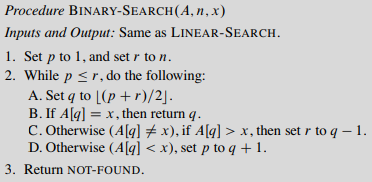
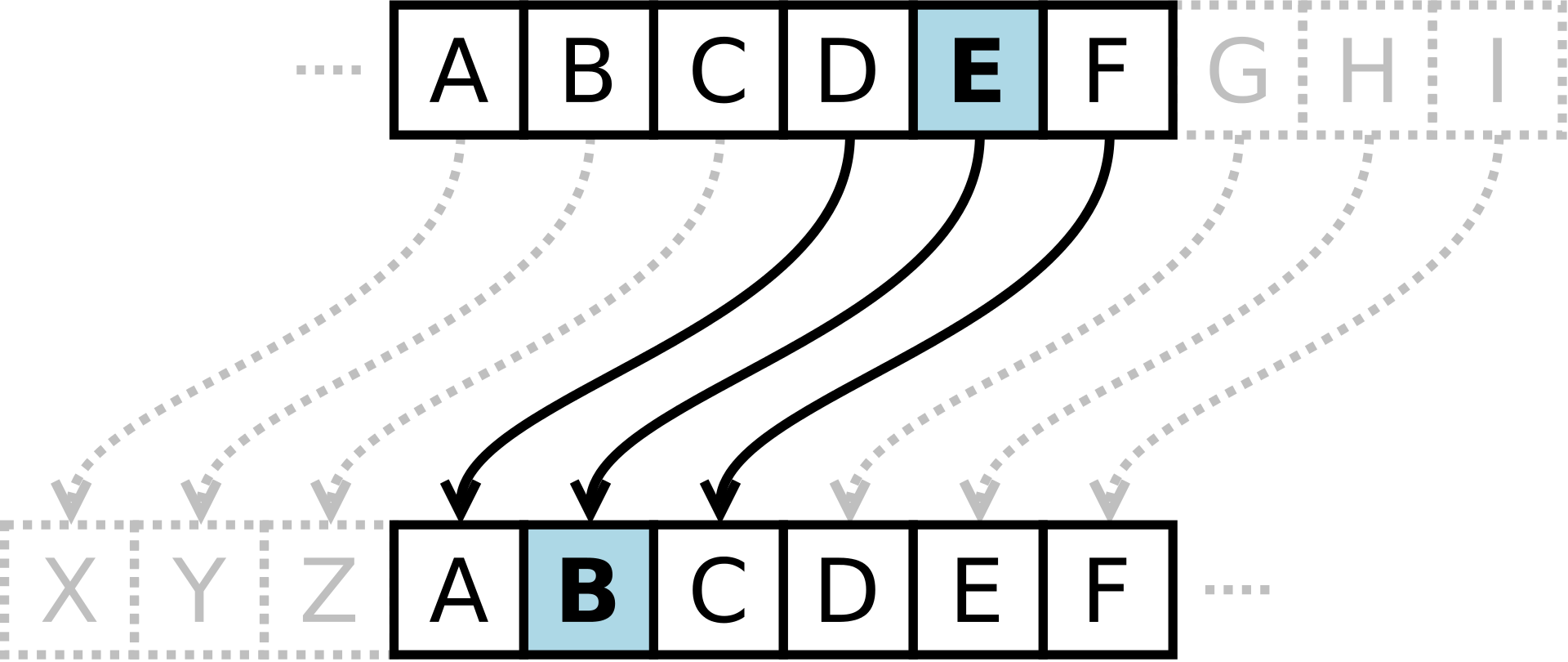
|  |  |
| --- | --- |
|  | DEPARTMENT OF COMPUTER SCIENCEFORMAN CHRISTIAN COLLEGE(A CHARTERED UNIVERSITY),LAHORE |

**Transmission Control Protocol (TCP)**

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll #: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Algorithms:**

Caesar Cipher





Instructions:

* Complete both tasks given below.
* Submit your work before the deadline.

# TASK1:

## Write a TCP client server program in Python, task1client.py and task1server.py, that sends a sorted array of integers and an element from clients’ side to server as input and the server returns that position of that element in the array using binary search as output to client.

|  |
| --- |
| #========================  #task1client.py  #---------------------------------------  # Paste your solution below:  from socket import \*  serverName = 'localhost'  serverPort = 8000  try:  clientSocket = socket(AF\_INET, SOCK\_STREAM)  clientSocket.connect((serverName, serverPort))  arr = [12, 35, 67, 98, 101]  print('Array is:', arr)  el = input('Choose your num: ')  arr\_str = ','.join(str(x) for x in arr)  clientSocket.send((str(el) + ';' + arr\_str).encode())  pos = clientSocket.recv(1024)  if pos:  pos = pos.decode()  if pos != "-1":  print(f"The index of {el} in the array is {pos}")  else:  print(f"{el} is not present in the array")  else:  print("Server did not respond")  except ConnectionRefusedError:  print(f"Could not connect to server at {serverName}:{serverPort}")  finally:  clientSocket.close()  #========================  #task1server.py  #---------------------------------------  # Paste your solution below:  from socket import \*  serverPort = 8000  serverSocket = socket(AF\_INET,SOCK\_STREAM)  serverSocket.bind(('localhost',serverPort))  serverSocket.listen(1)  print('The server is ready to receive...')  def bSearch(data: list, n: int) -> int:  low = 0  high = len(data) - 1  mid = 0    while low <= high:  mid = (high + low) // 2  if data[mid] < n:  low = mid + 1  elif data[mid] > n:  high = mid - 1  else:  return mid  return -1  while True:  connectionSocket, addr = serverSocket.accept()  data = connectionSocket.recv(1024).decode()  n, listt = data.split(';')  arr = list(map(int, listt.split(',')))  connectionSocket.send(str(bSearch(arr, int(n))).encode())  connectionSocket.close() |

# TASK2:

## Caesar Cipher is a type of substitution cipher, in which each letter in the plain text is replaced by another letter at some fixed positions from the current letter in the alphabet. For example, if we shift each letter by three positions to the right, each of the letters in our plain text will be replaced by a letter at three positions to the right of the letter in the plain text. For example if we encrypt the text HELLO WORLD using a right shift of 3, so the letter H will be replaced by K, E will be replaced by H, and so on. The final encrypted message for HELLO WORLD will be KHOOR ZRUOG.

## Below is a function which takes two parameters as input, text and shift. Your task is to simulate a TCP client server program where client sends plaintext and required shift to server and server returns an encrypted text back to client.

def encrypt(text, shift):

    result = ""

    # traverse text

    for i in range(len(text)):

        char = text[i]

        # Encrypt uppercase characters

        if (char.isupper()):

            result += chr((ord(char) + shift - 65) % 26 + 65)

        # Encrypt lowercase characters

        else:

            result += chr((ord(char) + shift - 97) % 26 + 97)

    return result

|  |
| --- |
| #========================  #task2client.py  #---------------------------------------  # Paste your solution below:  from socket import \*  serverName = 'localhost'  serverPort = 8080  clientSocket = socket(AF\_INET, SOCK\_STREAM)  clientSocket.connect((serverName, serverPort))  data = input('Enter your data: ')  clientSocket.send(data.encode())  encryptedData = clientSocket.recv(1024)  print("Encrypted data is: ", encryptedData.decode())  #========================  #task2server.py  #---------------------------------------  # Paste your solution below:  from socket import \*  serverPort = 8080  serverSocket = socket(AF\_INET,SOCK\_STREAM)  serverSocket.bind(('localhost',serverPort))  serverSocket.listen(1)  print('The server is ready to receive...')  def encrypt(text, shift):  result = ""    *# traverse text*  for i in range(len(text)):  char = text[i]    *# Encrypt uppercase characters*  if (char.isupper()):  result += chr((ord(char) + shift - 65) % 26 + 65)  elif (char == ' '):  result += ' '  *# Encrypt lowercase characters*  else:  result += chr((ord(char) + shift - 97) % 26 + 97)    return result  while True:  connectionSocket, addr = serverSocket.accept()  data = connectionSocket.recv(1024).decode()  connectionSocket.send(encrypt(data, 3).encode())  connectionSocket.close() |