

```
5) #include <stdio.h>
#include <ctype.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;
    }
  }
}
```

```
void generateKeytable (char key[], int ks,
                      char keyT [5][5])
```

```
{ int i, j, k, flag = 0, * dicty;
  dicty = (int *) calloc (26, size of (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++)
  {
```

Arshad

```

if (dicty[key[k]-97] == 2)
{
    dicty[key[k]-97] = 1;
    keyT[i][j] = key[k];
    j++;
    if (j == 5)
    {
        j++;
        j = 0;
    }
}
}

```

```

void search (char keyT[5][5], char a, char,
             int arr[])

```

```

{
    int i, j;
    if (a == 'j')
        a = 'i';
    else if (a == 'i')
        b = 'j';
    for (i = 0; i < 5; i++)
    {
        for (j = 0; j < 5; j++)
        {
            if (keyT[i][j] == a)
            {
                arr[0] = i;
                arr[1] = j;
            }
        }
    }
}

```

Abhishek

```
else if (keyT[i][j] == b)
```

```
{ arr[2] = i;
```

```
arr[3] = j;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
int mod5 (int a)
```

```
{ return (a % 5);
```

```
}
```

```
void encrypt(char str[], char key[])
```

```
{
```

```
char ps, ks, keyT[5][5];
```

```
ks = strlen(key);
```

```
ks = removeSpaces(key, ks);
```

```
toLowerCase(key, ks);
```

```
ps = strlen(str);
```

```
toLowerCase(str, ps);
```

```
ps = prepare(str, ps);
```

```
generateKeyTable(key, ks, keyT);
```

```
encrypt(str, keyT, ps);
```

25

Ahmed

```
int main()
```

```
{
```

```
    char str[size], key[size];
```

```
    strcpy(key, "KEY");
```

```
    strcpy(str, "Go with the flow");
```

```
    encrypt(str, key);
```

```
    printf("Cipher-Text : %s\n", str);
```

```
    return 0;
```

```
}
```

Encryption \times Decryption

Let T be the plain text and K be the key.

Let C be the cipher text.

Let P be the plain text.

Let K be the key.

Let C be the cipher text.

Let P be the plain text.

Let K be the key.

Let C be the cipher text.

Let P be the plain text.

Asht