***SAVEETHA SCHOOL OF ENGINEERING***

***SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCE***

**EXP NO 17:** Write a C program for PT-109 American patrol boat, under the command of Lieutenant John F.Kennedy, was sunk by a Japanese destroyer, a message was received at an Australian wireless station in  
Playfair code:  
KXJEY UREBE ZWEHE WRYTU HEYFS  
KREHE GOYFI WTTTU OLKSY CAJPO  
BOTEI ZONTX BYBNT GONEY CUZWR  
GDSON SXBOU YWRHE BAAHY USEDQ

**AIM**

To Write a C program for PT-109 American patrol boat, under the command of Lieutenant John F.Kennedy, was sunk by a Japanese destroyer, a message was received at an Australian wireless station in  
Playfair code:  
KXJEY UREBE ZWEHE WRYTU HEYFS  
KREHE GOYFI WTTTU OLKSY CAJPO  
BOTEI ZONTX BYBNT GONEY CUZWR  
GDSON SXBOU YWRHE BAAHY USEDQ

**PROCEDURE**

* Download and install any c application
* Traverse the given text one character at a time .
* For each character, transform the given character as per the rule,depending on whether we’re encrypting or decrypting the text.
* Return the new string generated.

**PROGRAM**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#define SIZE 5

// Function to remove spaces and non-alphabetic characters from the input text

void preprocess(char \*text) {

int i, j = 0;

for (i = 0; text[i] != '\0'; i++) {

if (isalpha(text[i])) {

text[j++] = toupper(text[i]);

}

}

text[j] = '\0';

}

// Function to find the position of a letter in the key square

void findPosition(char keySquare[SIZE][SIZE], char letter, int \*row, int \*col) {

int i, j;

for (i = 0; i < SIZE; i++) {

for (j = 0; j < SIZE; j++) {

if (keySquare[i][j] == letter) {

\*row = i;

\*col = j;

}

}

}

}

// Function to decrypt a bigram using the Playfair cipher

void decryptBigram(char keySquare[SIZE][SIZE], char bigram[2], char plaintext[3]) {

int row1, col1, row2, col2;

findPosition(keySquare, bigram[0], &row1, &col1);

findPosition(keySquare, bigram[1], &row2, &col2);

// Case 1: If the letters are in the same row

if (row1 == row2) {

plaintext[0] = keySquare[row1][(col1 - 1 + SIZE) % SIZE];

plaintext[1] = keySquare[row2][(col2 - 1 + SIZE) % SIZE];

}

// Case 2: If the letters are in the same column

else if (col1 == col2) {

plaintext[0] = keySquare[(row1 - 1 + SIZE) % SIZE][col1];

plaintext[1] = keySquare[(row2 - 1 + SIZE) % SIZE][col2];

}

// Case 3: If the letters form a rectangle

else {

plaintext[0] = keySquare[row1][col2];

plaintext[1] = keySquare[row2][col1];

}

plaintext[2] = '\0';

}

int main() {

char keySquare[SIZE][SIZE] = {

{'K', 'X', 'J', 'E', 'Y'},

{'U', 'R', 'E', 'B', 'Z'},

{'W', 'T', 'Y', 'H', 'F'},

{'S', 'G', 'A', 'L', 'O'},

{'P', 'C', 'D', 'M', 'N'}

};

char ciphertext[] = "KXJEYUREBEZWEHEWRYTUHEYFSKREHEGOYFIWTTTUOLKSYCAJPOBOTEIZONTXBYBNTGONEYCUZWRGDSONSXBOUYWRHEBAAHYUSEDQ";

int i;

char bigram[2];

char plaintext[3];

preprocess(ciphertext);

printf("Decrypted message: ");

for (i = 0; i < strlen(ciphertext); i += 2) {

bigram[0] = ciphertext[i];

bigram[1] = ciphertext[i + 1];

decryptBigram(keySquare, bigram, plaintext);

printf("%s", plaintext);

}

printf("\n");

return 0;

}

**OUTPUT**

