Name - SAKSHI MANRAL ①

Roll no:- 1022756 (46)

Q.2.) WACP to implement OTP Method with Plain text "onetime Pad" with Key "Perfect".

Ans:)

```
#include <stdio.h>
# include <string.h>
# include <ctyke.h>
main()
{int i,j, len1, len2, numstr[100], numKey[100],
  numcipher[100];
char str[100], Key[100], cipher[100];
Printf("Enter a string text to encrypt\n");
gets (str);
for(i=0; j=0; i<strlen(str); i++)
 { if (str[i] != ' ')
  { str[j] = toupper(str[i]);
    j++;
  }}
  str[j] = '\0';
    for(i=0; i<strlen(str); i++)
    { numstr[i] = str[i] - 'A';
    }
    Printf("Enter Key string of random text\n");
    gets (Key);
   for(i=0; j=0; i<strlen(Key); i++)
  { if(Key[i] != ' ')
   { Key[j] = toupper(Key[i]);
     j++;
```

Sakshi

```
}}
Key[j] = '\0';
for(i=0; i<strlen(Key); i++)
{ 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(Key); i++)
    { numkey[i] = Key[i] - 'A';
    }
    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("One Time Pad Cipher text is \n");
  for(i=0; i<strlen(str); i++)
    { printf("%c", (numcipher[i] + 'A'));
    }
    printf("\n");
    }
```

Lakshi

Name → SAKSHI MANRAL
Roll no:- 1022756 (46)

Q.)3 WACP to implement Route Cipher encryption decryption.

Ans.)

```
#include <stdio.h>
# include <ctyke .h>
# include <string .h>
int main()
{ char Plantxt[100], ar[5][5];
 char temptxt[] = "zzzzzzzzzzzzzzzzzzzzzz";
 Printf("enter plan txt \n");
 fflush(Stdin);
 fgets(Plantxt, Sizeof(Plantxt), stdin);
 int K=0, K2=0;
 while(Plantxt[K] != '\0')
{
 if(isalpha(Plantxt[K])){temptxt[K2]=Plantxt[K];
 K2++;
 }
  K++;
 }
  Puts(temptxt); Printf("\n");
   K=0;
   for(int j=0; j<5; j++)
 {
  for(int i=0; i<5; i++)
  { ar[i][j] =temptxt[K];
      K++;
      }}
```

Sakshi

```
for(int i = 0; i<5; ++i)
{for(int j = 0; j<5; ++j)
 {
  printf("%c", or[i][j]);
  }
  printf("\n");
 } return 0;
   }
```

OUTPUT

```
Enter Plan txt
Sakshimanral
Sakshimanral2222222222222222
```

S	i	a	z	z
a	m	l	z	z
K	a	z	z	z
S	n	z	z	z
h	r	z	z	z

Q5) WACP to implement Playfair Cipher....

Ans:

```
#include<stdio.h>
#include<stdlib.h>
# include<string.h>
# define SIZE 30
void toLowerCase(char plain[], int ps)
{ int i;
  for(i=0; i<ps; i++)
 { if Plain[i]>64 && Plain[i]<91)
        plain[i] += 32;
```

Lakshi

```
}}
int removeSpace(char* plain, int ps)
{
    int i, count = 0;
    for(i=0; i<PS; i++)
        if(plain[i] != ' ')
            plain[count++] = plain[i];
        plain[count] = '\0';
        return count;
    }
void generateKeyTable(char Key[], int KS, char KeyT[5][5])
{
    int i, j, Kflag = 0, *dicty;
    dicty = (int *) calloc(26, sizeof(int));
    for(i=0; i<KS; i++)
    {
        if(Key[i] != 'j')
            dicty[Key[i] - 97] = 2;
    }
    dicty[j - 97] = 1; i = 0; j = 0;
    for(K=0; K<KS; K++)
    {
        dicty[j - 97]
        if(dicty[Key[K] - 97] == 1)
            KeyT[i][j] = Key[K];
            j++;
        if(j == 5)
        {
            i++;
            j = 0;
        }
    }
}
```

Lakshi

```
for(K=0;K<26;K++)
{ if(dicty[K]==0)
  {
    KeyT[i][j]=(char)(K+97);
        j++;
    if(j==5)
   { i++;
     j=0;
   }
  }
}
void Search(char KeyT[5][5],char a,char b,int arr[])
{ int i,j;
  if(a==(j))
      a=(i);
  else if(b==(j))
      b=(i);
  for(i=0;i<5;i++)
  { for(j=0;i<5;j++)
    { if KeyT[i][j]==a)
      { arr[0]=i;
        arr[1]=j;
      }
    else if(KeyT[i][j]==b)
    { arr[2]=i;
      arr[3]=j;
    }}
  }
```

Lakshi

```
int mod5(int a)
{ return(a%5);
  }
int prepare(char str[], int ptrs)
 { if(Ptrs%2 != 0)
   { str[Ptrs++] = 'z';
     Str[Ptrs] = '\0';
    } return ptrs;
  }
 Void encrypt(char str[], char keyT[5][5], int ps)
 { int i, a[4];
 for(i=0; i<ps; i+=2)
 { Search(KeyT, str[i], str[i+1], a);
  if(a[0] == a[2])
  { str[i] = KeyT[a[0]] mod5 (a[1]+1)];
    str[i+1] = keyT[a[0]] mod5(a[3]+1)];
   }
 else if(a[1] == a[3])
 { str[i] = KeyT[mod5(a[0]+1)][a[1]]; }
   else { str[i] = KeyT[a[0]][a[3]];
          str[i+1] = KeyT[a[2]][a[1]];
      }}
      }
Void encryptByPlayfairCipher(char str[], char key[])
{ char ps, ks, keyT[5], [5];
   ks = strlen(key);
      ks = removeSpaces(key, ks);
      toLowerCase(key, ks);
```

Lakshi

```
PS = strlen (str);
    toLowerCase (str, ps);
    PS = prepare (str, PS);
    generateKeyTable (Key, KS, KeyT);
    encrypt (str, KeyT, PS);
}
int main()
{ char str[SIZE], Key[SIZE];
  strcpy (Key, "Playfair");
  Printf("Key text: %s \n", Key);
  strcpy (str, "~~mask~~Key");
  Printf("Plain text: %s \n", str);
  encryptByPlayfairCipher (str, Key);
  Printf ("Cipher text: %s \n", str);
  return 0;
}
```

Lakshi


```
Key text: key
Plain text: go with the flow
Cipher text: hnumzozoydmixu

--------------------------------
Process exited after 0.05294 seconds with return value 0
Press any key to continue . . .
```