3. Taylor Theorem

為了找出最好的k值，我寫了以下程式來計算我用泰勒展開式所得之e與`math.h`中的`M_E`常數的差異！

```
#include <stdio.h>
#include <stdint.h>
#include <math.h>
typedef double fp;
typedef int32_t i32;
typedef int64_t i64;
fp euler_exp(i64 exp, i64 k){
  i64 fact = 1, pow = 1;
  fp sum = 1;
  for(i64 i = 1 ; i <= k ; ++i){
    fact *= i; pow *= exp;
    sum += ((fp)pow/(fp)fact);
  }
  return sum;
}
int main(){
  fp mn = 1.0;
  i64 k;
  for(i64 i = 1 ; i < 100 ; ++i){
    fp result = euler_exp(1, i);
    fp diff = fabs(result-M_E);
    if(diff < mn){
      mn = diff;
      k = i;
    }
    printf("%lld : %.1000lg\n", i, diff);
  }
  printf("%lld : %.1000lg (%.1000lg)\n", k, euler_exp(1, k), mn);
  return 0;
}
```

所得結果如下：

```
1 : 0.718281828459045090795598298427648842334747314453125
2 : 0.218281828459045090795598298427648842334747314453125
3 : 0.05161516179237857215866824844852089881896972656 25
4 : 0.009948495125712053521738198469392955303192138671875
5 : 0.00161516179237874979435218847356736660003662109375
6 : 0.0002262729034896437951829284429550170898437 5
7 : 2.7860205076724042783098411746323108673095703125e-05
8 : 3.05861777505356258188840001821517944335937 5e-06
```

```
9 : 3.02885852843104430576204322278499603271484375e-07
10 : 2.73126605776496944599784910678863525390625e-08
11 : 2.260552189881082085776142776012420654296875e-09
12 : 1.72876379878061925410293042659759521484375e-10
13 : 1.2285727990501982276327908039093017578125e-11
14 : 8.14903700074864900670945644378662109375e-13
15 : 5.0182080713057075627148151397705078125e-14
16 : 2.220446049250313080847263336181640625e-15
17 : 4.44089209850062616169452667236328125e-16
18 : 4.44089209850062616169452667236328125e-16
19 : 4.44089209850062616169452667236328125e-16
20 : 4.44089209850062616169452667236328125e-16
21 : 4.44089209850062616169452667236328125e-16
22 : 4.44089209850062616169452667236328125e-16
23 : 4.44089209850062616169452667236328125e-16
24 : 4.44089209850062616169452667236328125e-16
25 : 4.44089209850062616169452667236328125e-16
26 : 4.44089209850062616169452667236328125e-16
27 : 4.44089209850062616169452667236328125e-16
28 : 4.44089209850062616169452667236328125e-16
29 : 4.44089209850062616169452667236328125e-16
30 : 4.44089209850062616169452667236328125e-16
31 : 4.44089209850062616169452667236328125e-16
32 : 4.44089209850062616169452667236328125e-16
33 : 4.44089209850062616169452667236328125e-16
34 : 4.44089209850062616169452667236328125e-16
35 : 4.44089209850062616169452667236328125e-16
36 : 4.44089209850062616169452667236328125e-16
37 : 4.44089209850062616169452667236328125e-16
38 : 4.44089209850062616169452667236328125e-16
39 : 4.44089209850062616169452667236328125e-16
40 : 4.44089209850062616169452667236328125e-16
41 : 4.44089209850062616169452667236328125e-16
42 : 4.44089209850062616169452667236328125e-16
43 : 4.44089209850062616169452667236328125e-16
44 : 4.44089209850062616169452667236328125e-16
45 : 4.44089209850062616169452667236328125e-16
46 : 4.44089209850062616169452667236328125e-16
47 : 4.44089209850062616169452667236328125e-16
48 : 4.44089209850062616169452667236328125e-16
49 : 4.44089209850062616169452667236328125e-16
50 : 4.44089209850062616169452667236328125e-16
51 : 4.44089209850062616169452667236328125e-16
52 : 4.44089209850062616169452667236328125e-16
53 : 4.44089209850062616169452667236328125e-16
54 : 4.44089209850062616169452667236328125e-16
55 : 4.44089209850062616169452667236328125e-16
56 : 4.44089209850062616169452667236328125e-16
57 : 4.44089209850062616169452667236328125e-16
```

```
58 : 4.44089209850062616169452667236328125e-16
59 : 4.44089209850062616169452667236328125e-16
60 : 4.44089209850062616169452667236328125e-16
61 : 4.44089209850062616169452667236328125e-16
62 : 4.44089209850062616169452667236328125e-16
63 : 4.44089209850062616169452667236328125e-16
64 : 4.44089209850062616169452667236328125e-16
65 : 4.44089209850062616169452667236328125e-16
66 : inf
67 : inf
68 : inf
69 : inf
70 : inf
71 : inf
72 : inf
73 : inf
74 : inf
75 : inf
76 : inf
77 : inf
78 : inf
79 : inf
80 : inf
81 : inf
82 : inf
83 : inf
84 : inf
85 : inf
86 : inf
87 : inf
88 : inf
89 : inf
90 : inf
91 : inf
92 : inf
93 : inf
94 : inf
95 : inf
96 : inf
97 : inf
98 : inf
99 : inf
17 : 2.71828182845904553488480814849026501178741455078125
(4.44089209850062616169452667236328125e-16)
```

可見在$k=17$後就沒有任何誤差的進展了，甚至$k>65$後，因為$\frac{x^k}{k!}$太小了，所以跑出了`inf` 來

由程式可得$k=17$已是最優解了！