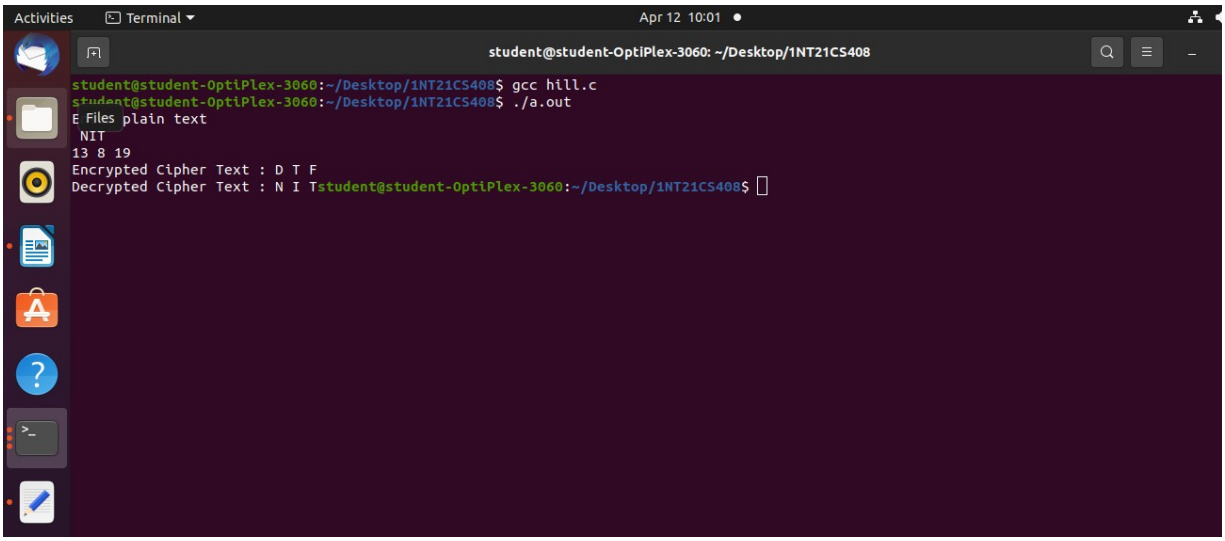
Implement Data Encryption and Decryption using Hill cipher method

```
#include<stdio.h>
#include<string.h>
int main() {
   unsigned int a[3][3] = { { 6, 24, 1 }, { 13, 16, 10 }, { 20, 17, 15 } };
   unsigned int b[3][3] = { { 8, 5, 10 }, { 21, 8, 21 }, { 21, 12, 8 } };
   int i, j;
   unsigned int c[20], d[20];
   char msg[20];
   int determinant = 0, t = 0;
   ;
   printf("Enter plain text\n ");
   scanf("%s", msg);
   for (i = 0; i < 3; i++) {
      c[i] = msg[i] - 65;
      printf("%d ", c[i]);
   }
   for (i = 0; i < 3; i++) {
      t = 0;
      for (j = 0; j < 3; j++) {
         t = t + (a[i][j] * c[j]);
      }
      d[i] = t % 26;
   }
   printf("\nEncrypted Cipher Text :");
   for (i = 0; i < 3; i++)
      printf(" %c", d[i] + 65);
   for (i = 0; i < 3; i++) {
      t = 0;
      for (j = 0; j < 3; j++) {
         t = t + (b[i][j] * d[j]);
      }
      c[i] = t % 26;
   }
   printf("\nDecrypted Cipher Text :");
   for (i = 0; i < 3; i++)
      printf(" %c", c[i] + 65);
   return 0;
}
```

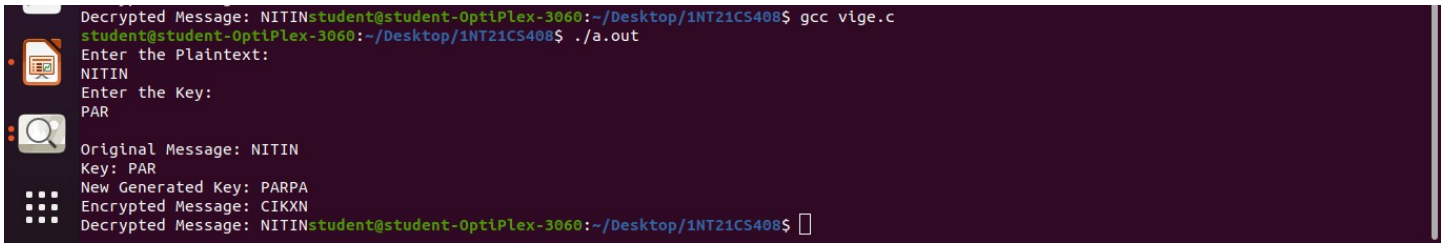



## vigenere cipher

```
#include<stdio.h>
#include<string.h>
int main(){
        char msg[30],key[30];
        printf("Enter the Plaintext:\n");
        scanf("%s",msg);
        printf("Enter the Key:\n");
        scanf("%s",key);
        int msgLen = strlen(msg), keyLen = strlen(key), i, j;
        for(i=0,j=0;i<keyLen;i++)
```

```
        {
                if(key[i]!='\0')
                {
                        key[j]=toupper(key[i]);
                        j++;
                }
        }
        for(i=0,j=0;i<msgLen;i++)
        {
                if(msg[i]!='\0')
                {
                        msg[j]=toupper(msg[i]);
                        j++;
                }
        }
                char newKey[msgLen], encryptedMsg[msgLen], decryptedMsg[msgLen];
        for(i = 0, j = 0; i < msgLen; ++i, ++j){
      if(j == keyLen)
        j = 0;
      newKey[i] = key[j];
    }
    newKey[i] = '\0';
    for(i = 0; i < msgLen; ++i)
      encryptedMsg[i] = ((msg[i] + newKey[i]) % 26) + 'A';
    encryptedMsg[i] = '\0';
    for(i = 0; i < msgLen; ++i)
      decryptedMsg[i] = (((encryptedMsg[i] - newKey[i]) + 26) % 26) + 'A';
    decryptedMsg[i] = '\0';
    for(i=0;i<msgLen;i++)
    {
        printf("%d\t",msg[i]-65);
    }
    printf("\n");
    for(i=0;i<msgLen;i++)
    {
        printf("%d\t",newKey[i]-65);
    }
    printf("\n");
    for(i=0;i<msgLen;i++)
    {
        printf("%d\t",encryptedMsg[i]-65);
    }
    printf("\n");
    printf("Original Message: %s", msg);
    printf("\nKey: %s", key);
    printf("\nNew Generated Key: %s", newKey);
    printf("\nEncrypted Message: %s", encryptedMsg);
    printf("\nDecrypted Message: %s", decryptedMsg);
return 0;
}
```

Output


Decrypted Message: NITINstudent@student-OptiPlex-3060:~/Desktop/1NT21CS408$ gcc vige.c
student@student-OptiPlex-3060:~/Desktop/1NT21CS408$ ./a.out
Enter the Plaintext:
NITIN
Enter the Key:
PAR
Original Message: NITIN
Key: PAR
New Generated Key: PARPA
Encrypted Message: CIKXN
Decrypted Message: NITINstudent@student-OptiPlex-3060:~/Desktop/1NT21CS408$