

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <ctype.h>
5
6  //?????? ???? ??? BinTreeElementType.
7  typedef struct
8  {
9      char data[20];
10     char dummy;
11 } BinTreeElementType;
12
13 typedef struct BinTreeNode *BinTreePointer;
14 struct BinTreeNode {
15     BinTreeElementType Data;
16     BinTreePointer LChild, RChild;
17 } ;
18
19 typedef enum {
20     FALSE, TRUE
21 } boolean;
22
23 //?????????????? ??? ?? HuffmanDecoding.c .
24 void BuildDecodingTree(BinTreePointer *Root, FILE *CodeFile);
25 void AddToTree(char ch[], char code[], int codelen, BinTreePointer *Root);
26 void Decode(BinTreePointer Root, FILE* MessageFile);
27
28 int main()
29 {
30     //?????? ????????????.
31     FILE *CodeFile,*MessageFile;
32     BinTreePointer ARoot;
33
34     //????????? ??? ????????.
35     CodeFile = fopen("codesRW.txt","r");
36
37     //?? ??????? ?????? ???? ?? ???????,????????? ?????????????? ???????????.
38     if(CodeFile == NULL)
39         printf("Can't open codesRW.txt\n");
40     else
41     {
42         //?????? ?????? ??? ??????????? BuildDecodingTree ??? ??? ???????????
43         //??? ???.
44         BuildDecodingTree(&ARoot, CodeFile);
45
46         //????????? ??? ??????? ?? ?? ?????????????????? ???????.
47         MessageFile = fopen("program.txt","r");
48
49         //?? ??????? ?????? ??? ???????,????????? ?????????????? ???????????.
50         if(MessageFile == NULL)
51             printf("Can't oped program.txt\n");
52         else
53             //?????? ?????? ??????????? Decode ??? ?????????????????? ??? ???????????.
54             Decode(ARoot,MessageFile);
55     }
56
57     //????????? ????????.
58     fclose(CodeFile);
59     fclose(MessageFile);
60
61     system("PAUSE");
62 }
63
64
65 /*????? ?? ?????????????? ?????????????? ?? ?????? ?? HuffmanDecoding.c,
66 ?????? ?????? ?? ?????????????? ?????????? ?????? ?? ?????????????? ??? ?????????? ??????

```

```

67  ??? ???? ???? ????????? ???? ???? ????.* /
68 void BuildDecodingTree(BinTreePointer *Root, FILE *CodeFile)
69 {
70     //????? ???? ???? ???? ???? ????????? ? ? ????????? ???? ????????? ?????.
71     char ch[20],code[10],termch;
72     int nscan;
73
74     (*Root) = (BinTreePointer)malloc(sizeof (struct BinTreeNode));
75     (*Root)->Data.dummy = '*';
76     (*Root)->LChild=NULL;
77     (*Root)->RChild=NULL;
78
79     while(TRUE)
80     {
81         /*????? ???? ? ? ????????? ???? ???? ???? ???? ????????????? ????(? ?????
82         ????????? ???? ???? ???? ???? ???? ????????? ???? ????????? ???????????.*/
83         nscan = fscanf(CodeFile, "%20[^\n]%c\n%10[^\n]\n",ch,&termch,code);
84
85         if(nscan == EOF ) break;
86         if(nscan != 3 || termch != '\n')
87         {
88             printf("Improper file format.\n");
89             break;
90         }
91         else
92             AddToTree(ch,code,strlen(code),&(*Root));
93     }
94 }
95
96 }
97 //????? ???? ???? ???? ???? ???? ????????? ? ? ????????? ???? ????????? ?????.
98 void AddToTree(char ch[], char code[], int codelen, BinTreePointer *Root)
99 {
100     int i = 0;
101     BinTreePointer TempPtr,p = *Root;
102
103     while(i < codelen)
104         if(code[i] == '0')
105         {
106             if(p->LChild == NULL)
107             {
108                 TempPtr = (BinTreePointer)malloc(sizeof (struct BinTreeNode));
109                 strcpy(TempPtr->Data.data,"*");
110                 TempPtr->Data.dummy = '*';
111                 TempPtr->LChild=NULL;
112                 TempPtr->RChild=NULL;
113                 p->LChild=TempPtr;
114             }
115             i++;
116             p = p->LChild;
117         }
118         else
119             if(code[i] == '1')
120             {
121                 if(p->RChild == NULL)
122                 {
123                     TempPtr = (BinTreePointer)malloc(sizeof (struct BinTreeNode));
124                     strcpy(TempPtr->Data.data,"*");
125                     TempPtr->Data.dummy = '*';
126                     TempPtr->LChild=NULL;
127                     TempPtr->RChild=NULL;
128                     p->RChild = TempPtr;
129                 }
130                 i++;
131                 p = p->RChild;
132             }

```

```

133         //?????? ???? ?? ??????? ??? ???? ???? ??? ??????.
134         strcpy(p->Data.data,ch);
135     }
136
137 void Decode(BinTreePointer Root, FILE* MessageFile)
138 {
139     char bit;
140     BinTreePointer p;
141
142     while(TRUE)
143     {
144         p = Root;
145
146         while(p->LChild != NULL || p->RChild != NULL)
147         {
148             bit = fgetc(MessageFile);
149
150             if(bit != EOF)
151             {
152                 printf("%c",bit);
153
154                 if(bit == '0')
155                     p = p->LChild;
156                 else
157                     if(bit == '1')
158                         p=p->RChild;
159                     else
160                         if(bit != '\n')
161                             printf("MH APODEKTO bit:%c \n",bit);
162
163             }
164             else break;
165         }
166         if(bit != EOF ) printf("----%s\n",p->Data.data);
167         else break;
168     }
169 }

```