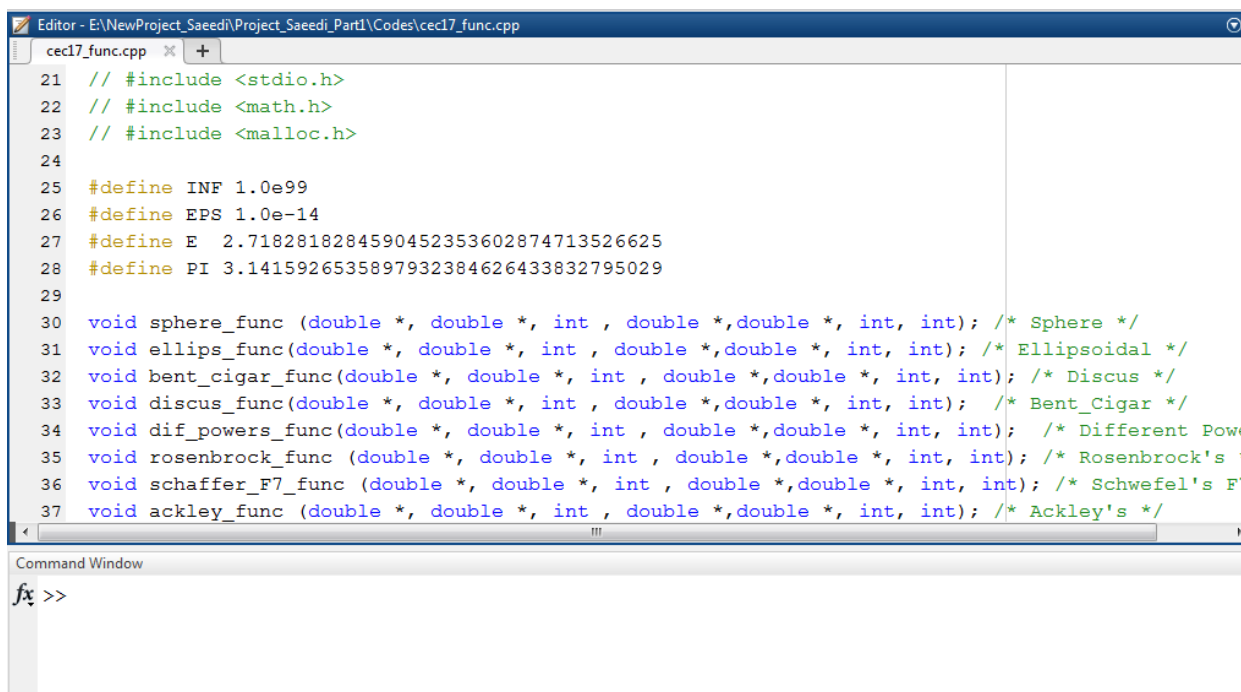


برای ارزیابی الگوریتم پیشنهادی و سایر الگوریتم‌های مورد مقایسه، از ۳۰ تابع بنچمارک ۲۰۱۷ یا CEC2017 استفاده کردیم. توابع به صورتی که کد C++ پیاده سازی شدن. هربار یکی از این توابع فراخوانی میشه و برای حلش از الگوریتم‌ها استفاده میکنیم. این دیتاست رو به راحتی میشه از نت دانلود کرد. شکل الف تصویری از کد این توابع رو نشون میده.



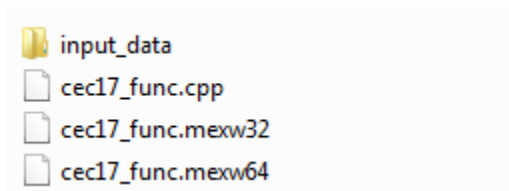
```
Editor - E:\NewProject_Saeedi\Project_Saeedi_Part1\Codes\cec17_func.cpp
cec17_func.cpp
21 // #include <stdio.h>
22 // #include <math.h>
23 // #include <malloc.h>
24
25 #define INF 1.0e99
26 #define EPS 1.0e-14
27 #define E 2.7182818284590452353602874713526625
28 #define PI 3.1415926535897932384626433832795029
29
30 void sphere_func (double *, double *, int , double *,double *, int, int); /* Sphere */
31 void ellips_func(double *, double *, int , double *,double *, int, int); /* Ellipsoidal */
32 void bent_cigar_func(double *, double *, int , double *,double *, int, int); /* Discus */
33 void discus_func(double *, double *, int , double *,double *, int, int); /* Bent-Cigar */
34 void dif_powers_func(double *, double *, int , double *,double *, int, int); /* Different Powe
35 void rosenbrock_func (double *, double *, int , double *,double *, int, int); /* Rosenbrock's *
36 void schaffer_F7_func (double *, double *, int , double *,double *, int, int); /* Schwefel's F7
37 void ackley_func (double *, double *, int , double *,double *, int, int); /* Ackley's */
```

شکل الف: بخشی از کد توابع بنچمارک ۲۰۱۷

بعد از اینکه نتایج ۳۰ تابع بنچمارک رو برای هر الگوریتم بدست آوردیم، الگوریتم‌ها رو از طریق تست آماری فریدمن، رتبه بندی میکنیم، الگوریتمی بهتر عمل کرده که رتبه اش بهتر شده باشه.

خب حالا بریم سراغ محتویات داخل فایل. در این پوشه، کدهای زیر رو داریم:

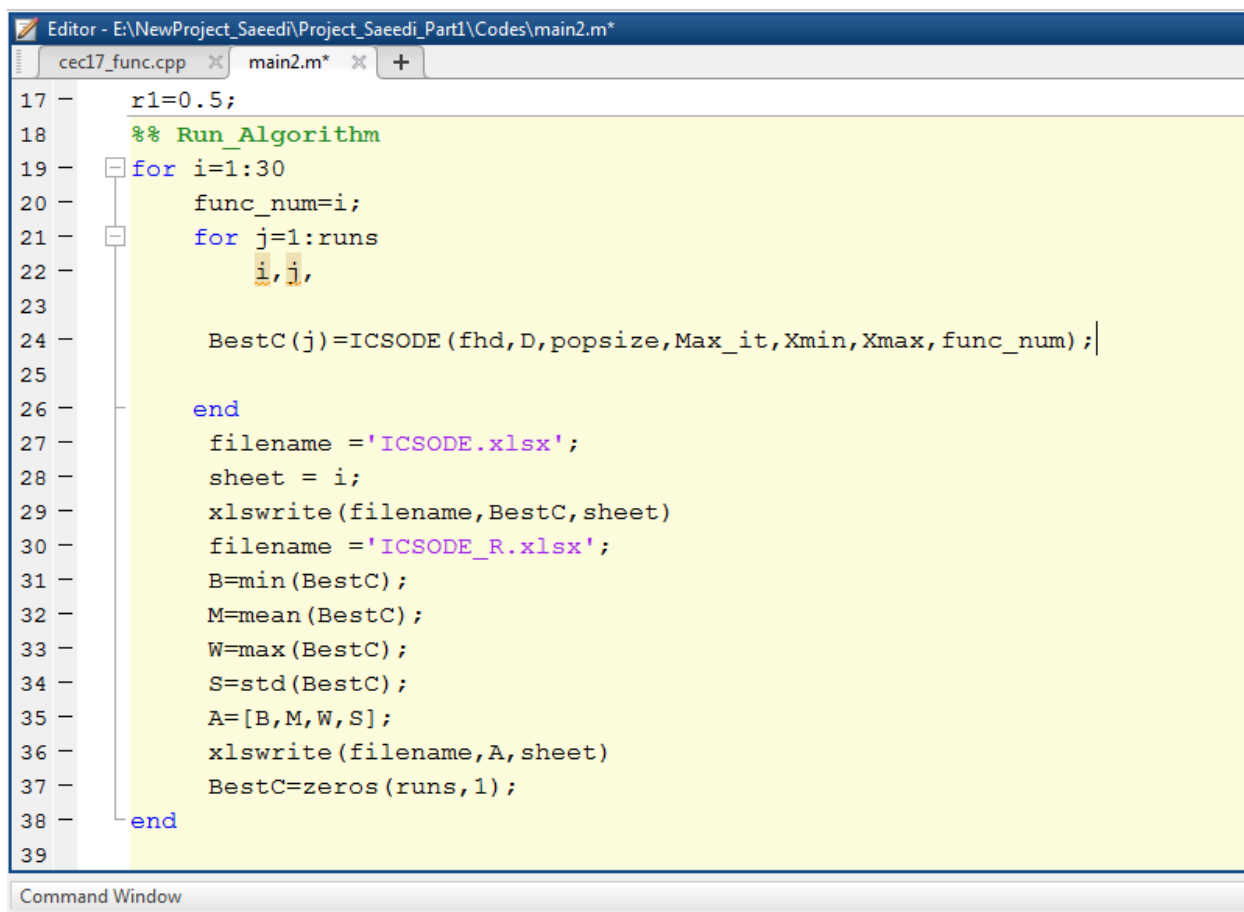
اول، فایل‌های مربوط به دیتاست رو داریم، داخل پوشه‌ی input_data داده‌های مربوط به شیف و چرخش رو داریم، این دیتاست اومده مسائل ریاضی که معمولا NP-hard هستن رو با حالت شیف و چرخش، سخت ترشون کرده. بعد از اون، فایل C++ رو داریم که کدشو تو شکل الف دیدیم. بعد دوتا کد دیگه که مربوط به متلب میشه و خودم هم نمیدونم دقیقا چی هستن، کلا لازم نیست بدونیم اینا چی هستن



شکل ب: فایل‌های مربوط به دیتاست CEC2017

کد "main2.m" شامل کدی برای اجرای خودکار الگوریتم‌ها جهت گرفتن نتایج هست. در واقع این کد با تنظیم پارامترهای الگوریتم‌ها از قبیل تعداد اعضای جمعیت، حداکثر تعداد تکرار و غیره، هر الگوریتم را فراخوانی کرده و به تعداد مشخصی از آن اجرا

میگیرد و نتایج را در هر بار اجرا، در یک فایل اکسل ذخیره می‌نماید. در نهایت از این فایل اکسل برای تعیین "بهترین نتیجه (Best)"، "بدترین نتیجه (Worst)"، "میانگین (Mean)" و "انحراف استاندارد (Std.)" استفاده میکنیم و نتیجه‌ی این معیارها را در فایل اکسل دوم قرار می‌دهیم (شکل ج).



```
Editor - E:\NewProject_Saeedi\Project_Saeedi_Part1\Codes\main2.m*
cecl7_func.cpp  main2.m*  +
17 - r1=0.5;
18 - %% Run_Algorithm
19 - for i=1:30
20 -     func_num=i;
21 -     for j=1:runs
22 -         i,j,
23 -
24 -         BestC(j)=ICCODE(fhd,D,popsize,Max_it,Xmin,Xmax,func_num);
25 -
26 -     end
27 -     filename='ICCODE.xlsx';
28 -     sheet = i;
29 -     xlswrite(filename,BestC,sheet)
30 -     filename='ICCODE_R.xlsx';
31 -     B=min(BestC);
32 -     M=mean(BestC);
33 -     W=max(BestC);
34 -     S=std(BestC);
35 -     A=[B,M,W,S];
36 -     xlswrite(filename,A,sheet)
37 -     BestC=zeros(runs,1);
38 - end
39
```

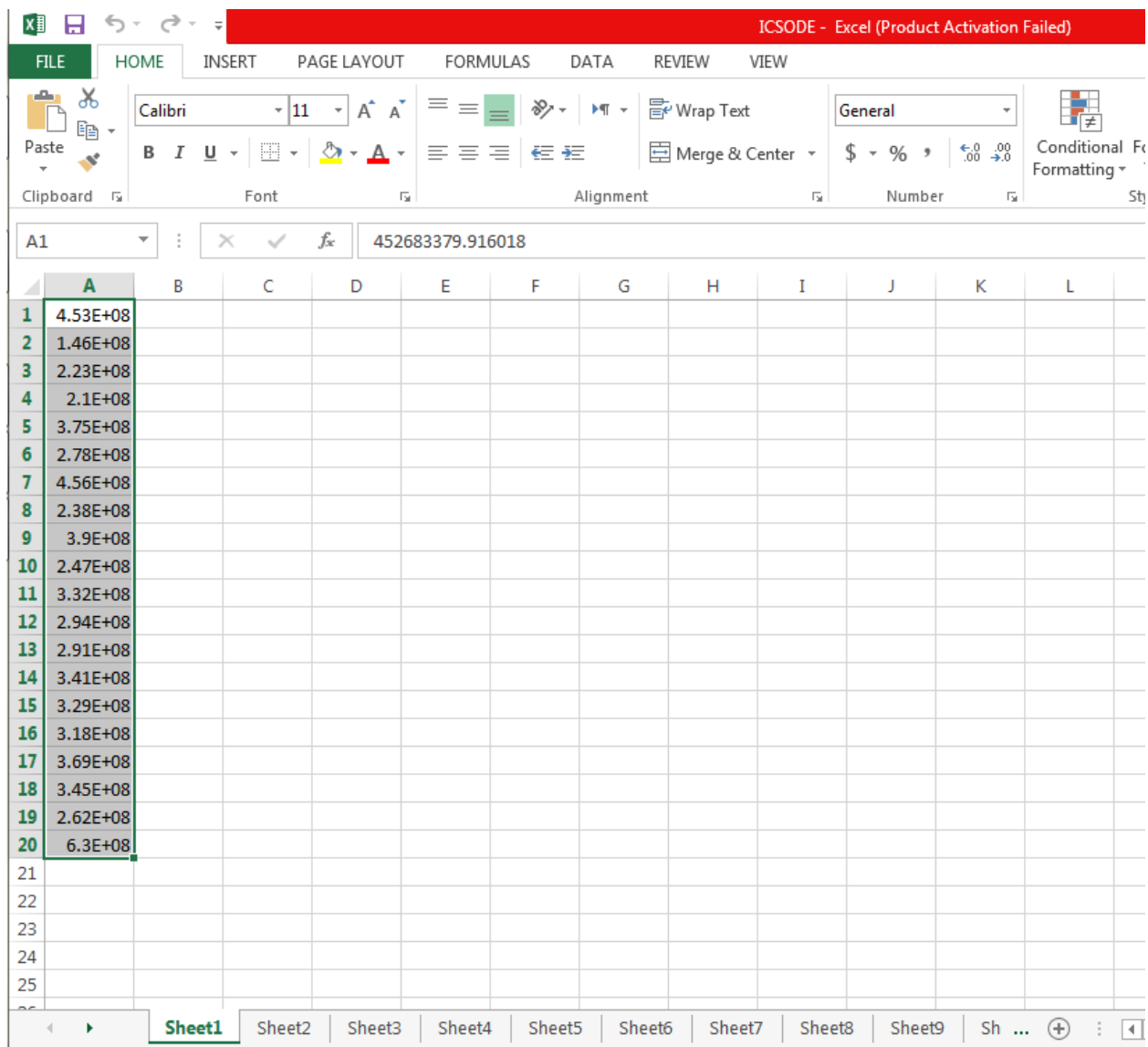
Command Window

شکل ج: کد main2 و ایجاد فایل‌های اکسل در آن برای ذخیره کردن نتایج اجرا

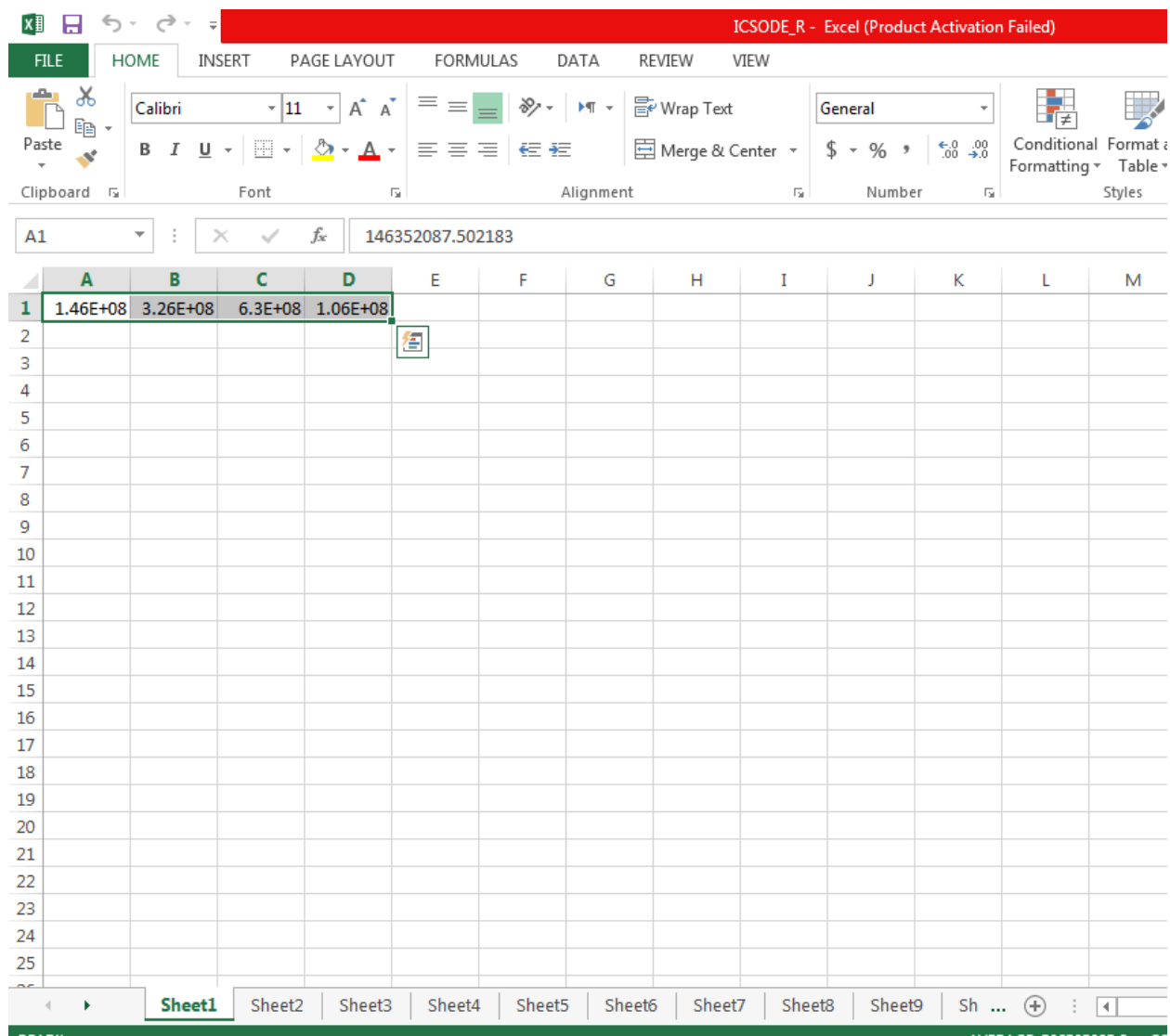
همانطور که در شکل د مشاهده می‌شود، فایل اکسل اول که در اینجا برای الگوریتم ICSODE ساخته شده، شامل ۳۰ تا Sheet هست. هر Sheet برای یکی از توابع بنچمارک در نظر گرفته شده. مثلاً Sheet1 نتایج ۲۰ بار اجرای مستقل الگوریتم را برای تابع شماره ۱ نشان می‌دهد.

همچنین، در فایل اکسل دوم که در اینجا برای الگوریتم ICSODE ساخته شده، شامل ۳۰ تا Sheet هست. هر Sheet برای یکی از توابع بنچمارک در نظر گرفته شده. مثلاً Sheet1 به ترتیب شامل بهترین نتیجه (Best)، میانگین نتایج (Mean)، بدترین نتیجه (Worst) و انحراف استاندارد (Std) برای ۲۰ بار اجرای مستقل الگوریتم را برای تابع شماره ۱ نشان می‌دهد (شکل ز).

۱- کد "freidman" شامل کدی برای انجام تست آماری فریدمن هست (شکل خ). نتایج این تست، در یک فایل اکسل ذخیره میشه که اسم این فایل رو به دلخواه "Friedman_Results" قرار دادیم. این فایل هم شامل ۳۰ تا sheet هست، که هر sheet بیانگر نتایج رتبه بندی الگوریتم‌ها برای یک تابع است.



شکل د: نتایج موجود در فایل اکسل اول در کد main2



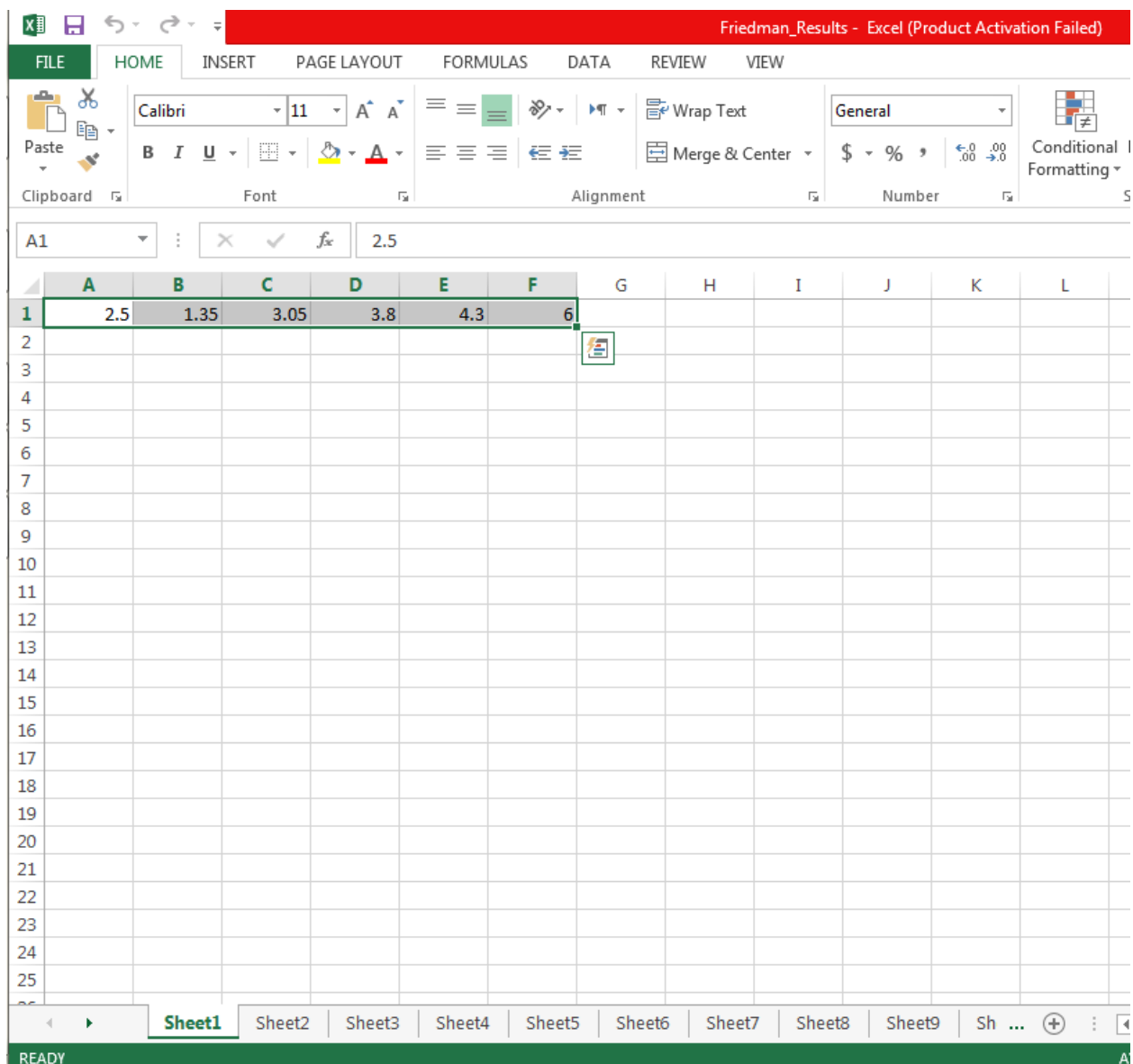
شکل ز: نتایج موجود در فایل اکسل دوم در کد main2

```
Editor - E:\NewProject_Saeedi\Project_Saeedi_Part1\Codes\freidman.m
cec17_func.cpp  main2.m  freidman.m  +
2 - clear;
3 - close all;
4 - %% load data
5 - for i=1:30
6 -
7 -     sheet = i;
8 -     CSO = xlsread('CSO.xlsx',sheet);
9 -     ICSODE = xlsread('ICSODE.xlsx',sheet);
10 -    ICSO = xlsread('ICSO.xlsx',sheet);
11 -    ICSO_Chaoitic = xlsread('CSO_Chaoitic.xlsx',sheet);
12 -    ICSO_CrossOver = xlsread('CSO_CrossOver.xlsx',sheet);
13 -    BAT = xlsread('bat.xlsx',sheet);
14 -
15 -    Friedman1=[CSO';ICSODE';ICSO';ICSO_Chaoitic';ICSO_CrossOver';BAT'];
16 -
17 -    [p,tbl,stats] = friedman(Friedman1,1,'off');
18 -    MRank=stats.meanranks(1:end);
19 -    % fprintf(fid2, '%.4g',MRank);
20 -    filename = 'Friedman_Results.xlsx';
21 -    xlswrite(filename,MRank,sheet)
22 -
23 - end
24
```

Command Window

شکل خ: کد مربوط به تست فریدمن

دقت کن تو کد فریدمن داریم از فایل اکسل اول که شامل نتایج اجراهاست استفاده میکنیم، این تست، براساس میانگین کار میکنه (به تعریف درست و حسابی که تونستم ازش پیدا کنم و تو داکيومنت خودم هم بنویسم این بود: "آزمون فریدمن، یکی از آزمون‌های نا پارامتری آماری است که برای مقایسه چندین گروه وابسته کاربرد دارد و با توجه به میانگین، رتبه‌های گروه را مشخص می‌کند"). نتایج رتبه بندی الگوریتم‌ها رو در فایل اکسل در شکل چ مشاهده میشه. در شکل چ، نتایج از چپ به راست به ترتیب مربوط به رتبه الگوریتم CSO، ICSODE، ICSO، ICSO_Chaoitic، ICSO_CrossOver، BAT می‌باشد.



شکل چ: نتایج رتبه بندی الگوریتم ها براساس تست فریدمن