# InfyTQ (Longest substring with unique chars)

August 8, 2019

### 0.1 InfyTQ Problem

Find longest substring of unique characters which is case sensitive. If there are 2 or more such longest substrings, print the one that occurs first.

Optimised Solution

```
In [37]: mystr=input()
         def subString(s):
             temp=[]
             temp2=[]
             n=len(s)
             x=0
             for i in range(n):
                 temp=''.join(z for z in temp)
                 if s[i].lower() not in temp and s[i].upper() not in temp:
                     temp=list(s[x:i+1])
                 else:
                     temp=list(temp)
                     x=i
                     if len(temp)>len(temp2):
                         temp2=temp
                     temp=[]
             print(''.join(z for z in temp2))
         subString(mystr)
```

```
A@bcd1abx
A@bcd1
```

Unoptimised Solution

```
In [36]: mystr=input()
         mysubstr=[]
         startindex=0
         endindex=0
         def subString(s):
             n=len(s)
             for i in range(n):
```

```
        for x in range(i+1,n+1):
            if len(s[i:x])==len(set(s[i:x])):
                mysubstr.append(s[i:x])
                mysubstr.sort(key=len)

                if mysubstr[-1]==s[i:x] and len(mysubstr)>1:
                    if len(mysubstr[-1])!=len(mysubstr[-2]):
                        startindex=i
                        endindex=x
    print(mystr[startindex:endindex])
subString(mystr.lower())
```

```
A@bcd1abx
A@bcd1
```