Project in ME001 – Sampling system

Group 1

By: Chen YuXuan 1809853J-I011-0011
& Wang Yuan 1809853G-I011-0030
& He PeiLin 1809853U-I011-0078

Outline

- Restatement of the problem
- Basic Ideas
- Essential Codes and Functions Analysis
- Program Test
- Summary

Restatement of the problem

In this project, we are expected to extract a subset of samples of big data. Assume there are m samples $(45 \leq m \leq 54)$, any $n(7 \leq n \leq 25)$ samples out of these m samples are selected.
There are C_n^m groups of n samples. From one of these groups of n samples, we randomly selected $k(4 \leq k \leq 7)$ samples to form some groups. So there will be C_n^k groups of k samples selected. There are at least **ONE** group of k samples, in which $s(3 \leq s \leq 7)$ samples have been selected from the j (where $s \leq j \leq k$) samples.
Among these groups of k samples, we would like to optimize them by selecting **ONLY** some of them.
We can divide the problem into two parts, $j = s$ and $j \neq s$.

Basic Ideas

When $j = s$:

Algorithm to Find Subsets

Now, we have a set whose number of the element is n. Then we want to find out all the subsets whose number of the element is k.

Algorithm:

- First, we put the origin set to a container, and then we label every element to one(illustrate the picture below). We assume that the origin set is S, $S = \{1, 2, 3, 4, 5\}$ in Table 1.

5	4	3	2	1
1	1	1	1	1

Table 1

Then, the subset which has the same element with the original set's is labeled the element to 1, otherwise labeling it to 0. For example, we suppose that one the subset is S_1,$S_1 = \{1, 2, 4\}$. We can represent it as Table 2.

Basic Ideas

- Now we can change the number below the array to a binary number, which means that each subset can be represented by a unique number from 0(empty set) to $2^n - 1$(original set). Just like the example above set S can be represented by $11111_2 = 31_{10}$ and S_1 can be expressed as $01011_2 = 11_{10}$

5	4	3	2	1
0	1	0	1	1

Table 2

Basic Ideas

- Subsequently, we know how to find subsets of the original set, but I want to know how to find the subset with the specific number of elements. Therefore, we only need to know the subset whose binary number representation contains k 1s. As the example in Table 2, $S_1 = \{1, 2, 4\}$, So, the S_1 contains three elements, because it has three *1*s. In this way, we can easily find out the subset whose number of elements is k from 0 to $2^n - 1$, the code block **findSubsetOfk** illustrates the situation.

Basic Ideas

```
1  void findSubsetOfk(int n,int k, vector<int> subsetK){
2      int count=0;//number of 1s
3      for(int i = 1 ; i < (1<<n); i++){
4          for(int j = 0; j < n; j++){
5              //the binary number representation
6              //of subset has an 1 on the jth position
7              if(i & (1<<j)!=0){
8                  count++;
9              }
10         }
11         if(count==k)
12             susetK.empalce_back(i);
13         count=0;
14     }
15
16 }
```

Basic Ideas

- However, we can easily find that the binary number representation of the subset whose number of elements is k is no less than $2^k - 1$. Therefore, we the code above, we can have an optimization on the i. The optimized code **findSubsetOfkOptim** is

```
1 void findSubsetOfkOptim(int n,int k, vector<int> subsetK){
2     int count=0;//number of 1s
3     for(int i = (1<<k)-1 ; i < (1<<n); i++){
4         for(int j = 0; j < n; j++){
```

Basic Ideas

```
5                   //the binary number representation
6                   //of subset has an 1 on the jth position
7                   if(i & (1<<j)!=0){
8                       count++;
9                   }
10              }
11              if(count==k)
12                  susetK.empalce_back(i);
13              count=0;
14          }
15
16  }
```

- Currently, we can use the same way what we say above to find out the subset of the set whose number of element is k and its number of elements is s.

Basic Ideas

Calculate the Combination Number

If we calculate the combination number directly, it is likely to out of bounds of int. So we can use **combination formula**:

$$C_n^m = C_{n-1}^{m-1} + C_{n-1}^m$$

to calculate the combination number. And the specific implementation code can be seen in **calculateCombination**.

```
int calculateCombinationNumber(int n,int m){
    for(int i=0;i<=n;i++)
        C[i][0]=1;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=i;j++)
            C[i][j]=C[i-1][j-1]+C[i-1][j];
    return C[n][m];
}
```

Basic Ideas

Greedy Algorithm to Calculate the Set Coveraged

We denote that the input is a set $\mathcal{U}$ of n elements, and a collection $S = \{S_1, S_2, ..., S_m\}$ of m subsets of $\mathcal{U}$ such that $\cup_i S_i = \mathcal{U}$. Our goal is to take as few subsets as possible from S such that their union covers $\mathcal{U}$. We can solve this problem easily by greedy algorithm. The algorithm is below in Table 3:

Greedy Cover(S,$\mathcal{U}$)
1. repeat
2. pick the set that covers the maximum number of uncover element
3.mark elements in the chosen set as covered
4.remove the set from S to the result set
5. done

Table 3

Basic Ideas

Based on the three lemmas above, we can easily transform the problem to that the set $\mathcal{U} = \{1, 2, \cdots, C_n^j\}$, which means that we map each different subset whose the number of the elements is j to a unique code from 1 to C_n^j. Each subset of S, represents the each k set's subsets whose number of elements is j. Ultimately, we can solve the problem easily.

Basic Ideas

When $j \neq s$:
The way to solve the problem is just like the way we mentioned above. However, after finishing finding the subset of the k set whose element number is s, we should know how many sets whose the number of elements is j include it. Therefore, we use **DFS(depth first search)** to find out them.
Assuming that $n = 5, s = 3, j = 4$, and the subset whose number of elements is equal to 3 is labeled as 01011_2. Therefore, we can expand it as below in Table 4.

Basic Ideas

5	4	3	2	1
0	1	0	1	1
0	1	1	1	1
1	1	0	1	1

Table 4

Then, we should mark the last two rows of the set above in the $\mathcal{U}$ as covered.

Essential Codes and Functions Analysis

Realization of Modifying DB files

As the request said, we need output the group of k samples and corresponding result in DB files.

First of all, we choose an OOP program language **C#** which runs on**. Net framework** and**. Net core**(completely open source, cross platform) to help realize combine with modifying DB files.

Depending on **C#** powerful library and interface, we can apply our algorithm source code on GUI platform, and realizing the operation of creating new files(Code.1) as well as exporting result into corresponding files(Code.2).

Essential Codes and Functions Analysis

```
1  public void CreateTableInToMdb(string fileNameWithPath)
2  {
3      try
4      {
5          OleDbConnection myConnection = new OleDbConnection
6              ("Provider=Microsoft.Jet.OLEDB.4.0; Data Source="
7                  + fileNameWithPath);
8          myConnection.Open();
9          OleDbCommand myCommand = new OleDbCommand();
10         myCommand.Connection = myConnection;
11         myCommand.CommandText =
12         "CREATE TABLE my_table([m] NUMBER," +
13             "[n] NUMBER, [k] NUMBER, [j] Number," +
14                 "[s] NUMBER, [n numbers] TEXT," +
15                     "[minium number of sets] NUMBER,"+
16                         "[answer] TEXT)";
17         myCommand.ExecuteNonQuery();
18         myCommand.Connection.Close();
19     }
20     catch { }
21 }
```

Essential Codes and Functions Analysis

```
1 public void InsertToMdb(string fileNameWithPath)
2 {
3     var con = new OleDbConnection(
4         "Provider = Microsoft.Jet.OLEDB.4.0; Data Source = "
5             + fileNameWithPath);
6     var cmd = new OleDbCommand();
```

Essential Codes and Functions Analysis

```
7       cmd.Connection = con;
8       cmd.CommandText = "insert into my_table ([m],[n],[k],[j]," +
9           "[s],[n numbers],[minium number of sets], [answer])" +
10              "values (@m, @n, @k,@j,@s,@series1 , @number, @answer);";
11      cmd.Parameters.AddWithValue("@m", numericUpDown1.Value);
12      cmd.Parameters.AddWithValue("@n", numericUpDown2.Value);
13      cmd.Parameters.AddWithValue("@k", numericUpDown3.Value);
14      cmd.Parameters.AddWithValue("@j", numericUpDown4.Value);
15      cmd.Parameters.AddWithValue("@s", numericUpDown5.Value);
16      cmd.Parameters.AddWithValue("@series1", series1Fordb());
17      cmd.Parameters.AddWithValue("@number", vs.Count());
18      cmd.Parameters.AddWithValue("@answer", series2Fordb());
19      con.Open();
20      cmd.ExecuteNonQuery();
21      con.Close();
22  }
```

Essential Codes and Functions Analysis

Multi-Threading

We adopt multi-threading programming way. We split the program into two parts, which are the GUI part and the calculation part. In this way, even if the program haven't figured out, the window of the program won't be stick. The specific implemented function is bound in **button2_Click**.

Essential Codes and Functions Analysis

```
1  private async void button2_Click(object sender, EventArgs e)
2  // Run buttom
3  {
4      button2.Enabled = false;
5      Algorithm algorithm = new Algorithm(
6          (int)numericUpDown2.Value,
7              (int)numericUpDown3.Value,
8                  (int)numericUpDown4.Value,
9                      (int)numericUpDown5.Value,
10                         totalList, judgeNumber);
11     if (numericUpDown4.Value == numericUpDown5.Value)
12     {
13         vs= await Task.Run(()=>algorithm.ExecuteAlgorithm1());
14     }
15     else
```

Essential Codes and Functions Analysis

```
16      {
17          vs = await Task.Run(()=>algorithm.ExecuteAlgorithm2());
18      }
19      //InsertToMdb(openFileDialog1.FileName);
20      //UpdateToMdb(openFileDialog1.FileName);
21      textBox3.Text = GetSeries2();
22      //textBox3.Enabled = false;
23
24
25  }
```

Program Test

The detailed steps that teach how to use program have been in the report. We just show the result in today's presentation.
If the program window (Figure 1)can be displayed normally, you can enter the value for verification. The conditions of 1, 2, 3 and 4, 5 and 6, 7 in the project requirement file are similar, so we choose 1(Figure 2), 4(Figure 3), and 6(Figure 4) as the demo of our program.

Program Test

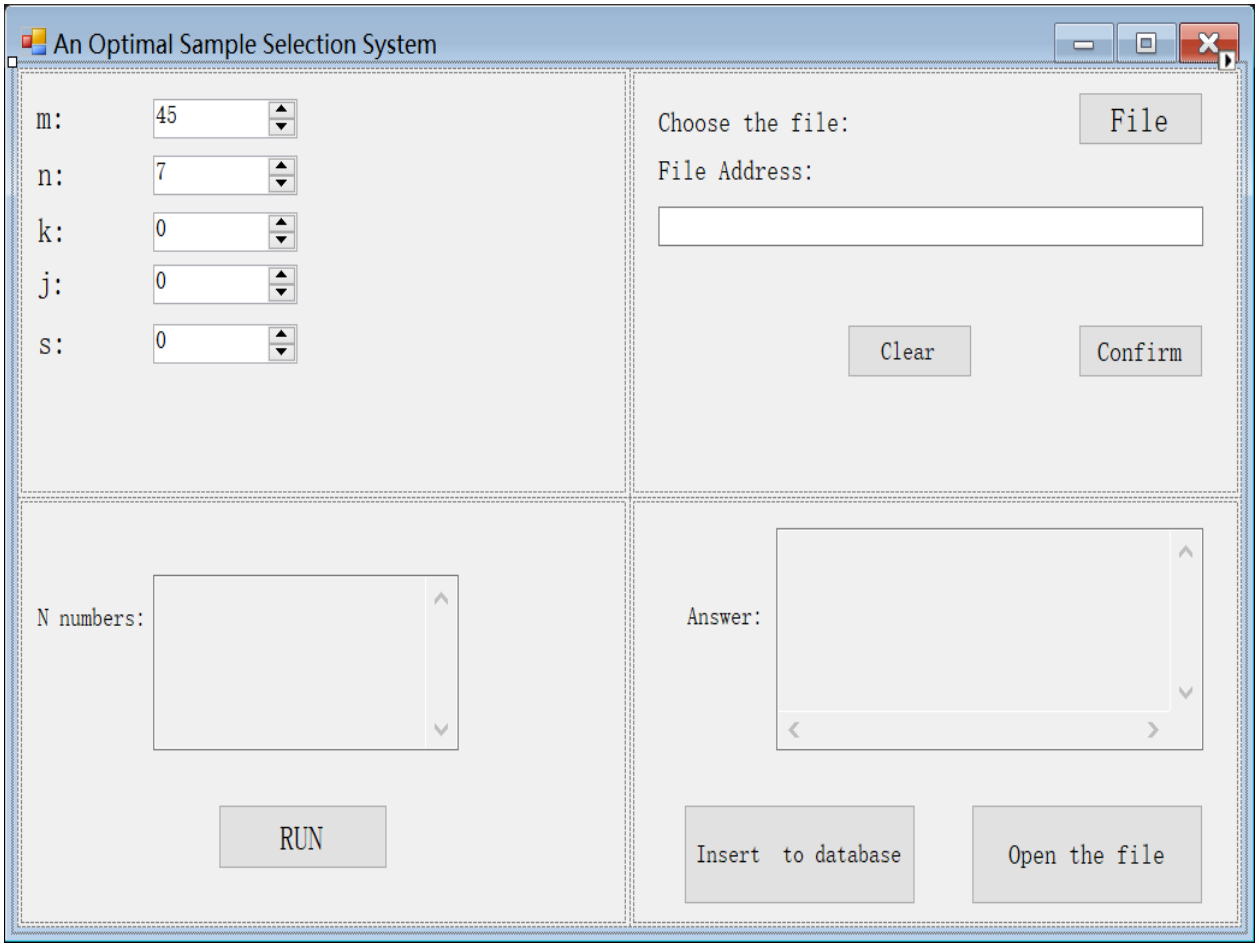

Figure 1: initial program window

Program Test

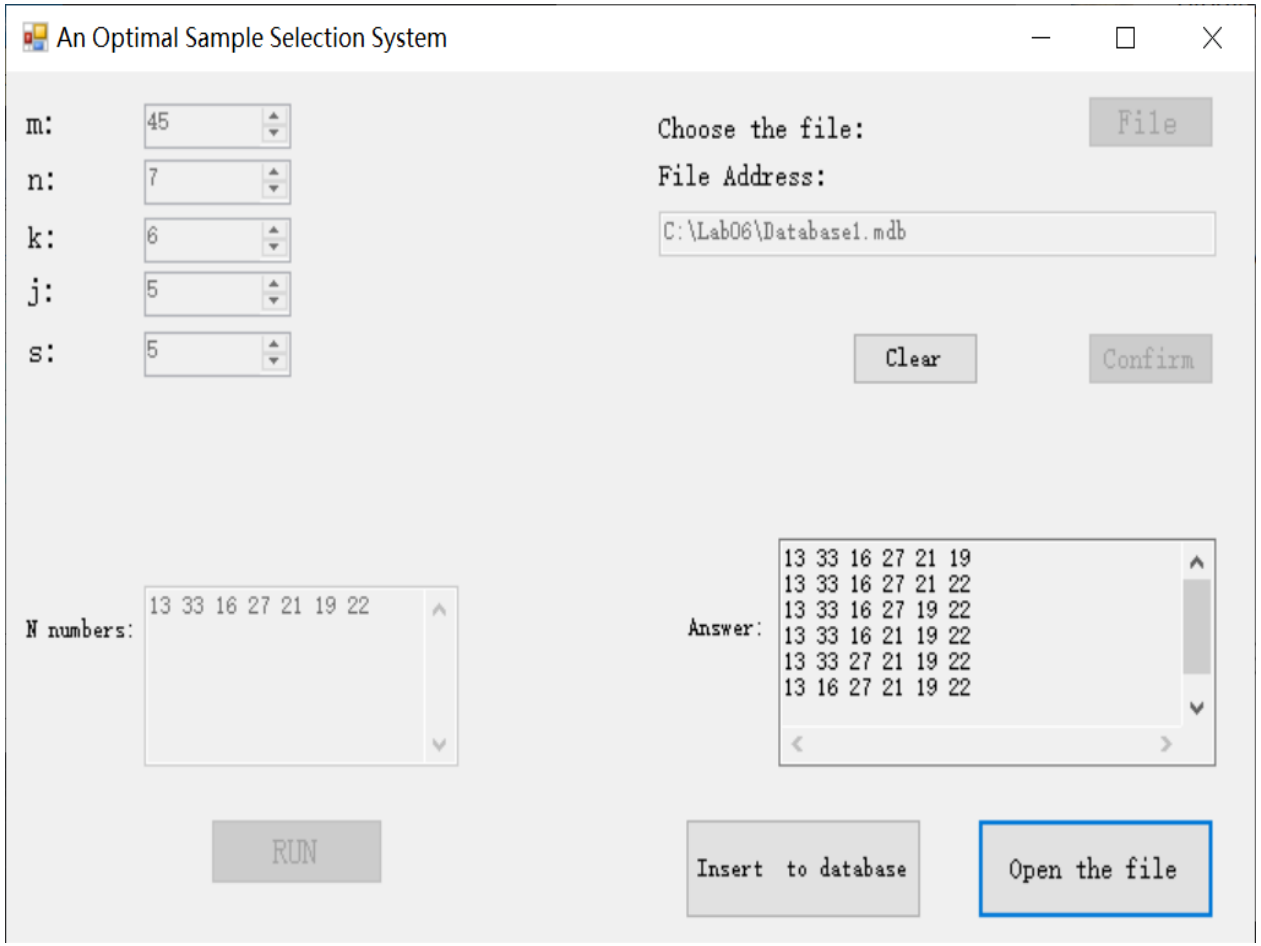

Figure 2: E.g.1: Input the data: $m = 45, n = 7, k = 6, j = 5, s = 5$.

Program Test

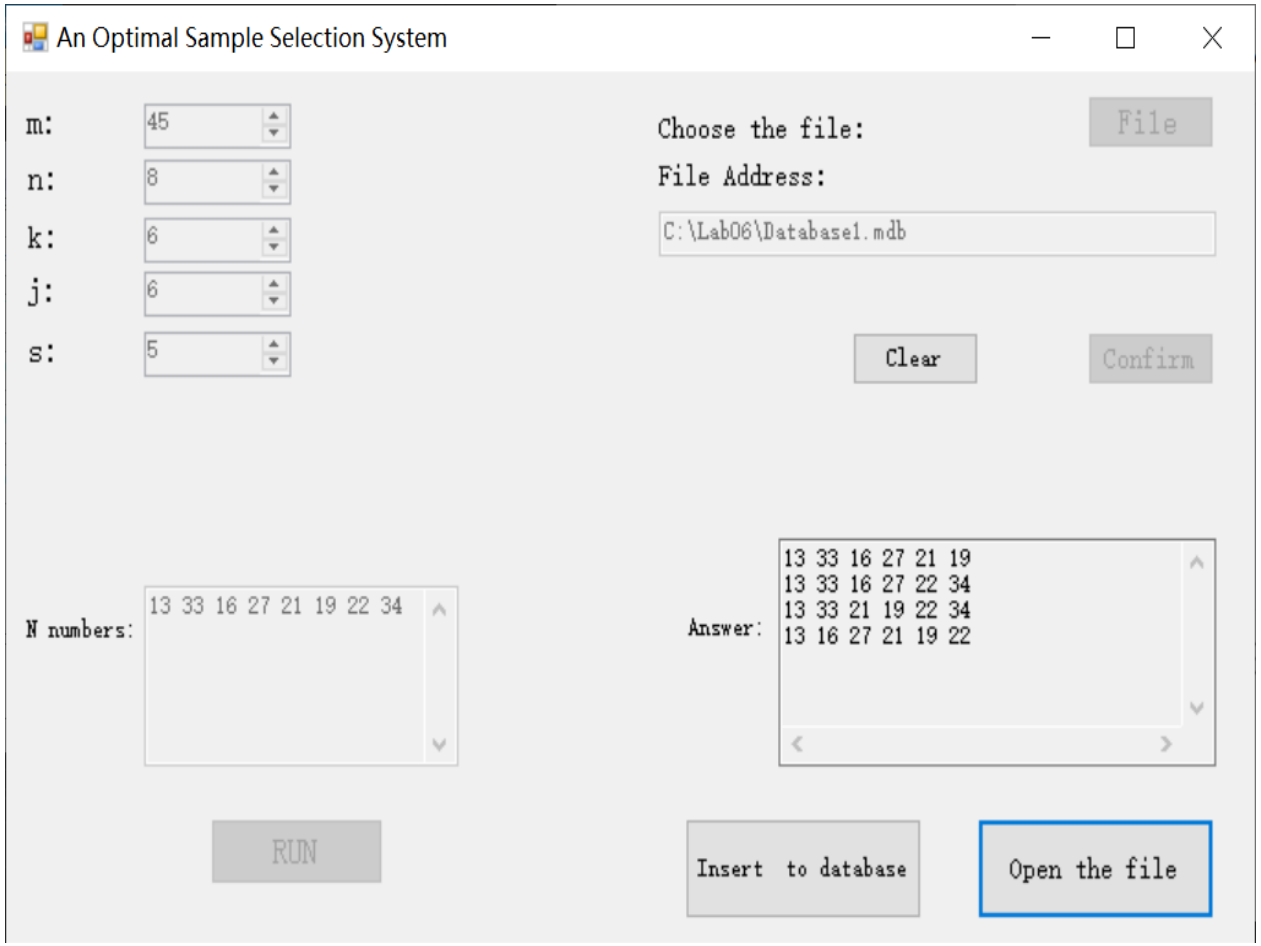

Figure 3: E.g.4: Input the data: $m = 45, n = 8, k = 6, j = 6, s = 5$.

Program Test

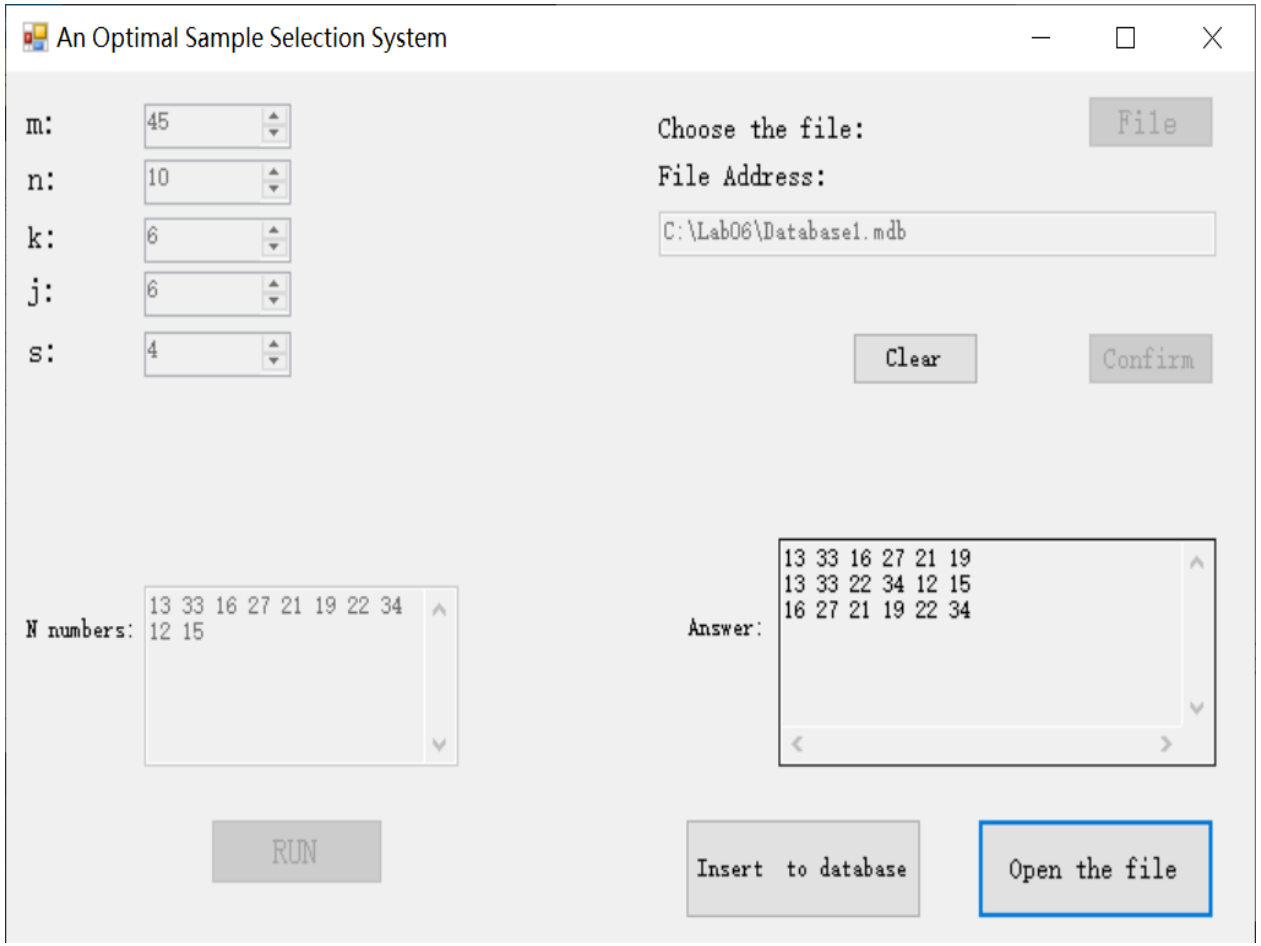

Figure 4: E.g.6: Input the data: $m = 45, n = 10, k = 6, j = 6, s = 4$.

Program Test

On-site test: The teacher can select values for random parameters, and we will test on site

Summary

This project is based on the theoretical direction of **ME001** subject and combines some knowledge of data structure and mathematic, including optimal algorithms and combinatorics. But there are no correct understanding of some part of difficult and profound mathematic problems like fuzzy set. Authors point out about converse decimal digits to the binary make the big data abstraction in order to descend the time complexity. Besides, authors have already comprehend the core of program language **C#** with utilizing **. Net Framework**. By using program, authors realize the process from theory to practice reflecting the theoretical view of unity of knowledge and practice. When writing large-scale projects, people often need cooperation and collaborative development, authors use **GitHub** for collaborative development and submit own patch code to collaborator's repository. In this article, we will utilize ideas to achieve team cooperation on **GitHub**.

Summary

The last but not least, the goal of the future study and work is to work harder to learn this knowledge, in order to enrich, improve our level.