```
Regular Exporession :
    Regular Expressions one marry used for pattern matching.
   set Import re
       Matchen - re-findites ("[abc]", "dharmaaaa 1234 Df 8#6")
      for match in matchen?
         print (motch-Start(),"....", matche group())
    ole: 2... a
         5 ... 0
  Since here it checking abc ance available & where it is
 available at which position they are available
 Ent import re
    matcher = re. finditer ("[@]", input ("enter"))
   for match in matcher:
        Print (match-Start (1," --- ", match.group ())
   op: enter 123@45
Hatchen in an iterator object, let
     import re
      matcher = re. find 9 ter (" LOJ", injut ("enter")
     Print (matcher)
       ( collable - Herotor object at 0,000027827, D. Sofia)
   offer enter 125045
```

```
Import re
       matcher = re finditer ("[a-ZA-ZD-9]", absAC34.$")
        print (matcher)
       for month on matchers
           Print (match, Start (), ", ....", motch. group ())
      Olpt-try it yourself.
       imbart se
       matcher = re-finditer (" [a-zA-z0-9]", "abcABC123. 4")
      for match in matcher:
           Print (match. Start(), "---", matche goog (1)
   0) p = < callable - iterator object at 0x0001593CAO>
            12.... $
Note & re module
                      Ps mandatay.
     impost me
      pattern = input ( Tenter the pattern to check")
     m=re. Motch ( pattern, "abbaabba")
          print ("Hatch is available)
  of :- enter the pattern to chack abba
         Match & orvailable.
```

pattern = input (renter the pattern to check?) m= re-fullmatch (pattern, "abba") of m!=Nones print ("fall moth is available") · (rantiphory of the