

# SE Individual Project

---

2106050130 徐圣翔

## SE Individual Project

四则运算——初级

题目

需求分析

设计

编码

测试

思路说明

时间花费比较

## 四则运算——初级

---

### 题目

编写一个四则运算程序，满足如下要求

1. 题目要求：100以内加减法，满足小学二年级数学口算需求
2. 使用参数控制生成题目的个数
3. 每道题目中出现的运算符个数不超过两个
4. 程序一次运行生成的题目不能重复，请思考关于重复的定义。生成的题目存入执行程序的当前目录下的Exercises.txt文件，格式如下：
  1. 四则运算题目1
  2. 四则运算题目2
5. 再生成题目的同时，计算出所有题目的答案，并存入执行程序的当前目录下的Answers.txt文件，格式如下：
  1. 答案1
  2. 答案2
6. 估计需求分析、设计、编码、测试各阶段时间，记录实际工作中各项工作时间花费，并列表进行对比

### 需求分析

start at 20:30

1. 首先，100以内加减法，100以内加减法是指参与运算的数不超过100，还是运算结果不超过100？  
通过对市面上的100以内加减法的习题的分析研究，以及对于小学二年级家长与老师的采访调研，得出100以内加减法，100以内加减法是指参与运算的数和运算结果都不超过100的含义。
2. 使用参数控制生成题目的个数就是需要用户输入数量
3. 每道题目中出现的运算符个数不超过两个即为，运算符只能是一个或者两个
4. 一次运行生成的题目不能重复，那么对于一个运算符的情况，不能出现1+2和2+1或1+2和1+2的情况，对于两个运算符情况，不能出现1+1+2与1+2+1的情况
5. 生成的题目及其答案需要对应的存入相应的文件里面

end at 20:42

cost time 12min

## 设计

start at 22:35

1. 用num存储生成题目的个数。
2. 用随机数ra的0和1表示运算符的个数，0代表1个运算符，1代表2个运算符。
3. 再用随机数opt的0和1代表+-运算符，0代表+，1代表-。
4. 随机生成参与运算的0-100之间数字，首先计算结果，如果大于100就不记入已产生的四则运算式子。
5. 对于重复的，我们可以采用数据结构来存储，将加减号转换为正负号，存入[-100,+100]的两个或者三个数组内。
6. 使用文件流操作，将数据读入txt其中。

end at 22:48

cost time 13min

## 编码

begin at 0:07

```
1  #include <bits/stdc++.h>
2  // #define LOCAL
3  using namespace std;
4  typedef long long ll;
5  #define IOS ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
6  const double pi=acos(-1.0);
7  const int INF=1000000000;
8  const int maxn=1000005;
9  int num,ans=0;
10 int plustwo[110][110],minutwo[110][110],operatthree[500][500][500];
11 int main()
12 {
13     IOS;
14     #ifdef LOCAL
15         freopen("input.txt","r",stdin);
16         freopen("output.txt","w",stdout);
17     #endif
18     srand(time(NULL));
19     memset(plustwo,0,sizeof(plustwo));
20     memset(minutwo,0,sizeof(minutwo));
21     memset(operatthree,0,sizeof(operatthree));
22     cout<<"please input the number of the questions that you need:"<<endl;
23     cin>>num;
24     fstream f;
25     f.open("Exercises.txt",ios::out);
26     f.close();
27     f.open("Answers.txt",ios::out);
28     f.close();
29     while(ans<num){
30         int ra=rand()%2+1;
31         if(ra==1){
32             int a1=rand()%100+1,a2=rand()%100+1,opt=rand()%2+1,res;
33             if(opt==1){
```

```

34         res=a1+a2;
35         if(res<0||res>100) continue;
36         if(plustwo[a1][a2]==1) continue;
37         plustwo[a1][a2]=1;
38         //cout<<ans+1<<": "<<a1<<"+ "<<a2<< "="<<res<<endl;
39         f.open("Exercises.txt",ios::out|ios::app);
40         f<<ans+1<<": "<<a1<<"+ "<<a2<< "="<<endl;
41         f.close();
42         f.open("Answers.txt",ios::out|ios::app);
43         f<<ans+1<<": "<<res<<endl;
44         f.close();
45     }
46     else{
47         res=a1-a2;
48         if(res<0||res>100) continue;
49         if(minutwo[a1][a2]==1) continue;
50         minutwo[a1][a2]=1;
51         //cout<<ans+1<<": "<<a1<< "- "<<a2<< "="<<res<<endl;
52         f.open("Exercises.txt",ios::out|ios::app);
53         f<<ans+1<<": "<<a1<< "- "<<a2<< "="<<endl;
54         f.close();
55         f.open("Answers.txt",ios::out|ios::app);
56         f<<ans+1<<": "<<res<<endl;
57         f.close();
58     }
59     ans++;
60 }
61 else {
62     int
a1=rand()%100+1,a2=rand()%100+1,a3=rand()%100+1,res,opt1=rand()%2+1,opt2=rand()
d()%2+1;
63     int b1=a1,b2=a2,b3=a3;
64     if(opt1==2) b2=-b2;
65     if(opt2==2) b3=-b3;
66     res=b1+b2+b3;
67     if(res<0||res>100) continue;
68     if(operatthree[b1][200+b2][200+b3]==1) continue;
69     operatthree[b1][200+b2][200+b3]=1;
70     /*cout<<ans+1<<": "<<a1;
71     if(opt1==1) cout<<"+ ";
72     else cout<< "- ";
73     cout<<a2;
74     if(opt2==1) cout<<"+ ";
75     else cout<< "- ";
76     cout<<a3<< "="<<res<<endl;*/
77     f.open("Exercises.txt",ios::out|ios::app);
78     f<<ans+1<<": "<<a1;
79     if(opt1==1) f<<"+ ";
80     else f<< "- ";
81     f<<a2;
82     if(opt2==1) f<<"+ ";
83     else f<< "- ";
84     f<<a3<< "="<<endl;
85     f.close();
86     f.open("Answers.txt",ios::out|ios::app);
87     f<<ans+1<<": "<<res<<endl;
88     f.close();
89     ans++;

```

```
90     }
91     }
92     return 0;
93 }
```

end at 0:33

cost time 26min

## 测试

start at 0:37

无bug

输入: 20

```
1 //Exercises.txt
2 1:46+88-51=
3 2:90-78=
4 3:69-48=
5 4:47+100-91=
6 5:18+52=
7 6:69-80+47=
8 7:85+77-64=
9 8:41+42+14=
10 9:98-66=
11 10:95-67=
12 11:8+66+17=
13 12:8+64-59=
14 13:16-9=
15 14:38+39=
16 15:81-56+26=
17 16:70-32+53=
18 17:69-23+40=
19 18:87-74=
20 19:64+8=
21 20:72-97+44=
```

```
1 //Answers.txt
2 1:83
3 2:12
4 3:21
5 4:56
6 5:70
7 6:36
8 7:98
9 8:97
10 9:32
11 10:28
12 11:91
13 12:13
14 13:7
15 14:77
16 15:51
17 16:91
18 17:86
```

19	18:13
20	19:72
21	20:19

end at 0:43

cost time 6min

## 思路说明

首先输入要生成多少条四则运算题目。

然后对于每一条题目我们来随机生成是一个运算符还是两个运算符。

如果是一个运算符，随机生成a+b或者a-b的形式。

如果是两个运算符，就随机生成a+b+c或a+b-c或a-b+c或a-b-b的形式。

然后我们来考虑如何避免重复的情况发生。

对于一个运算符+的情况，我们用二维数组来存储a和b是否出现过。

对于一个运算符-的情况，我们再用另一个二维数组来存储a和b是否出现过。

对于两个运算符，我们将减100变成加-100，从而全部转换到+域上来。

简而言之就是当运算符为-的时候，将后面的一个数字变为相反数。

然后我们只需要用一个三维数组来存储a、b、c是否出现过即可。

最后将运算式及其答案分别存入文件即可。

## 时间花费比较

需求分析	12minutes
设计	13minutes
编码	26minutes
测试	6minutes