

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      ///=====ENKRIPSI=====
6      char plan[100], key[100], chipper[100]="";
7      char abjad[26]="abcdefghijklmnopqrstuvwxyz";
8      char temp;
9      int a, b, i, j, turun;
10
11     printf("=====PROGRAM ENKRIPSI DAN DEKRIPSI=====\\n\\n");
12     printf("masukkan plantext : "); gets(&plan);
13     printf("masukkan key text : "); gets(&key);
14
15     ///menghitung panjang key dan plan text
16     a = strlen(key);
17     b = strlen(plan);
18
19     ///menghilangkan space pada string plan
20     for(i=0; i<b; i++)
21     {
22         for(j=0; j<b; j++){
23             if(plan[j] == ' ')
24             {
25                 temp = plan[j];
26                 plan[j] = plan[j+1];
27                 plan[j+1] = temp;
28             }
29         }
30     }
31
32     ///menggabungkan space pada string key
33     for(i=0; i<a; i++)
34     {
35         for(j=0; j<a; j++){
36             if(key[j] == ' ')
37             {
38                 temp = key[j];
39                 key[j] = key[j+1];
40                 key[j+1] = temp;
41             }
42         }
43     }
44
45     ///menghitung panjang key dan plan text setelah menghilangkan space
46     a = strlen(key);
47     b = strlen(plan);
48
49     ///menghitung urutan array key
50     int panjang[a];
51     int angka=1;
52     for(i=0; i<26; i++)
53     {
54         for(j=0; j<a; j++)
55         {
56             if(abjad[i]==key[j])
57             {
58                 panjang[j]=angka;
59                 angka++;
60             }
61         }
62     }
63
64     ///mencetak urutan key
65     printf("urutan key : ");
66     for(i=0; i<a; i++)

```

```

67     {
68         printf("%d ", panjang[i]);
69     }
70     printf("\n\n\n");
71     printf("===== ENKRIPSI =====\n\n");
72     ///mengidentifikasi panjang matrik menurun
73     if(strlen(plan)%a==0){
74         turun = b/a;
75     }else{
76         turun=b/a+1;
77     }
78
79     ///membuat string matrik
80     int k=0;
81     char matrik[turun][a];
82     for(i=0; i<turun; i++)
83     {
84         for(j=0; j<a; j++)
85         {
86             if(k<b){
87                 matrik[i][j] = plan[k];
88             }else{
89                 matrik[i][j] = ' ';
90             }
91             k++;
92         }
93     }
94
95     ///mencetak matrik
96     printf("==MATRIK ENKRIPSI==\n");
97     for(i=0; i<turun; i++)
98     {
99         for(j=0; j<a; j++)
100         {
101             printf("%c ", matrik[i][j]);
102         }
103         printf("\n");
104     }
105
106
107
108     ///membuat chipper
109     angka=1;
110     for(i=0; i<a; i++)
111     {
112         for(j=0; j<a; j++)
113         {
114             if(panjang[j]==angka)
115             {
116                 for(k=0; k<turun; k++)
117                 {
118                     if(matrik[k][j]!=' ')
119                     {
120                         int P = strlen(chipper);
121                         chipper[P]=matrik[k][j];
122                     }
123                 }
124                 angka++;
125             }
126         }
127     }
128
129     ///output chipper text
130     printf("\n\nChipper text adalah : %s\n\n\n", chipper);
131     printf("===== DEKRIPSI =====\n\n");
132

```

```

133
134 //=====DEKRIPSI=====
135 printf("==MATRIK DEKRIPSI==\n");
136 ///membuat matrik dekripsi
137 char matrikdone[turun][a];
138 angka=1;
139 int urut=0;
140 int kosong = a-((turun*a)%b);
141 for(i=0; i<a; i++)
142 {
143     for(j=0; j<a; j++)
144     {
145         if(panjang[j]==angka)
146         {
147             if(j>=kosong)
148             {
149                 for(k=0; k<turun; k++)
150                 {
151                     if(k!=turun-1)
152                     {
153                         matrikdone[k][j] = chipper[urut];
154                         urut++;
155                     }else
156                     {
157                         matrikdone[k][j] = ' ';
158                     }
159                 }
160             }else
161             {
162                 for(k=0; k<turun; k++)
163                 {
164                     matrikdone[k][j] = chipper[urut];
165                     urut++;
166                 }
167             }
168             angka++;
169         }
170     }
171 }
172
173 ///mencetak matrik hasil dekripsi
174 for(i=0; i<turun; i++)
175 {
176     for(j=0; j<a; j++)
177     {
178         printf("%c ", matrikdone[i][j]);
179     }
180     printf("\n");
181 }
182
183 fflush(stdin);
184 ///membuat string dari matrik dekripsi
185 int jumlah = turun*a-((turun*a)%b);
186 urut = 0;
187 char plandone[100]="";
188 for(i=0; i<turun; i++)
189 {
190     for(j=0; j<a; j++)
191     {
192         plandone[urut] = matrikdone[i][j];
193         urut++;
194     }
195 }
196
197 ///mencetak string hasil dekripsi
198 printf("\nHasil deskripsi adalah : %s\n", plandone);

```

```
199  
200     return 0;  
201 }
```