def max\_distinct\_char(str, n):

count = [0] \* NO\_OF\_CHARS

for i in range(n):

count[ord(str[i])] += 1

max\_distinct = 0

for i in range(NO\_OF\_CHARS):

if (count[i] != 0):

max\_distinct += 1

return max\_distinct

def smallesteSubstr\_maxDistictChar(str):

n = len(str)

max\_distinct = max\_distinct\_char(str, n)

minl = n # result

for i in range(n):

for j in range(n):

subs = str[i:j]

subs\_lenght = len(subs)

sub\_distinct\_char = max\_distinct\_char(subs,

subs\_lenght)

if (subs\_lenght < minl and

max\_distinct == sub\_distinct\_char):

minl = subs\_lenght

return minl

if \_\_name\_\_ == "\_\_main\_\_":

# Input String

str = "AABBBCBB"

l = smallesteSubstr\_maxDistictChar(str);

print( "The length of the smallest substring",

"consisting of maximum distinct",

"characters :", l)