

VIRAS SINGH PANWAR

BSc(IT) 6th sem

(1022772)

Ans 73

```
#include <stdio.h>
```

```
#include <string.h>
```

```
void cipher (int i, int c);
```

```
int findMin();
```

```
void makeArray (int, int);
```

```
char arr[22][22], data[22][22], cmessage[111],  
      setmessage[111], key[55];
```

```
char temp[55], temp2[55];
```

```
int k=0;
```

```
int main() {
```

```
    char *message, *dmessage;
```

```
    int i, j, klen, emlen, flag=0;
```

```
    int r, c, index, min, rows;
```

```
    clrscr();
```

```
    printf("Enter the key\n");
```

```
    fflush(stdin);
```

```
    gets(key);
```

```
    printf("Enter message to be Ciphersed");
```

```
    fflush(stdin);
```

```
gets (message);  
strcpy (temp, key);  
klen = strlen (key);
```

```
k=0;
```

```
for (i=0; ; i++) {  
    if (flag==1)  
        break;
```

```
    for (j=0; key[j] != NULL; j++) {  
        if (message[k] == NULL) {  
            flag = 1;  
            arr[i][j] = '-';
```

```
        } else {
```

```
            arr[i][j] = message[k++];
```

```
        } } }
```

```
r=i;
```

```
c=j;
```

```
for (i=0; i<r; i++) {
```

```
    for (j=0; j<c; j++) {
```

```
        printf("%c", arr[i][j]); }
```

```
    printf("\n"); }
```

```
k=0;
```

```
for (i=0; i<klen; i++) {
```

```
    index = findMin();
```

```
    cipher(index, r); }
```



```

emessage[k] = '\0';
printf("Encrypted message is\n");
for (i=0; emessage[i] != NULL; i++)
    printf("%c", emessage[i]);
printf("\n\n");
emlen = strlen(emessage);
strcpy(temp, key);
rows = emlen / klen;
rows;
do. j=0;
for (i=0; k=1; emessage[i] != NULL; i++, k++) {
    printf("\n Emilen = %d", emlen);
    temp2[j++] = emessage[i];
    if ((k % rows) == 0) {
        temp2[j] = '\0';
        index = find_min();
        makeArray(index, rows);
        j=0;
    }
    printf("\n");
}

```

```

void cipher(int i, int d) {
    int j;
    for (j=0; j<d; j++) {
        emessage[k++] = arr[j][i];
    }
}

```

```
// cmessage[k] = '10'; }
```

```
void makeArray (int col, int row) {  
    int i, j;
```

```
    for (i=0; i<row; i++) {
```

```
        data[i][col] = temp2[i]; } }
```

```
int findMin () {
```

```
    int i, j, min, index;
```

```
    min = temp[0];
```

```
    index = 0;
```

```
    for (j=0; temp[j] != NULL; j++) {
```

```
        if (temp[j] < min) {
```

```
            min = temp[j];
```

```
            index = j;
```

```
        }
```

```
        temp[index] = 123;
```

```
        return (index);
```

```
    }
```