Prakhar Joshi

Roleno-1022749/39

Course - B.Sc (IT) 6th

Pjoshi

Q5 Play fair Cipher

```
# include < stdio.h>
# include < stdlib.h>
# include < string.h>
# define SIZE 50
  void to Lower Case (char plain [ ], int ps)
    {
      int i;
     for (i=0; i<ps; i++) {
      if (plain [ i ] > 64 && plain[i] < 91)
          plain [ i ] += 32;
       }
    }
  int remove Spaces (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 generate Key Table ( char key [ ], int ks, char key [5][5])
     {
      int i, j, k, flag = 0, * d;
      d = (int*) calloc (26, sizeof ( int));
      for ( i = 0; i < ks; i++)
        {
         if (key [ i ] != 'j')
         d [key [ i ] - 97] = 2;
        }
      d [ 'j' - 97] = 1;
        i = 0;
        j = 0
```

```
for (k = 0; k < ks; k++) {
    if (d[key[k] - 97] == 2) {
     d[key[k] - 97] -= 1;
    key[i][j] = key[k];
      j++;
     if (j == 5) {
        i++;
       j = 0;
     }
   }
 }

 for (k = 0; k < 26; k++) {
    if (d[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; j < 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;
      }
     }
    }
   }
int mod5(int a)
   { return (a % 5);
   }
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]];
       str[i+1] = keyT[mod5(a[2]+1)][a[1]];
     }

    }
   }
  }
void encryptByPlayfairCipher(char str[], char key[])
  { char ps, ks, keyT[5][5];
    ks = strlen(key);
    ks = removeSpace(key, ks);
    tolowerCase(key, ks);
   ps = strlen(str);
    to lowercase(str, ps);
    ps = removeSpaces(str, ps);
   generatekeyTable(key, ks, keyT);
    encrypt(str, keyT, ps);
   }
```

```
int main()
{
    char str[SIZE], key [SIZE];
    strcpy(key, "key");
    printf("Key Text: %s\n", key);
    strcpy(str, "go with the flow");
    printf("plain text : %s\n", str);
    encryptByPlayfairCipher(str, key);
    printf("Cipher text : %s\n", str);
    return 0;
}
```