

```

1  /*-----+
2  |   File   : errorcode_ass.cpp
3  |   purpose:
4  |   This program targets on finding the maxmim code size. User input the
5  |   codewordsize and minimum hamming distance. The program will perform
6  |   exhaustive search to find all the codewords that have minimum Hamming
7  |   distance. The processing time may quite long. ie, this program are not
8  |   for the general user.
9  |   Output : 1) Maximum code size
10 |              2) the minimum codewords
11 |
12 |   Dillian 23/7/04
13 |-----*/
14 #include <iostream.h>
15 #include "Math.h"
16
17 int minDistance;           //minimum hamming distance
18 int codewordSize;         //codeword size
19 int candidate[65536];     //codeword in 10-base digit form
20 int memberCount[65536];   //candidate[?] as the first round member
21                           //how many member of that code
22 bool isPass[65536];       //indicate this candidate is pass or not
23 bool bits_cand[65536][16]; //candidate in 8-bits form
24
25
26 //participate candidate have a new chance
27 void reset(void)
28 {
29     int i;
30     for(i=0;i<pow(2,codewordSize);i++)
31     {
32         isPass[i]=true;
33     }
34 }
35
36 //initialization
37 void init(void)
38 {
39     int i,j;
40     int l;
41     int a=0;
42
43     //extract parameter
44     cout << "Codeword size (3-16bits) ? : ";
45     cin>>codewordSize;
46     cout << "Minimum Hamming distance ? : ";
47     cin  >> minDistance;
48
49     //decleare all the candidate
50     for(i=0;i<65536;i++)
51     {
52         candidate[i] = i;
53         isPass[i] = false;
54         memberCount[i]=0;
55     }
56
57     //all is false, only the participate candidate
58     for(i=0;i<pow(2,codewordSize);i++)
59     {
60         isPass[i] = true;
61     }
62
63     //integer to binary
64     for(j=0;j<pow(2,codewordSize);j++)
65     {
66         a = candidate[j];
67         for(i=codewordSize-1;i>=0;i--)
68         {
69             l=a%2;           //modules
70             a=a/2;
71             if(l)
72                 bits_cand[j][i]=true;
73             else
74                 bits_cand[j][i]=false;
75         }
76     }

```

```

77 }
78
79 //print the date to screen
80 void print2Screen(void)
81 {
82     int i,j;
83
84     for(j=0;j<pow(2,codewordSize);j++)
85     {
86
87         if(isPass[j]==true)
88         {
89             cout<<"\n " <<j <<" ";
90             for(i=0;i<codewordSize;i++)
91             {
92                 if(bits_cand[j][i]==true)
93                     cout<<"1";
94                 else
95                     cout<<"0";
96             }
97         }
98     }
99 }
100
101 /*
102
103     Determine which candidate will reject, the routine will compare all the
104     candidates with the reference cand. If the min hamming distance cannot
105     fullfill, that candidate will be rejected. Of course, the reference, cand
106     itself will not rejected.
107 */
108 void checkReject(int cand)
109 {
110     int d,i,j;
111     for(j=0;j<pow(2,codewordSize);j++)
112     {
113         //find the codeword distance
114         d = 0;
115         for(i=0;i<codewordSize;i++)
116         {
117             if (!(bits_cand[cand][i]==bits_cand[j][i]))
118                 d++;
119         }
120
121         //others need to verify satisfy the min hamming distance or not
122         if (d < minDistance)
123             isPass[j] = false;
124     }
125     //itself must be true, so need to add itself to table
126     isPass[cand] = true;
127 }
128
129
130 /*
131     This routine will find all the winner with a give assigned winner
132 */
133 void playGameWithAssignedWinnerNo1(int winnerNo1)
134 {
135     int m,cand;
136
137     /*ROUND 1*/
138     cand = winnerNo1;
139     checkReject(cand);
140
141     /*ROUND 2..ROUND3..ROUND4.....*/ //That part search from 0, for
142     m = 0; //finding the second ref
143     while (m < pow(2,codewordSize))
144     {
145         //find the second true candidate
146         while ((isPass[m] == false) && (m < pow(2,codewordSize)))
147         {
148             m++;
149         }
150
151         //execute rejection, when we have second ref candidate
152         if (m < pow(2,codewordSize))

```

```

153         {
154             if (isPass[m] = true)
155             {
156                 cand = candidate[m];
157                 checkReject(cand);
158             }
159         }
160         //next
161         m = m + 1;
162     }
163 }
164
165 /*
166 Main program to find the maximum code size and the minimum codeword.
167 */
168 int main()
169 {
170     int i,m,maxMemberCount;
171
172     //Title
173     cout << "Error detecting code base on the principle of minimum Hamming distance" <<endl;
174     cout << "*Assistance program"<<endl;
175
176     while(1)
177     {
178
179         init();
180         maxMemberCount=0;
181
182         //start calculation
183         for(m=0;m<pow(2,codewordSize);m++)
184         {
185             //all candidates must have one time to be the winner no.1 and therefore
186             //will not miss to find the maximum number of codewords
187             //playGameWithAssignedWinnerNo1(candidate[m]);
188             playGameWithAssignedWinnerNo1(m);
189
190             //count the number of member of one code
191             for(i=0;i<pow(2,codewordSize);i++)
192             {
193                 if (isPass[i] == true)
194                     memberCount[m] = memberCount[m] + 1;
195             }
196             //max member size
197             if(memberCount[m]>maxMemberCount)
198                 maxMemberCount = memberCount[m];
199
200             //Final OUTPUT
201             // if(memberCount[m]>16)
202             //     cout<<"\nround : " <<m<<" member size : " <<memberCount[m];
203             // print2Screen();
204
205             reset();
206         }
207
208
209         cout<<"\nComplete" <<endl;
210
211         /*OUTPUT*/
212         //max member size, and min codeword
213         cout<<"codeword size : " <<codewordSize <<" Min distance : " <<minDistance<<endl;
214         cout<<"MaxMember size : " <<maxMemberCount;
215         for(i=0;i<pow(2,codewordSize);i++)
216         {
217             if(memberCount[i]==maxMemberCount)
218             {
219                 cout<<" Min codeword : " <<i<<endl;
220                 i=pow(2,codewordSize);
221             }
222         }
223
224
225         // cout<<"Press any key to close the program"
226         // cin>>input;
227
228     }

```

```
229     return 0;  
230 }
```