

**Question:**

Peaky blinders are forming up a prime gang. The gang is been separated due to a clash. Now the peaky blinders need to reform the gang with the shortest span of time. These members are been abducted from the prime gang. So find the nearer prime gang and reform it

**Input:**

9 13

**Output:**

1

**Hint:** The gang members have nearer prime members between the longest two.

**Solution:**

```
import queue
```

```
class Graph:
```

```
    def __init__(self, V):
```

```
        self.V = V;
```

```
        self.l = [[] for i in range(V)]
```

```
    def addedge(self, V1, V2):
```

```
        self.l[V1].append(V2);
```

```
        self.l[V2].append(V1);
```

```
    def bfs(self, in1, in2):
```

```
        visited = [0] * self.V
```

```
        que = queue.Queue()
```

```
        visited[in1] = 1
```

```
        que.put(in1)
```

```
        while (not que.empty()):
```

```
            p = que.queue[0]
```

```
            que.get()
```

```
            i = 0
```

```

while i < len(self.l[p]):
    if (not visited[self.l[p][i]]):
        visited[self.l[p][i]] = visited[p] + 1
        que.put(self.l[p][i])
    if (self.l[p][i] == in2):
        return visited[self.l[p][i]] - 1
    i += 1

```

```

def SieveOfEratosthenes(v):

```

```

    n = 9999

```

```

    prime = [True] * (n + 1)

```

```

    p = 2

```

```

    while p * p <= n:

```

```

        if (prime[p] == True):

```

```

            for i in range(p * p, n + 1, p):

```

```

                prime[i] = False

```

```

            p += 1

```

```

    for p in range(1000, n + 1):

```

```

        if (prime[p]):

```

```

            v.append(p)

```

```

def compare(num1, num2):

```

```

    s1 = str(num1)

```

```

    s2 = str(num2)

```

```

    c = 0

```

```

    if (s1[0] != s2[0]):

```

```

        c += 1

```

```

    if (s1[1] != s2[1]):
        c += 1

    if (s1[2] != s2[2]):
        c += 1

    if (s1[3] != s2[3]):
        c += 1

    return (c == 1)

def shortestPath(num1, num2)

    pset = []

    SieveOfEratosthenes(pset)

    g = Graph(len(pset))

    for i in range(len(pset)):

        for j in range(i + 1, len(pset)):

            if (compare(pset[i], pset[j])):

                g.addedge(i, j)

    in1, in2 = None, None

    for j in range(len(pset)):

        if (pset[j] == num1):

            in1 = j

    for j in range(len(pset)):

        if (pset[j] == num2):

            in2 = j

    return g.bfs(in1, in2)

if __name__ == '__main__':

    num1,num2 = map(list,int(input()))

    print(shortestPath(num1, num2))

```