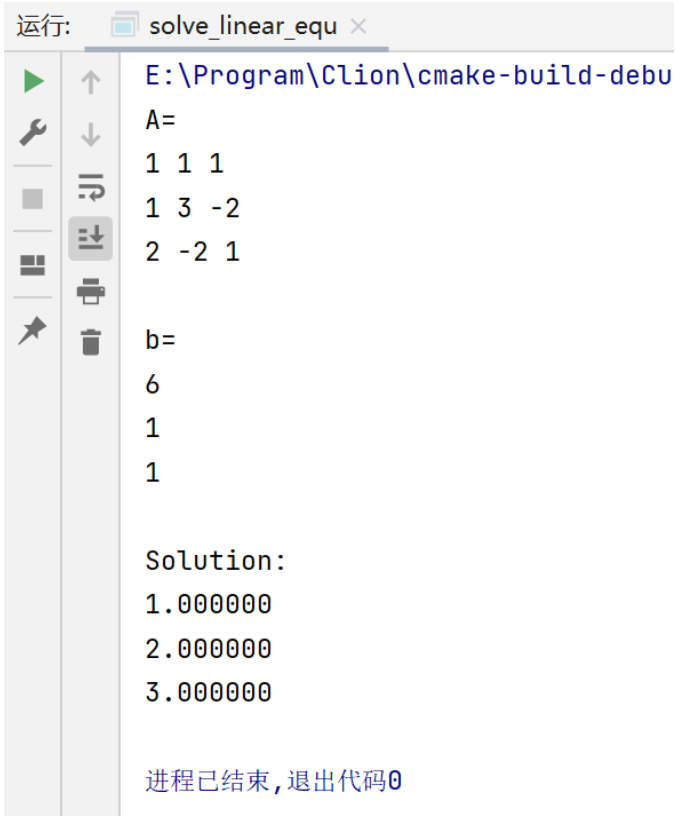
一、解线性方程组：

```
#include <iostream>
#include <fstream>
using namespace std;

int readData(char *filename, float** &A, float* &b);
void displayData(float** &A, float* &b,int N);
bool solve(float** &A, float* &b,float* &x,int N);

int main()
{
   char fileName[] = "E:\\Program\\Clion\\solve_linear_equ\\data.txt";
   float **A = nullptr, *b = nullptr;
   int N;
   //读取数据
   N = readData(fileName, A, b);
   if (N < 1)
   {
      printf("Wrong data.\n");
      return 1;
   }
   //展示数据
   displayData(A, b, N);
   //求解方程组
   float* x = new float[N];
   bool ok = solve(A,b,x,N);

   if (ok)
   {
      printf("\nSolution:\n");
      for (int i = 0; i < N; i++)
         printf("%f\n", x[i]);
   }
   else
      printf("Solution Failure!\n");

   //释放内存
   for(int i = 0; i < N; i++)
      delete[] A[i];
   delete[] A;
   delete[] b;
   delete[] x;
   return 0;
}
```

```
int readData(char *filename, float** &A, float* &b)
{
   fstream fs;
   fs.open(filename, ios::in);
   //读取矩阵阶数
   int N;
   fs >> N;
   //创建相应的动态数组
   A = new float* [N];
   for(int i = 0; i < N; i++)
      A[i] = new float [N];
   b = new float [N];
   //读取并存储矩阵
   for(int i = 0; i < N; i++)
   {
      for(int j = 0; j < N; j++)
         fs >> A[i][j];
      fs >> *(b+i);
   }

   fs.close();
   return N;
}


void displayData(float** &A, float* &b,int N)
{
   cout << "A=" << endl;
   for(int i = 0; i < N; i++)
   {
      for(int j = 0; j < N; j++)
         cout << A[i][j] << " ";
      cout << endl;
   }

   cout << endl;

   cout << "b=" << endl;
   for(int i = 0; i < N; i++)
      cout << b[i] << endl;
}
```

```
bool solve(float** &A, float* &b, float* &x,int N)
{
   //消元
   float mik;
   for (int k=0; k<N - 1; k++)
   {
      if (!A)
         return false;
      for (int i=k+1; i<N; i++)
      {
         mik = A[i][k] / A[k][k];
         for (int j=k; j<N; j++)
         {
            A[i][j] = A[i][j] - mik*A[k][j];
         }
         b[i]= b[i] - mik*b[k];
      }
   }
   //回代
   float S;
   x[N - 1] = b[N - 1] / A[N - 1][N - 1];
   for (int k=N - 2; k>=0; k--)
   {
      S=b[k];
      for (int j=k+1; j<N; j++)
      {
         S=S - A[k][j] * x[j];
      }
      x[k] = S / A[k][k];
   }

   return true;
}
```

运行结果：

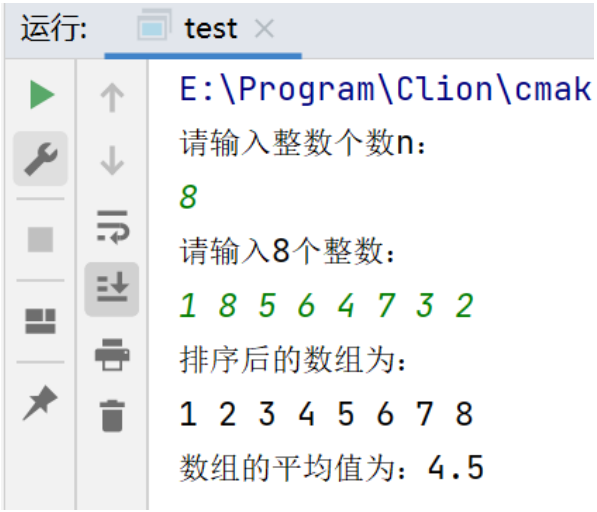


二、数组排序：

```
#include<iostream>
using namespace std;

void sort_array(int * arr, int n);
float average_array(int * arr, int n);

int main()
{
   int n;
   cout << "请输入整数个数 n：" << endl;
   cin >> n;

   int* num = new int [n];
   cout << "请输入" << n << "个整数：" << endl;
   for (int i = 0; i < n; i++)
      cin >> num[i];

   sort_array(num, n);
   cout << "排序后的数组为：" << endl;
   for (int i = n - 1; i >= 0; i--)
      cout << num[i] << " ";
   cout << endl;

   float average = average_array(num, n);
   cout << "数组的平均值为：" << average << endl;
   return 0;
}

void sort_array(int* arr, int n)
{
   for(int i = n; i >= 0; i--)
   {
      for(int j = 0; j < i; j++)
      {
         if (*(arr + j) < *(arr + j + 1))
         {
            int temp = *(arr + j);
            *(arr + j) = *(arr + j + 1);
            *(arr + j + 1) = temp;
         }
      }
   }
}
```

```
float average_array(int * arr, int n)
{
   float sum = 0;
   for(int i = 0; i < n; i++)
      sum += arr[i];
   return sum/n;
}
```

运行结果：



三、分数登记：

直接做的第三题，先输出所有合格同学的信息，后输出所有不合格同学的信息

```
# include <iostream>
# include <fstream>
# include <cstdlib>
# include <iomanip>
using namespace std;

struct Student{
   char* name;
   char* stu_id;
   char* sort;
   int fail;
};

int main()
{
   int n;
   cout << "请输入学生个数 n : " << endl;
   cin >> n;
   cin.sync();
   //为变量分配内存
   Student* student = new Student [n];
   for(int i = 0; i < n; i++)
   {
      student[i].name = new char[10]; //姓名最多十个字符
      student[i].stu_id = new char[5];
      student[i].sort = new char[5];
   }
```

```
    for(int i = 0; i < n; i++)
    {
        //获取数据
        string A;
        cout << "请输入第" << (i+1) << "个同学的名字、学号和分数：" << endl;
        getline(cin, A);
        //存储数据（三项分别保存在不同变量中）
        istringstream str(A);
        str >> student[i].name >> student[i].stu_id >> student[i].sort;

        //判断是否及格
        // 0 表示及格； 1 表示不及格
        //atof 函数用于将字符数组转为浮点数
        if(atof(student[i].sort) < 60.0 )
            student[i].fail = 1;
        else
            student[i].fail = 0;
    }

    //存入文件中
    const char* filename = "E:\\Program\\Clion\\score_registration\\data.txt";
    fstream fs;
    fs.open(filename, ios::out);

    fs << "合格名单" << endl;
    fs << "姓名      学号      分数" << endl;
    for(int i = 0; i < n; i++)
    {
        if(student[i].fail == 0)
            fs << left << setw(10) << student[i].name
               << setw(10) <<  student[i].stu_id
               << setw(10) << student[i].sort << endl;
    }

    fs << "不合格名单" << endl;
    fs << "姓名      学号      分数" << endl;
    for(int i = 0; i < n; i++)
    {
        if(student[i].fail == 1)
            fs << left << setw(10) << student[i].name
               << setw(10) <<  student[i].stu_id
               << setw(10) << student[i].sort << endl;
    }

    return 0;
}
```

运行结果：


运行:
score_registration
E:\Program\Clion\cmake-build-debug\
请输入学生个数 n ：
5
请输入第1个同学的名字、学号和分数：
A 1001 53
请输入第2个同学的名字、学号和分数：
B 1002 78
请输入第3个同学的名字、学号和分数：
C 1003 85
请输入第4个同学的名字、学号和分数：
D 1004 100
请输入第5个同学的名字、学号和分数：
E 1005 23
进程已结束,退出代码0


data.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
合格名单
姓名 学号 分数
B 1002 78
C 1003 85
D 1004 100
不合格名单
姓名 学号 分数
A 1001 53
E 1005 23