			A recess of the first contract in
	2.	Crinen:	
		n = 17	
		a=5	
		Private Key of Alice : 4	
		Private Key of Alice: 4 Private Key of Bob: 6	
		Public Key of Alice: 5 Private Key of Alice mod 17	i s i si s
	=	5 Privated key of Alice mod 17	(CSAC rejains res
	=	54 mod 17	
	-	13	
		Public key of Bob:	
	24	Public key of Bob: 5 private key of Bob mod 17	
	•-	56 mod 17	
	-	2	

Secret key obtained by Alice:

= 2th mod 17

= 16

Secret key obtained by Bob:

= 13 private key of Bob mod 7

= 16

Finally both parties get same value of secret key.

· Value = 16

· Option (A): 16 is correct.

Q4.	String = "GEEKSFORGEEKS" Keyword = "SHARAN"		
	Keyword = "SHARAN"		
	MARIE PRESIDENT ESPERANCE AND		
	des generate Key (string, key):		
	if len (string) = = len(key):		
	return (key)		
	else:		
	for i in range (len (string) - len (key)):		
	ter i in range (len(string)-len(key)): key. append (key [i]/. len(key)]) return ("". join to (key))		
	return (" ". join to (key))		
	per mist. " world?		
	def encrypt_cipher Text (string , key)		
	cipher_text = [] for i in range (lin (string)):		
	for i in range (len (string)):		
	n = ((ord(string [i]) + ord (key [i])) -/. 26) =+ ord('A')		
	return ("". join (cipher_text))		
	return (" '' join (cipher_text))		
	the man to the second second second second		
	key = generatekey (string, keyword)		
	toping to the second to the se		
	print ('Original Message:, stoing)		
	print (Keyword: , keyword)		
	cipher_text = encrypt_ciphertext (string, key) print ('Ciphertext:', cipher_text)		
	print ('Ciphertext:', cipher_text)		
	2 KENEN + M & BENY ' + 1 K + 1		
	Output: Original Message: GEEKSFORGEEKS		
	Keyword: SHARAN		
(Sundaram)	Ciphertext FOR EDUCATIONAL USE: YLEBSSCTYGVEXK		
	(2002-0190)		

```
Decruption of Vignere Cipher:
                                             and Arthur Haartella Commence
                                                  ri
Difference Springers
ciphertext = "YLEBSSGYGVEXK"
Keyword = "SHARAN"
                                            ાં, અતાં ઉત્તર કરાયા વ્યવસાય હતું ક
det generate key (ciphertext, key):
      Key = 1ist (key)
                                         i .... i savajis — i sa, i su. i s
     if len (string) = = len (key):
return key
                    · Jacob and Carack and service in
       for i in range (len (string) - len (key)):

Key append (key [i / lon (key]) =
          return ("" join(key))
 det decrypt-originaltext (ciphertext, key):
                                               that wife therein
     erigtext: []
     tor i in range (con(ciphertext)):
        x = ((0rd (ciphertext [i] - Ord (key [i])) -1.26) + ord ('A')
        origitext appead (un(n))
                                             . Him has in a g
    return (" .join (origtent))
                                          construction of a significant
Key = generate key (ciphertext, keyword)
print ('Uphertext:, ciphertext)
print ('keyword!, keywore)
String = decrypt_original text (ciphertext, key)
print ('Original text=", String)
                                               water and the same
OUTPUT:
ciphertext= 1 YLEBSSGYGVEXK 1
Keyword = 'SHARAN'
original text= GEEKSFORGEEKS
```