表計算　課題

2013

if関数等の活用とアイデア

（Ａ）自然数を入力しよう　⇒　3

E4の数値を2で割った余り⇒　1

偶・奇を判定します　⇒　奇数　　=IF(MOD(E4,2)=1,"奇数","偶数")

（Ｂ）一様乱数⇒　0.94000934

（Ｃ）おみくじ
F9を押してください。結果⇒　大吉　　=IF(RAND()>0.5,"大吉","中吉")

（Ｄ）大吉・中吉・凶？　⇒　大吉　応用　　=IF(E10>0.4,IF(E10>0.7,"大吉","中吉"),"凶")

（Ｅ）励ましの言葉　⇒
よかったね　応用　　=IF(E15="凶","またいいこともあるよ","よかったね")

【課題：if関数等の活用とアイデア】

(A) エクセルで余りを求める関数 **=MOD(被除数,除数)** を使って，Ｅ４セルを２で割った余りをＥ６セルに表示させなさい。
関数 **=IF(条件,真の場合の処理,偽の場合の処理)** を用いて，Ｅ８セルに入力された数値の偶奇を表示させなさい。

(B) Ｅ４セルに乱数を返す関数 **=RAND()** を正しく記述しなさい。
RANDの括弧内は何も書かないこと。
Ｆ９キーを押して，Ｅ４の数値が0以上1未満で変化することを確かめなさい。

(C) IF関数とRAND関数を用いて，**F9キー**を押すと[大吉]か[中吉]が表示されるようにしなさい。

【応用：早くできた者】
(D)　ＩＦ関数を入れ子にして２個使い，（Ａ）を利用して
[大吉]か[中吉]か[凶]を表示させなさい。例えば，
=IF(**条件**,IF(**内側IF条件**,**内側IF真**,**内側IF偽**),**外IF偽**)
内側のIFの前にイコールは不要。

(E)　（Ｄ）で[凶]がでた時だけ，励ましの言葉を表示させよう。
それ以外の時には，他の言葉を表示させてもよい。

正の平方根の計算

2　　1.414213562

回数	横	縦	面積（縦×横）
1	1	2	2
2	1.5	1.333333333	2
3	1.416666667	1.411764706	2
4	1.414215686	1.414211438	2
5	1.414213562	1.414213562	2
6	1.414213562	1.414213562	2
7	1.414213562	1.414213562	2
8	1.414213562	1.414213562	2
9	1.414213562	1.414213562	2
10	1.414213562	1.414213562	2
11	1.414213562	1.414213562	2
12	1.414213562	1.414213562	2
13	1.414213562	1.414213562	2
14	1.414213562	1.414213562	2
15	1.414213562	1.414213562	2
16			
17			
18			
19			
20			
21			
22			
23			
24			

【課題：正の平方根の計算】

エクセルには **=sqrt()** という，平方根を計算する関数がある。
F21セルに入力して結果を確認しなさい。

次に，相対参照と絶対参照の実習を兼ねて，正の平方根をシミュレーションしてみよう。

(1) Ｄ２３セル(長方形の横)を 1 とする。
(2) Ｅ２３セル(長方形の縦)はＥ２１の値を表示させる。
(3) Ｄ２４セルは上記縦横の平均。
(4) Ｅ２４セルは面積がＥ２１になるように。
(5) Ｆ列は，面積の確認用。

相対参照　例：E21
コピーすると相対的な位置関係が保たれる。
絶対参照　例：E21
コピーしても位置が固定化されたまま。
どちらを選ぶかは **F4キー** で決める。

上の課題が早くできた人へ

クラスに同じ誕生日の人がいない確率はおよそ何パーセントか？

人数		誕生日が全員異なる確率		同じ誕生日の人がいる確率
1	365	1	1	0.00%
2	364	0.997260274	0.997260274	0.27%
3	363	0.994520548	0.991795834	0.82%
4	362	0.991780822	0.983644088	1.64%
5	361	0.989041096	0.972864426	2.71%
6	360	0.98630137	0.959537516	4.05%
7	359	0.983561644	0.943764297	5.62%
8	358	0.980821918	0.925664708	7.43%
9	357	0.978082192	0.905376166	9.46%
10	356	0.975342466	0.883051822	11.69%
11	355	0.97260274	0.858858622	14.11%
12	354	0.969863014	0.832975211	16.70%
13	353	0.967123288	0.805589725	19.44%
14	352	0.964383562	0.776897488	22.31%
15	351	0.961643836	0.74709868	25.29%
16	350	0.95890411	0.716395995	28.36%
17	349	0.956164384	0.684992335	31.50%
18	348	0.953424658	0.653088582	34.69%
19	347	0.950684932	0.620881474	37.91%
20	346	0.947945205	0.588561616	41.14%
21	345	0.945205479	0.556311665	44.37%
22	344	0.942465753	0.524304692	47.57%
23	343	0.939726027	0.492702766	50.73%
24	342	0.936986301	0.461655742	53.83%
25	341	0.934246575	0.431300296	56.87%
26	340	0.931506849	0.40175918	59.82%
27	339	0.928767123	0.373140718	62.69%
28	338	0.926027397	0.345538528	65.45%
29	337	0.923287671	0.319031463	68.10%
30	336	0.920547945	0.293683757	70.63%
31	335	0.917808219	0.269545366	73.05%
32	334	0.915068493	0.246652472	75.33%
33	333	0.912328767	0.225028146	77.50%
34	332	0.909589041	0.204683135	79.53%
35	331	0.906849315	0.185616761	81.44%
36	330	0.904109589	0.167817894	83.22%
37	329	0.901369863	0.151265992	84.87%
38	328	0.898630137	0.135932179	86.41%
39	327	0.895890411	0.121780336	87.82%
40	326	0.893150685	0.108768819	89.12%
41	325	0.890410959	0.096848389	90.32%
42	324	0.887671233	0.085969528	91.40%
43	323	0.884931507	0.076077144	92.39%
44	322	0.882191781	0.067114631	93.29%
45	321	0.879452055	0.059024101	94.10%
46	320	0.876712329	0.051747157	94.83%
47	319	0.873972603	0.045225597	95.48%
48	318	0.871232877	0.039402027	96.06%
49	317	0.868493151	0.034220391	96.58%
50	316	0.865753425	0.02962642	97.04%
51	315	0.863013699	0.025568007	97.44%
52	314	0.860273973	0.021995491	97.80%
53	313	0.857534247	0.018861887	98.11%
54	312	0.854794521	0.016123037	98.39%
55	311	0.852054795	0.013737711	98.63%
56	310	0.849315068	0.011667645	98.83%
57	309	0.846575342	0.009877541	99.01%
58	308	0.843835616	0.008335021	99.17%
59	307	0.84109589	0.007010552	99.30%
60	306	0.838356164	0.005877339	99.41%
61	305	0.835616438	0.004911201	99.51%
62	304	0.832876712	0.004090425	99.59%
63	303	0.830136986	0.003395613	99.66%
64	302	0.82739726	0.002809521	99.72%
65	301	0.824657534	0.002316893	99.77%
66	300	0.821917808	0.001904295	99.81%
67	299	0.819178082	0.001559957	99.84%
68	298	0.816438356	0.001273609	99.87%
69	297	0.81369863	0.001036334	99.90%
70	296	0.810958904	0.000840424	99.92%

【課題：同じ誕生日の人が存在しない確率】
1年は365日とする。人数50人までのクラスで確率の計算結果を表示させよう。

○ たくさんの数の積を計算する場合は，**PRODUCT()**という関数を用いるとよい。

参考：よく使われる関数の例
和 SUM()　　数値の個数 COUNT()　　平均 AVERAGE()
最大値 MAX()　　最小値 MIN()　　乱数 RAND()
分岐処理 IF()　　四捨五入ROUND()　　順位 RANK()

※ ここからは，上がはやくできた人のための，上とは別の課題です。

【課題:同じ誕生日の人が存在するかどうか，シミュレーションしよう】

○ 乱数を発生させてシミュレーション(難)

ヒント
・日付を 1～365 までの整数値と考え，乱数で誕生日を生成する。
・誕生日に順位をつけて，同じ順位があれば誕生日が同じとみなす。
・40人の順位の和が1～40の和と同じなら全員の誕生日は異なり，40人の順位の和が1～40の和より小さいなら，同じ誕生日の生徒が存在するとみなせる。

使用する関数等
・**RAND()*365** によって，0以上365未満の実数が得られる。
・**ROUNDUP(数値,0)** で，小数部分を切り上げた整数にできる。
・順位は **RANK(数値,範囲,0)** で計算できる(引数 0:降順，1:昇順)。
・IF関数で，判定を表示させることができる。
・F9を押して再計算させる。

出席番号	誕生日	順位	判定⇒　同じ誕生日あり
1	146	13	左の和⇒　815
2	149	14	
3	49	4	
4	221	29	
5	86	8	
6	215	27	
7	176	20	
8	128	12	
9	298	35	
10	175	19	
11	218	28	
12	36	1	
13	337	39	
14	349	40	
15	241	30	
16	190	24	
17	166	17	
18	330	38	
19	190	24	
20	292	34	
21	113	11	
22	180	21	
23	49	4	
24	50	6	
25	277	32	
26	109	10	
27	154	16	
28	325	37	
29	204	26	
30	189	23	
31	276	31	
32	277	32	
33	41	2	
34	149	14	
35	41	2	
36	108	9	
37	188	22	
38	79	7	
39	169	18	
40	304	36	