{

//================【开发篇】

1）d其他，先跟他打声招呼或者看文档，了解一下具体逻辑入参跟返回值，再调用该功能

解释：避免代码冗余，代码冗余会导致后面的版本升级需要更改多个方法。

2）a~b~c~d不一定符合公司的规范和个人习惯的写法

解释：每个人的写法不一样，公司要求的规范也不一样，搜索到的代码应该检查之后转为符合公司规范和个人习惯的代码使用，个人习惯要在公司要求的规范内按照自己习惯的方式写代码，后期对代码维护也能快速理解。

3）b

解释：变量名是让自己、同事、程序员能够快速的了解该变量的用途，如果随意命名，会导致该变量名的用途被曲解、遗忘

4）b

解释：注释能够让人理解即可。

5）b

解释：内存，分配给全局变量的内存要等程序结束才会释放并且全局变量提高了耦合度，后期维护麻烦也不便于其他人理解

6）b

解释：要看时间复杂度跟空间复杂度，短不代表精

7）b

解释：敲代码快，如果导致不易读或者浪费内存、运行速度，敲代码快也就没意义

8）c

解释：正常情况下，运行速度越快是好的，但是如果为了运行速度的快，而浪费大量的空间、不易读、后期维护和扩展麻烦，显然是不划算的

9）b

解释：合理的代码发布途径是git

10）b

解释：合理的代码发布途径是git

11）b

解释：合理的代码发布途径是git

12）b

解释：合理的代码发布途径是git

//================【测试篇】

1）a~b~c~d根据实际情况测试

解释：写多种难度的测试，根据不同难度和实际情况的测试结果分析程序的情况

2）b

解释:简单的测试也会出错，测试就要把所有的可能性测试清楚

3）a

解释：有多种难度的测试对该功能进行测试，放过一次，后续的测试也会对该功能继续测试，如果选择b，出现的错误太多，导致测试不可信，那测试的意义也就不大了

4）d使用多种难度的自动化测试,一份测试的代码出错不代表所有难度的代码都会出错

解释：写多种难度的自动化测试来保证，比其他选项更节省成本跟安全。

5）a~b~c~d

解释：都有可能，自己先排查清楚情况

6）f检查一下是不是自己不小心改了测试或者检查是不是第一次运行测试有一些变量的值改变导致第二次运行不通过

解释：从自己排查问题（不是很了解这个）

7）a~b~c~d~f根据实际情况测试

解释：开发文档或type规范中明令禁止的情况代表是还没解决的问题、或者有一定的要求，已经知道会有问题了，没有必要再浪费时间测试

//================【综合篇】

1）e使用自动化测试测试一下所有功能

解释：使用自动化测试检查整个程序，影响到的地方就会被测试出来，测试过后发现没问题就可以慢慢修改，这样就不会怕修改

2）b

解释：不一定，开发专注于开发，测试专注于测试，偶尔可以兼任，但不应该一起做，一方面是术业有专攻，另一方面是开发的时候按照自己的逻辑去开发，开发的人会觉得自己的逻辑没有问题，按照这个想法往往很难找到程序的漏洞，而测试是要找到程序的漏洞

3）a

解释：TDD,测试驱动开发，要实现某个功能时，先编写测试代码，明确这个功能如何使用，如何测试，然后再开发

4）数学

//================【基本功篇】

1）

2）

3）O(n)

4）O(1)

5）O(1)

6）O（logN)

7）O（n）

8）O（logN)

9）

//================【开放问题篇】

1）

{

先将拿到的资料整体的看一遍，对自己的任务有大概的了解，总结一下自己理解的任务，再去询问上级自己的任务是不是自己理解的

}

2)

{

写一个自动化脚本测试这个除法程序有没有bug，再看一下使用这个除法程序的时间是不是比编程语言自带的除法更节省时间或者空间，是否有必要使用这个除法程序

空间情况使用vs工具查看内存

该程序测试这个这个除法程序有没有bug与运行时间

计算运行时间取决于当前电脑进程还有cpu等情况，运行时间的好坏在时间充足的情况下最好通过计算时间复杂度来评价

using System;

using System.Diagnostics;

namespace ConsoleApp1

{

class Program

{

//假设这是别人自己写的一个 除法程序，并且声称比编程语言自带的 除法 更好

static double division(double d1, double d2)

{

double d = d1 / d2;

return d;

}

static void Main(string[] args)

{

try

{

//测试别人写的除法程序耗时

DateTime start1 = DateTime.Now;

//检查该除法程序当d1,d2为正数的除法运算

for (int i = 0; i <= 100; i++)

{

//有测试过随机数的问题，看到的基本都是不重复的

Random rn1 = new Random();

Random rn2 = new Random();

double d1 = rn1.Next(10000000)+1;

double d2 = rn2.Next(10000000)+1;

//如果出现异常，方便查看出现异常的数值情况

Console.WriteLine(d1 + "/" + d2);

double d = division(d1, d2);

Console.WriteLine("d1,d2为正数的结果" + d);

}

//检查该除法程序当d1,d2为负数的除法运算

for (int i = 0; i <= 100; i++)

{

Random rn1 = new Random();

Random rn2 = new Random();

double d1 = (rn1.Next(10000000)+1) \* -1;

double d2 = (rn2.Next(10000000)+1) \* -1;

//如果出现异常，方便查看出现异常的数值情况

Console.WriteLine(d1 + "/" + d2);

double d = division(d1, d2);

Console.WriteLine("d1,d2为负数的结果" + d);

}

//检查该除法程序当d1为正数,d2为负数的除法运算

for (int i = 0; i <= 100; i++)

{

Random rn1 = new Random();

Random rn2 = new Random();

double d1 = rn1.Next(10000000);

double d2 = rn2.Next(10000000) \* -1;

//如果出现异常，方便查看出现异常的数值情况

Console.WriteLine(d1 + "/" + d2);

double d = division(d1, d2);

Console.WriteLine("d1为正数,d2为负数的结果" + d);

}

//检查该除法程序当d1为负数,d2为正数的除法运算

for (int i = 0; i <= 100; i++)

{

Random rn1 = new Random();

Random rn2 = new Random();

double d1 = (rn1.Next(10000000) + 1) \* -1;

double d2 = rn2.Next(10000000)+1;

//如果出现异常，方便查看出现异常的数值情况

Console.WriteLine(d1 + "/" + d2);

double d = division(d1, d2);

Console.WriteLine("d1为负数,d2为正数的结果" + d);

}

TimeSpan spn1 = DateTime.Now - start1;

//测试编程语言自带的耗时

DateTime start2 = DateTime.Now;

for (int i = 0; i <= 100; i++)

{

//有测试过随机数的问题，看到的基本都是不重复的

Random rn1 = new Random();

Random rn2 = new Random();

double d1 = rn1.Next(10000000) + 1;

double d2 = rn2.Next(10000000) + 1;

//如果出现异常，方便查看出现异常的数值情况

Console.WriteLine(d1 + "/" + d2);

double d = d1 / d2;

Console.WriteLine("d1,d2为正数的结果" + d);

}

//检查该除法程序当d1,d2为负数的除法运算

for (int i = 0; i <= 100; i++)

{

Random rn1 = new Random();

Random rn2 = new Random();

double d1 = (rn1.Next(10000000) + 1) \* -1;

double d2 = (rn2.Next(10000000) + 1) \* -1;

//如果出现异常，方便查看出现异常的数值情况

Console.WriteLine(d1 + "/" + d2);

double d = d1 / d2;

Console.WriteLine("d1,d2为负数的结果" + d);

}

//检查该除法程序当d1为正数,d2为负数的除法运算

for (int i = 0; i <= 100; i++)

{

Random rn1 = new Random();

Random rn2 = new Random();

double d1 = rn1.Next(10000000);

double d2 = rn2.Next(10000000) \* -1;

//如果出现异常，方便查看出现异常的数值情况

Console.WriteLine(d1 + "/" + d2);

double d = d1 / d2;

Console.WriteLine("d1为正数,d2为负数的结果" + d);

}

//检查该除法程序当d1为负数,d2为正数的除法运算

for (int i = 0; i <= 100; i++)

{

Random rn1 = new Random();

Random rn2 = new Random();

double d1 = (rn1.Next(10000000) + 1) \* -1;

double d2 = rn2.Next(10000000) + 1;

//如果出现异常，方便查看出现异常的数值情况

Console.WriteLine(d1 + "/" + d2);

double d = d1 / d2;

Console.WriteLine("d1为负数,d2为正数的结果" + d);

}

TimeSpan spn2 = DateTime.Now - start2;

Console.WriteLine("该除法程序耗时：" + spn1.ToString());

Console.WriteLine("编程语言自带的除法程序耗时：" + spn2.ToString());

}catch (Exception e)

{

Console.WriteLine("程序出现异常" + e.ToString());

}

try

{

//检查该除法程序当d1为0或者d2为0时的处理情况

//d2为零是为了查看该除法程序当遇到d2为零的情况时，如何处理

Random rn3 = new Random();

double d3 = 0;

double d4 = rn3.Next(10000000);

double dR1 = division(d3, d4);

Console.WriteLine("d1为0结果" + dR1);

Random rn5 = new Random();

double d5 = rn5.Next(10000000);

double d6 = 0;

double dR2 = division(d5, d6);

Console.WriteLine("d2为0结果" + dR2);

}

catch (Exception e)

{

Console.WriteLine("该除法程序对当d1为0或者d2为0时没有做出异常处理" + e.ToString());

}

Console.ReadKey();

}

}

}

}

3)

{

放到程序上运行（我使用的vs），会出现个控制台窗口，将答题者的代码复制粘贴到窗口里，复制就全选复制即可，粘贴只需要在窗口里鼠标右键就会把代码粘贴到窗口里，粘贴结束之后在窗口输入1，代表粘贴结束开始判卷，单选题程序直接判分，多选题由面试官给分，题目的分数面试官也可修改，面试官判分结束程序会给出答题者的成绩

程序只计算开发篇、测试篇、综合篇的答题情况，答案我参考了我自己的答题情况。

}

using System;

using System.Collections.Generic;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

//没搜到适合的，就用了list，查询会比较慢，不过这样子写可读性比较好

//存放开发篇答案

List<string> developmentList = new List<string>();

developmentList.Add("d");

developmentList.Add("a~b~c~d");

developmentList.Add("b");

developmentList.Add("b");

developmentList.Add("b");

developmentList.Add("b");

developmentList.Add("b");

developmentList.Add("c");

developmentList.Add("b");

developmentList.Add("b");

developmentList.Add("b");

developmentList.Add("b"); //12个元素 索引为11

//存放测试篇答案

developmentList.Add("a~b~c~d");

developmentList.Add("b");

developmentList.Add("a");

developmentList.Add("d");

developmentList.Add("a~b~c~d");

developmentList.Add("f");

developmentList.Add("a~b~c~d~f");//7个元素 索引为18'

//存放综合篇答案

developmentList.Add("e");

developmentList.Add("b");

developmentList.Add("a");

developmentList.Add("数学");//4个元素 索引为22

//存放开发篇答案分数

List<int> developmentGradeList = new List<int>();

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

//存放测试篇答案分数

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

//存放综合篇答案分数

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

developmentGradeList.Add(2);

//存放答题者答案

String result = "";

//存放答题者分数

double gradeSum = 0.0;

//读取多行文本,str表示何时结束

String str ="";

//while (str != "");

while (str != "1")

{

result += str;

str = Console.ReadLine();

}

String[] results=result.Split("============");

//开发篇判卷步骤

for (int i = 1; i <= 12; i++)

{

int index1 = results[1].IndexOf(i + "）");

int index2 = results[1].IndexOf(i + 1 + "）");

//每道题的答案

String answerString = "";

if (i != 12)

{

answerString = results[1].Substring(index1, index2 - index1 - 1);

}

else

{

answerString = results[1].Substring(index1);

}

//Console.WriteLine(answer);

if (i == 1 || i == 2)

{

if(i == 1)

{

Console.WriteLine("1)【多选题】如果你需要的一个功能正好被公司里其他同事之前已经写过（需求一模一样）了 ，（或是被自己之前写过了）\na先跟他打声招呼，然后把他的代码复制粘贴过来用\nb不复制他的代码而是手动重新照着敲一遍他的代码 这样还能做些因地制宜的修改\nc避免重复，故意换一种不同方法去实现。\nd其他\_\_\_\_\_\_\_\_\_\_\_");

}

else

{

Console.WriteLine("2)【多选题】如果你正要写的一块代码碰巧在网上能搜索到，直接复制过来用可能有哪些不好？\na侵犯版权\nb网上免费找的代码质量不一定好（不规范）\nc你怎么知道网上别人写的代码就是对的？\nd其他\_\_\_\_\_\_\_\_\_\_\_");

}

/\* int a=answer.IndexOf("d");

Console.WriteLine(answer.Substring(1, a));

Console.WriteLine(answer.Substring(a, index2 - index1 - 5));\*/

Console.WriteLine("这道题正确答案为" + developmentList[i - 1]);

Console.WriteLine("答题者选择"+ answerString.Substring(2, index2 - index1 -3));

Console.WriteLine("请判分，这道题你的满分设定为"+ developmentGradeList[i-1]);

gradeSum += double.Parse(Console.ReadLine());

}

else

{

String answer = "";

if (i == 10 || i == 11 || i == 12)

{

answer=answerString.Substring(3, 1);

Console.Write("第"+i+"题答题者选择" + answer);

}

else

{

answer = answerString.Substring(2, 1);

Console.Write("第"+i+"题答题者选择" + answer);

}

if(answer.Trim()== developmentList[i - 1])

{

Console.WriteLine(",正确" );

gradeSum += developmentGradeList[i - 1];

}

else

{

Console.WriteLine(",错误");

}

}

}

//测试篇判卷步骤

for (int i = 1; i <= 7; i++)

{

int index1 = results[2].IndexOf(i + "）");

int index2 = results[2].IndexOf(i + 1 + "）");

//每道题的答案

String answerString = "";

if (i != 7)

{

answerString = results[2].Substring(index1, index2 - index1 - 1);

}

else

{

answerString = results[2].Substring(index1);

}

//Console.WriteLine(answer);

if (i == 1 || i == 4 || i == 5 || i == 6 || i == 7)

{

if (i == 1)

{

Console.WriteLine("1)【多选题】写测试的基本原则\na专门做高难度测试\nb专门做低难度测试\nc专门做中难度测试\nd其他\_\_\_\_\_\_\_\_\_\_\_");

}

else if (i == 4)

{

Console.WriteLine("4)【多选题】如何保证用来【测试】的代码不会出问题\na给【测试】再写一份测试，即测试的测试。然后为了进一步保障测试的测试不出问题，我们应该还要写测试的测试的测试才能做到万无一失。\nb每次修改后都把代码编译出来自己亲自试用一下，还是人肉测试靠谱\nc出问题就出问题吧，不被老板发现就行\nd其他\_\_\_\_\_\_\_\_\_\_\_");

}

else if (i == 5)

{

Console.WriteLine("5)【多选题】第一次运行测试的时候没通过说明什么：\na写代码的人出错了\nb写测试的人出错了\nc环境没配置好\nd测试模板没继承好\ne其他\_\_\_\_\_\_\_\_\_\_\_");

}

else if (i == 6)

{

Console.WriteLine("6)【多选题】第二次运行测试的时候没通过，但是之前明明通过了的，怎么办：\na邪门了！多运行几次，一直重试到它通过为止。然后就别再去碰它了！\nb电脑出问题了 ? 重启一下就好了。重启电脑的时候顺便去多喝热水。\nc对所有人吼一声： 谁把我的代码改了！！！\nd对所有人吼一声： 谁把测试改了！！！\ne一定是公司的电脑有问题，甚至可能中毒了，被入侵了，叫老板给我换更贵的电脑。\nf其他\_\_\_\_\_\_\_\_\_\_\_");

}

else if (i == 7)

{

Console.WriteLine("6)【多选题】第二次运行测试的时候没通过，但是之前明明通过了的，怎么办：\na邪门了！多运行几次，一直重试到它通过为止。然后就别再去碰它了！\nb电脑出问题了 ? 重启一下就好了。重启电脑的时候顺便去多喝热水。\nc对所有人吼一声： 谁把我的代码改了！！！\nd对所有人吼一声： 谁把测试改了！！！\ne一定是公司的电脑有问题，甚至可能中毒了，被入侵了，叫老板给我换更贵的电脑。\nf其他\_\_\_\_\_\_\_\_\_\_\_");

}

Console.WriteLine("这道题正确答案为" + developmentList[i + 11]);

if (index2 == -1)

{

Console.WriteLine("答题者选择" + answerString.Substring(2));

}

else

{

Console.WriteLine("答题者选择" + answerString.Substring(2, index2 - index1 - 3));

}

Console.WriteLine("请判分，这道题你的满分设定为" + developmentGradeList[i + 11]);

gradeSum += double.Parse(Console.ReadLine());

}

else

{

String answer = answerString.Substring(2, 1);

Console.Write("第" + i + "题答题者选择" + answer);

if (answer.Trim() == developmentList[i + 11])

{

Console.WriteLine(",正确");

gradeSum += developmentGradeList[i + 11];

}

else

{

Console.WriteLine(",错误");

}

}

}

//综合篇判卷步骤

for (int i = 1; i <= 4; i++)

{

int index1 = results[3].IndexOf(i + "）");

int index2 = results[3].IndexOf(i + 1 + "）");

//每道题的答案

String answerString = "";

if (i != 4)

{

answerString = results[3].Substring(index1, index2 - index1 - 1);

}

else

{

answerString = results[3].Substring(index1);

}

//Console.WriteLine(answer);

if (i == 1)

{

Console.WriteLine("1)【多选题】如何保证修改某处的一点点代码后不会影响其他地方\na每次修改后都把其他可能被影响的代码检查一遍 所以越到后面要检查的代码越多也是没办法的事情\nb每次修改后都把代码编译出来自己试用一下\nc每次修改后都扔给别人去测试\nd每次修改后都把代码编译出来扔给用户去测试\ne其他\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("这道题正确答案为" + developmentList[i - 1]);

Console.WriteLine("答题者选择" + answerString.Substring(2, index2 - index1 - 3));

Console.WriteLine("请判分，这道题你的满分设定为" + developmentGradeList[i - 1]);

gradeSum += double.Parse(Console.ReadLine());

}

else

{

String answer = "";

if (i == 4)

{

answer = answerString.Substring(2, 2);

}

else

{

answer = answerString.Substring(2, 1);

}

Console.Write("第" + i + "题答题者选择" + answer);

if (answer.Trim() == developmentList[i +18])

{

Console.WriteLine(",正确");

gradeSum += developmentGradeList[i + 18];

}

else

{

Console.WriteLine(",错误");

}

}

}

int sum = 0;

foreach (int i in developmentGradeList)

{

sum += i;

}

if(gradeSum>sum \* 0.9)

{

Console.WriteLine("满分为"+sum+",答题者分数为"+ gradeSum+",该答题者成绩优秀");

}

else if (gradeSum > sum \* 0.8)

{

Console.WriteLine("满分为" + sum + ",答题者分数为" + gradeSum + ",该答题者成绩良好");

}

else if(gradeSum > sum \* 0.6)

{

Console.WriteLine("满分为" + sum + ",答题者分数为" + gradeSum + ",该答题者及格");

}

else

{

Console.WriteLine("满分为" + sum + ",答题者分数为" + gradeSum + ",该答题者不及格");

}

Console.WriteLine(sum);

Console.WriteLine(gradeSum);

Console.ReadKey();

}

}

}

}