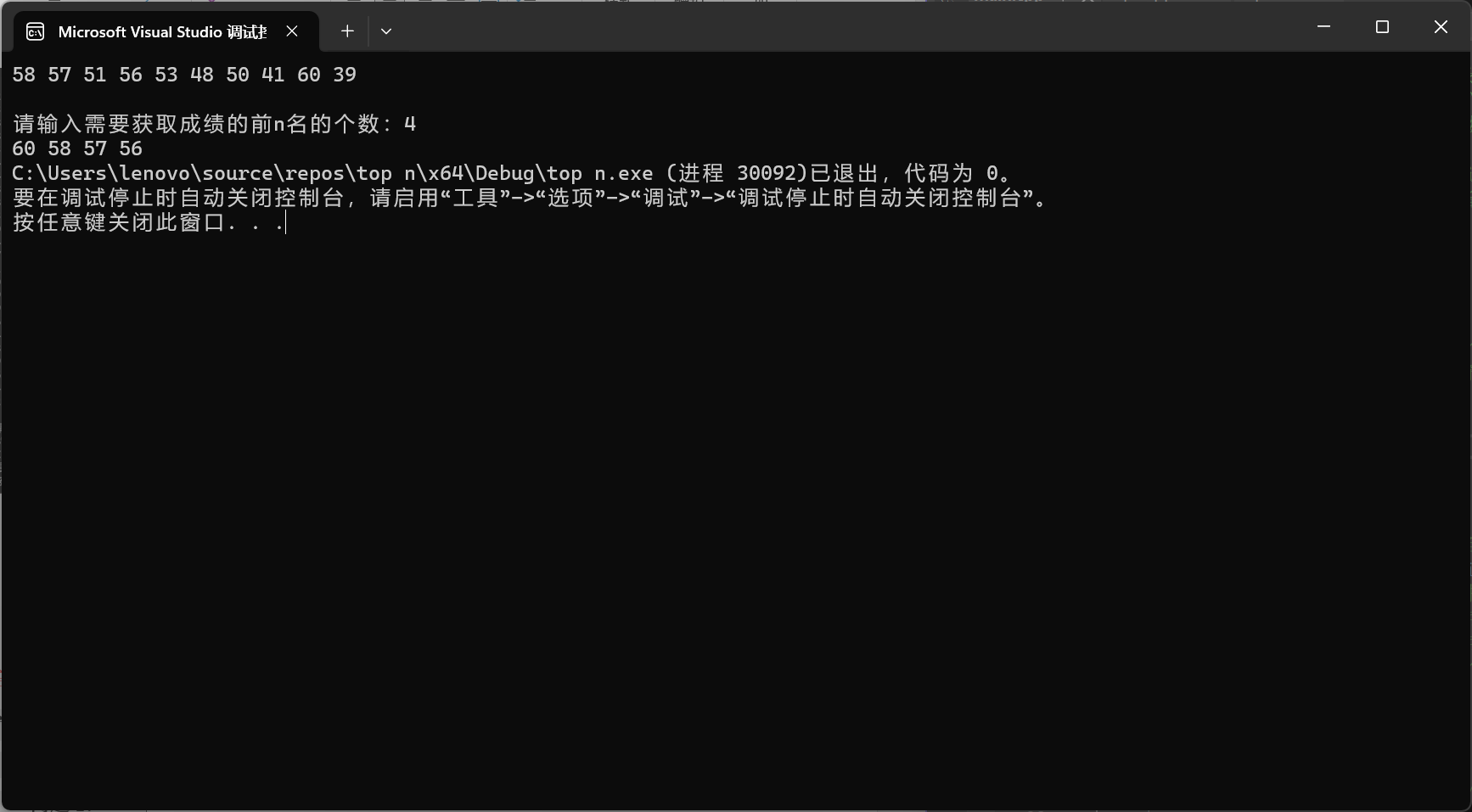
**五、测试数据及其结果分析**

应准备多组测试数据（包括合法数据、非法数据及边界数据等），对测试输出的结果进行分析；测试数据可采用三元组形式呈现：（输入，理想输出，实际输出）

Top\_n问题，图片3是对于实验的测试数据及结果分析，图片4是为了确定程序的准确性而简化的输出

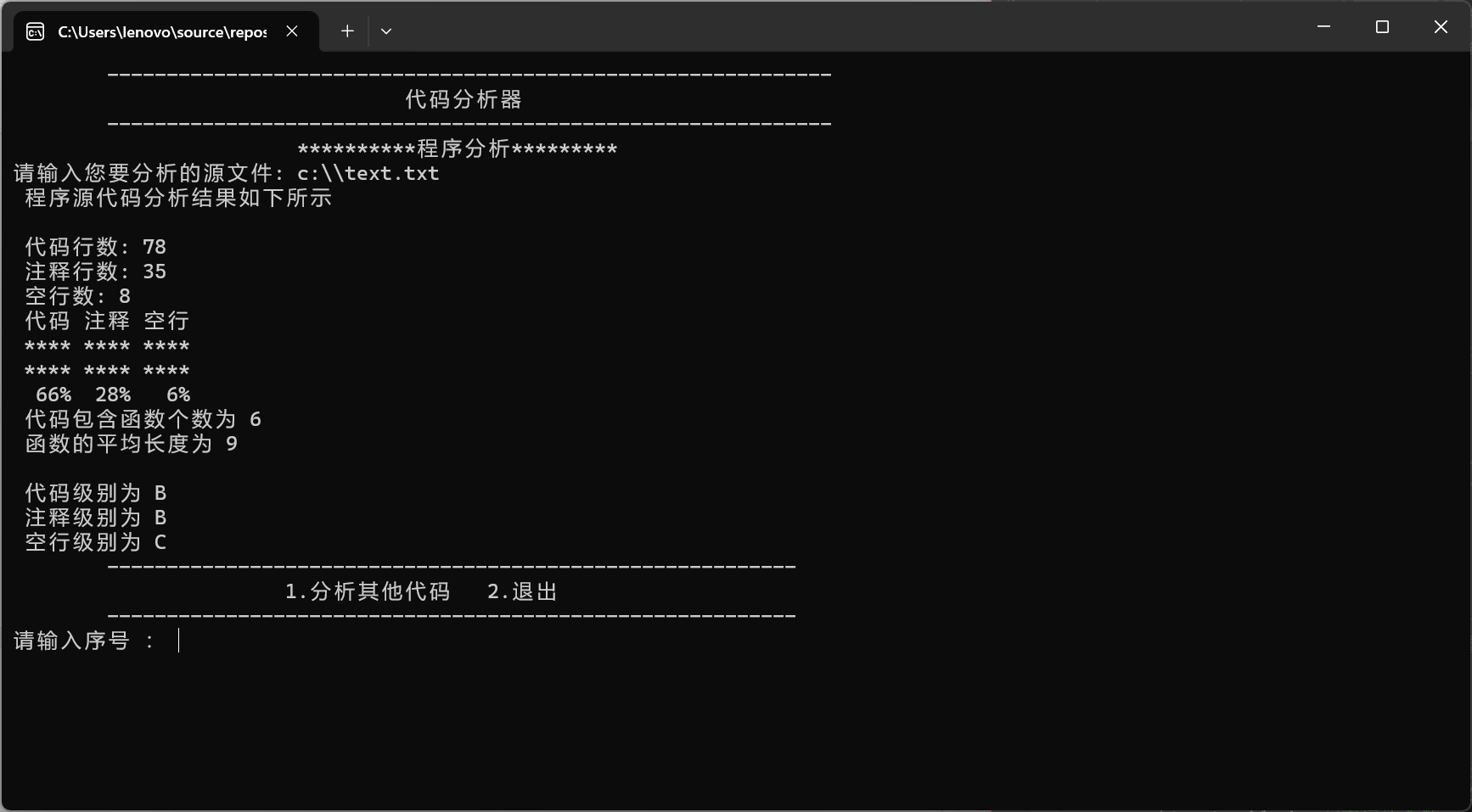


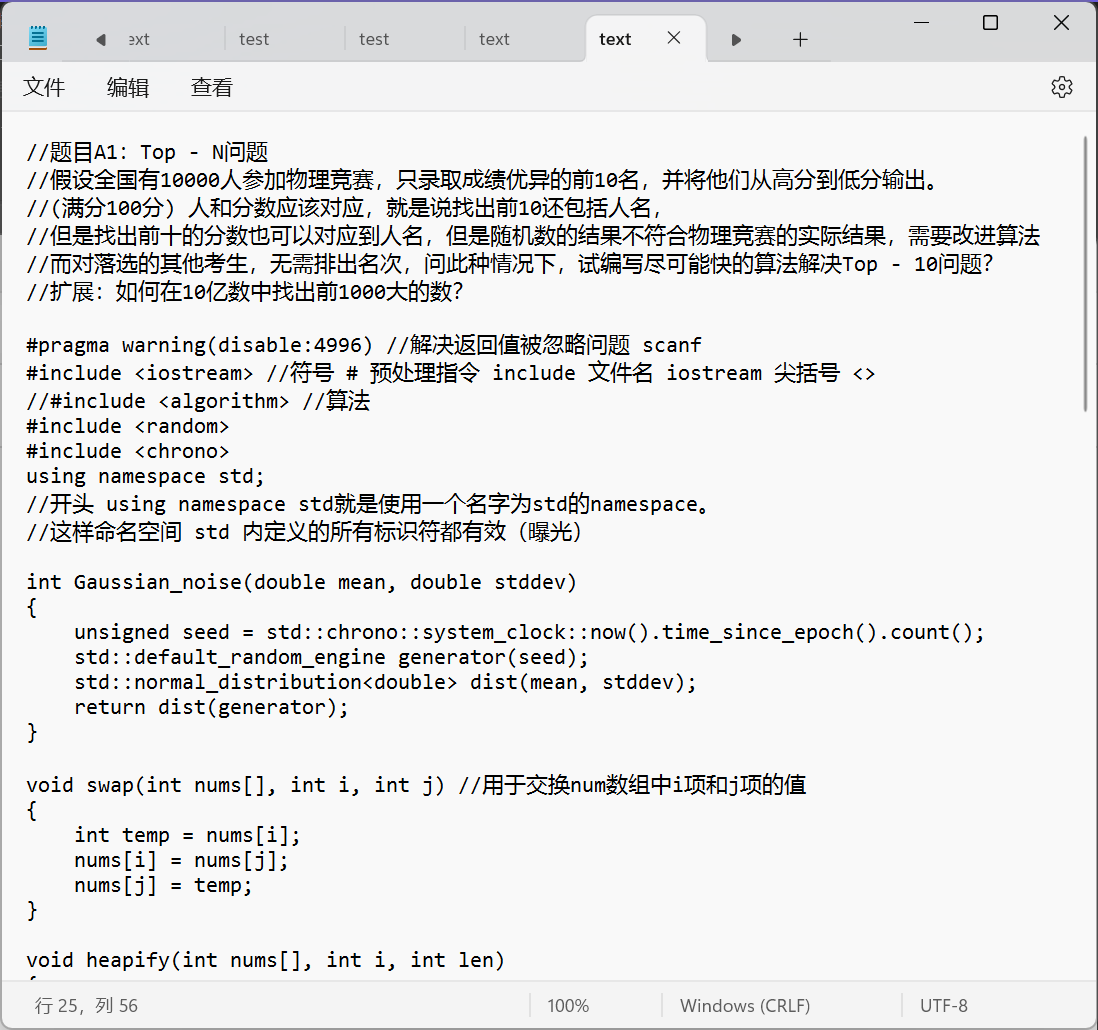
图片3



图片4

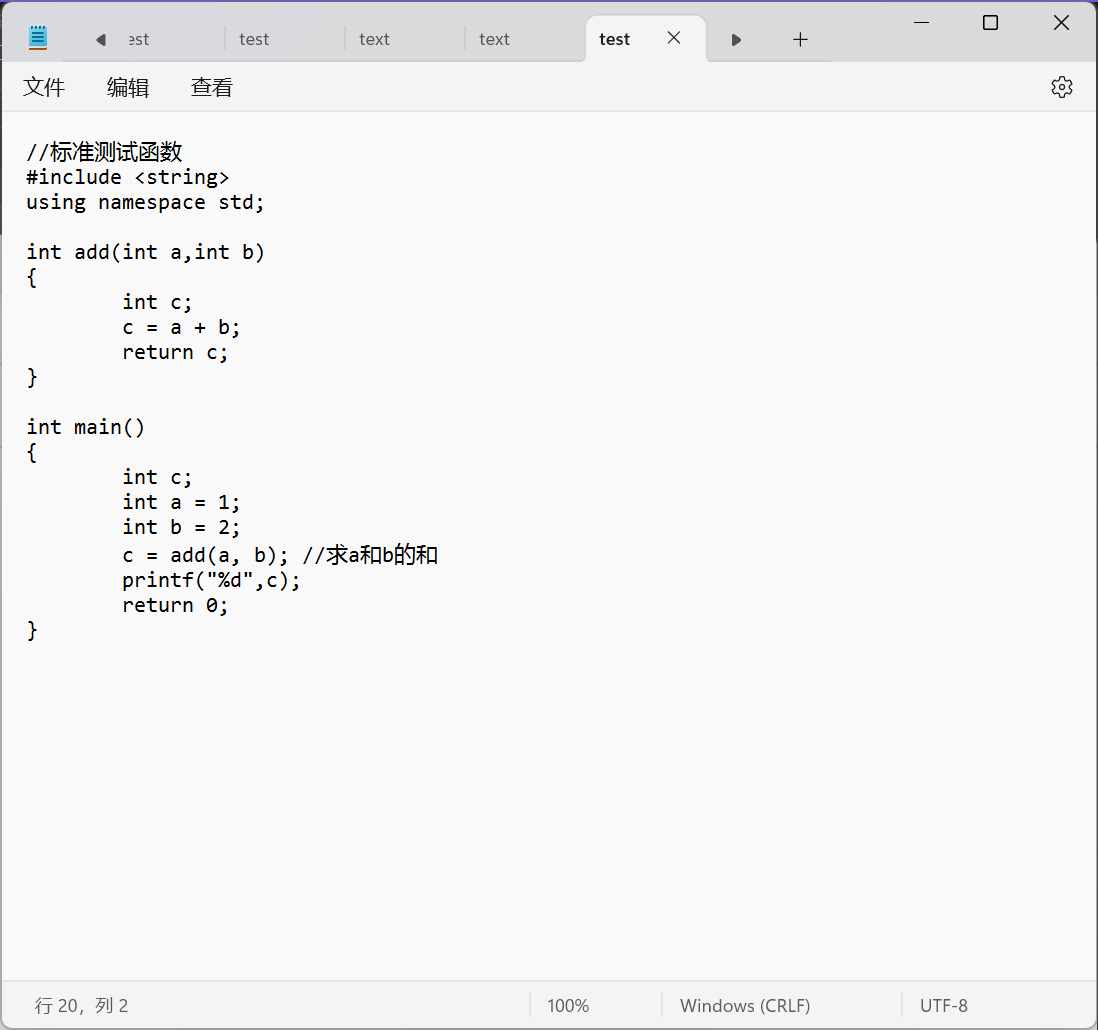
代码分析器，图片5是对文件text.txt，图片6是对文件test.txt，其中text.txt为我的top\_n代码





图片5





图片6

**六、算法设计和程序调试过程中的问题**

算法设计和程序调试时遇到哪些问题？你是如何解决的？（至少写3点）例如：

**问题1**：正常的随机分布不符合物理竞赛的实际

**解决方法**：采用了正态分布的算法

**问题2**：scanf存在返回值被忽略问题

**解决方法**：#pragma warning(disable:4996) //解决返回值被忽略问题 scanf

**问题3**：多行注释无法识别

**解决方法**：正则

**问题4**：代码总数为0作为除数的问题

**解决方法**：分类讨论

**七、课程设计总结**

算法进一步改进的设想：实现对物理竞赛实际数据的分析；实现对各种语言的代码的分析。

课程设计过程的收获和感受：重温了c 语言，又再次了解到了c语言编程的一些相关知识。