|  |  |
| --- | --- |
| **TERMWORK 5(Diffie Helman Exchange)**  #include<stdio.h>  long int power (int a, int b, int mod)  {  long long int t;  if(b==1)  return a;  t=power (a, b/2, mod);  if(b%2==0)  return (t\*t) %mod;  else  return (((t\*t) %mod) \*a) %mod;  }  long long int calculateKey (int a, int x, int q)  {  return power (a, x, q);  }  int main ()  {  int q, alpha, x, a, y, b;  printf("Enter the prime number and prime root : ");  scanf("%d %d", &q, &alpha);  printf("Enter the private key of A : ");  scanf("%d", &x);  a=power(alpha, x, q);  printf("Enter the private key of B : ");  scanf("%d", &y);  b=power(alpha,y,q);  printf("A computes key K : %lld \n", power(b, x, q));  printf("B computes key K : %lld\n", power(a, y, q));  return 0;  }  **TERMWORK 6(Hill Cipher)**  #include <stdio.h>  #include <string.h>  void getKeyMatrix(char key[6], int keyMatrix[][3])  {  int k = 0;  for (int i = 0; i< 3; i++)  {  for (int j = 0; j < 3; j++)  {  keyMatrix[i][j] = (key[k]) % 65;  k++;  } } } | void encrypt(int cipherMatrix[][1],  int keyMatrix[][3],  int messageVector[][1])  {  int x, i, j;  for (i = 0; i< 3; i++)  {  for (j = 0; j < 1; j++)  {  cipherMatrix[i][j] = 0;  for (x = 0; x < 3; x++)  {  cipherMatrix[i][j] +=  keyMatrix[i][x] \* messageVector[x][j];  }  cipherMatrix[i][j] = cipherMatrix[i][j] % 26;  } } }  void HillCipher(char message[3], char key[9])  {  int keyMatrix[3][3];  getKeyMatrix(key, keyMatrix);  int messageVector[3][1];  for (int i = 0; i< 3; i++)  messageVector[i][0] = (message[i]) % 65;  int cipherMatrix[3][1];  encrypt(cipherMatrix, keyMatrix, messageVector);  char CipherText[3];  for (int i = 0; i< 3; i++)  CipherText[i] = cipherMatrix[i][0] + 65;  printf("%s", CipherText);  }  int main()  {  char message[3],key[9];  printf("Enter the plaintext :");  scanf("%s",message);  printf("Enter the key :");  scanf("%s",key);  HillCipher(message, key);  return 0; }  **TERMWORK 7(Key gen in simplified DES)**  #include<stdio.h>  int main()  {  int i, cnt=0, p8[8]= {6,7,8,9,1,2,3,4};  int p10[10]= {6,7,8,9,10,1,2,3,4,5};  char input[11], k1[10], k2[10], temp[11];  char LS1[5], LS2[5];  printf("Enter 10 bits input:"); |

|  |  |
| --- | --- |
| scanf("%s",input);  input[10]='\0';  for(i=0; i<10; i++)  {  cnt = p10[i];  temp[i] = input[cnt-1];  }  temp[i]='\0';  printf("\nYour p10 key is :");  for(i=0; i<10; i++)  {  printf("%d,",p10[i]);  }  printf("\nBits after p10 :");  puts(temp);  for(i=0; i<5; i++)  {  if(i==4)  temp[i]=temp[0];  else  temp[i]=temp[i+1];  }  for(i=5; i<10; i++)  {  if(i==9)  temp[i]=temp[5];  else  temp[i]=temp[i+1];  }  printf("Output after LS-1 :");  puts(temp);  printf("\nYour p8 key is :");  for(i=0; i<8; i++)  {  printf("%d,",p8[i]);  }  for(i=0; i<8; i++)  {  cnt = p8[i];  k1[i] = temp[cnt-1];  }  printf("\nYour key k1 is :");  puts(k1);  }  **TERMWORK 8(Vigenere Cipher)**  #include<stdio.h>  #include<string.h>  #include<ctype.h> | #include<stdlib.h>  main()  {  int i,j,k,numstr[100],numkey[100],numcipher[100];  char str[100],key[100];  printf("Enter a string\n");  gets(str);  for(i=0,j=0; i<strlen(str); i++)  {  if(str[i]!=' ')  {  str[j]=toupper(str[i]);  j++; } }  str[j]='\0';  printf("Entered string is : %s \n",str);  for(i=0; i<strlen(str); i++)  {  numstr[i]=str[i]-'A';  }  printf("Enter a key\n");  gets(key);  for(i=0,j=0; i<strlen(key); i++)  {  if(key[i]!=' ')  {  key[j]=toupper(key[i]);  j++; } }  key[j]='\0';  for(i=0; i<strlen(str);)  {  for(j=0; (j<strlen(key))&&(i<strlen(str)); j++)  {  numkey[i]=key[j]-'A';  i++; } }  for(i=0; i<strlen(str); i++)  {  numcipher[i]=numstr[i]+numkey[i];  }  for(i=0; i<strlen(str); i++)  {  if(numcipher[i]>25)  {  numcipher[i]=numcipher[i]-26;  } }  printf("Vigenere Cipher text is\n");  for(i=0; i<strlen(str); i++)  {  printf("%c",(numcipher[i]+'A'));  }  printf("\n"); } |