CP.2.3.2, Sauer3

For the n-by-n matrix with entries $a_{ij} = 1/(|i-j|-1)$, set $x = (1,\ldots,1)^T$ and $b = Ax$. Use the Use the python program from CP.2.2.2 or Numpy's **numpy.linalg.solve** command to compute **x_c**, the double precision computed solution. Find the infinity norm of the forward error and the error magnification factor of the problem $Ax = b$,and compare it with the condition number of A: (a) $n = 6$, (b) $n = 10$.

Hint: We should always be careful with Python's indexing that starts at 0, and Sauer's and Matlab's indexing that starts at 1. But here the $a_{ij} = 1/(|i-j|-1)$ leads to the same entries whether we start counting at 0 or at 1. Here is a code snippet to generate A with $n = 5$.

```
n = 5
A = np.zeros( [ n, n ], dtype=float )
for i in range(0,n):
  for j in range(0,n):
    A[i,j] = 1. / ( abs( i - j ) + 1. )
print(A)
```

```
[[1.         0.5        0.33333333 0.25       0.2        ]
 [0.5        1.         0.5        0.33333333 0.25       ]
 [0.33333333 0.5        1.         0.5        0.33333333]
 [0.25       0.33333333 0.5        1.         0.5        ]
 [0.2        0.25       0.33333333 0.5        1.         ]]
```