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

void cipher(int i, int c);
int findMin();
void makeArray(int, int);

char arr[22][22], darr[22][22],
 emenage[11], retmenage[11], key
[55];
char temp[55], temp2[55];
int k = 0;

int main()
{
  char *menage, *dmenage;

  int i, j, klen, emlen, flag=0;
  int r, c, index, min, rows;
  clrscr();

  Printf("Enter the key \n");
  fflush(stdin);
   gets(key);

  Printf("\n Enter menage to be
       ciphered \n");
```

```
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 = j;
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();
ciphere[index] = k[i];

}
emessage[k] = '\0';
printf ("\n\n Encrypted message: \n");
for (i=0; emessage[i] != NULL; i++)
printf ("%c", emessage[i]);

Printf ("\n\n");
// decipering.
```

```
emlen = strlen (emessage);
// emlen is length of encrypted m...

strcpy (temp, keep);

rows = emlen / klen;
// rows is no of rows of the
 array to made from
 rows;
j = 0;

for (i=0, k=1; emessage[i] != NUL
                    i++, k++, k++)

{
// printf ("\n Emlen = %d", emlen)
 temp2[j++] = emessage[i];
 if ((k % rows) == 0)

{
 temp2[j] = '\0';
 index = findMi ();
 makeArray (index, rows);
 j = 0;
}
}

Printf ("\n Array Retrieved is \n");
```

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

{
Printf ("%c ", darr[i][j]);
// retrieving message
retmessage [k++] = darr[i][j];

}
Printf ("\n");
}
 retmessage [k] = '\0';

Printf ("\n Message ketrie is \n");

for (j=0; retmessage[j] != NULL; it
printf ("%c", retmessage[j]);

getch ();
 return (0);

}
 void makearray (int col, int row
{
 int i, j;
```

```
for (i=0; i<no
{
darr[i][col] = temp2[i];

}
int findMin()
{
int i,j, min, index;

  min = temp[0];
  index = 0;
  for (j=0; temp[j] != node; j++

if (temp[j]<min
{
min = temp[j];
  Index = j;

}
 }
  temp[index] = 123;
    return (index);

}.
```