|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TERMWORK 1(Substitution Cipher)**  #include<stdio.h>  int main(){  char message[100], ch, str;  int i, key, x;  printf("Enter a message : ");  gets(message);  printf("Enter key: ");  scanf("%d", &key);  printf("\nPlease choose following options:\n");  printf("1 = Encrypt the string.\n");  printf("2 = Decrypt the string.\n");  scanf("%d", &x);  //using switch case statements  switch(x){  case 1:  for(i = 0; message[i] != '\0'; ++i) {  ch = message[i];  if(ch>= 'a' &&ch<= 'z'){  ch = ch + key;  if(ch> 'z') { ch = ch - 'z' + 'a' - 1;  } message[i] = ch;  }  else if(ch>= 'A' &&ch<= 'Z') {  ch = ch + key;  if(ch> 'Z'){  ch = ch - 'Z' + 'A' - 1;  }message[i] = ch;  } }  printf("Encrypted message: %s", message);  break;  case 2:  for(i = 0; message[i] != '\0'; ++i) {  ch = message[i];  if(ch>= 'a' &&ch<= 'z') {  ch = ch - key;  if(ch< 'a') {  ch = ch + 'z' - 'a' + 1;  } message[i] = ch;  } else if(ch>= 'A' &&ch<= 'Z')  {ch = ch - key;  if(ch< 'A'){  ch = ch + 'Z' - 'A' + 1;  }message[i] = ch;  } }  printf("Decrypted message: %s", message);  break;  default:  printf("\nError\n");  } return 0;} | | | **TERMWORK 2(RSA Algo)**  #include <stdio.h>  #include<stdlib.h>  int gcd(int a,int b) {  int c;while(a!=b) {  if(a<b) {  c=a;a=b;b=c;  } a-=b;  } return a;  } int mod(int m,inte,int n)  {  int a=1;  while(e) {  a=(a\*m)%n;  e--;  } return a;  }  int main() {  int p,q,n,e,m,c,d,x,z;  int en[100],de[100],j=0;  printf("\nEnter the value of P & Q\n");  scanf("%d%d",&p,&q);  n=p\*q;  z=(p-1)\*(q-1);  for(e=1;e<n;e++) {  if(gcd(e,z)==1) {  en[j]=e;  printf(" %d",en[j++]);  } }  printf("\nChoose e\n");  scanf("%d",&e);  if(gcd(e,z)!=1) {  printf("\nThe value not from list\n");  exit(0);  } printf("Enter the message(integer value) to be encrypted:\n");  scanf("%d",&m);  printf("Before encryption:%d\n",m);  c=mod(m,e,n);  printf("After encryption:%d\n",c);  printf("The possible Decryption keys Are:");  for(d=0;d<n;d++) {  if((d\*e)%z==1) {  de[j]=d;  printf(" %d",de[j++]);  } }  printf("\nChoose D\n");  scanf("%d",&d);  x=mod(c,d,n);  printf("After decryption=%d\n",x);  return 0; } | |
| **TERMWORK 3(Password Strength)**  int main()  {  int i,n,a=0,d=0,s=0;  char p[10];  printf("Enter the Password: ");  gets(p);  n=strlen(p);  if(n>=6){  for(i=0;i<n;i++){  if(isalpha(p[i])) {  a+=1;}  else if(isdigit(p[i])){  d+=1;}  else {  s+=1;  } }  if(a>=1 && d>=1 && s>=1)  {  printf("Strong Password");  } else if((a>=1 && d>=1) || (a>=1 && s>1) || (d>=1 && s>=1))  { printf("Moderate Password");  } else  {  printf("Weak Password");  } }  else  {  printf("Invalid Password");  } }  **Sample Output**  Enter password : aw1  Invalid Password  Enter password : adckex  Weak Password  Enter password : abc123  Moderate Password  Enter password : abc2#@  Strong Password | **TERMWORK 4(Rail Fence)**  #include<stdio.h>  #include<string.h>  void encryptMsg(char msg[], int key)  {  int msgLen = strlen(msg), i, j, k = -1, row = 0, col = 0;  char railMatrix[key][msgLen];  for(i = 0; i< key; ++i)  for(j = 0; j <msgLen; ++j)  railMatrix[i][j] = '\n';  for(i = 0; i<msgLen; ++i)  {  railMatrix[row][col++] = msg[i];  if(row == 0 || row == key-1)  k= k \* (-1);  row = row + k;  }  printf("\nOutput :");  char nlet[100];  for(i = 0; i< key; ++i)  for(j = 0; j <msgLen; ++j)  if(railMatrix[i][j] != '\n')  {  printf("%c", railMatrix[i][j]);  strncat(nlet, &railMatrix[i][j], 1);  }  }  int main()  {  char msg[100];  int key;  printf("Encryption:");  printf("\nInput:");  scanf("%[^\n]s",msg);  printf("Key = ");  scanf("%d",&key);  encryptMsg(msg, key);  return 0;  }  **Sample Output**  Encryption  Input : attack at once  Key = 2  Output : atctoctaka ne | |