

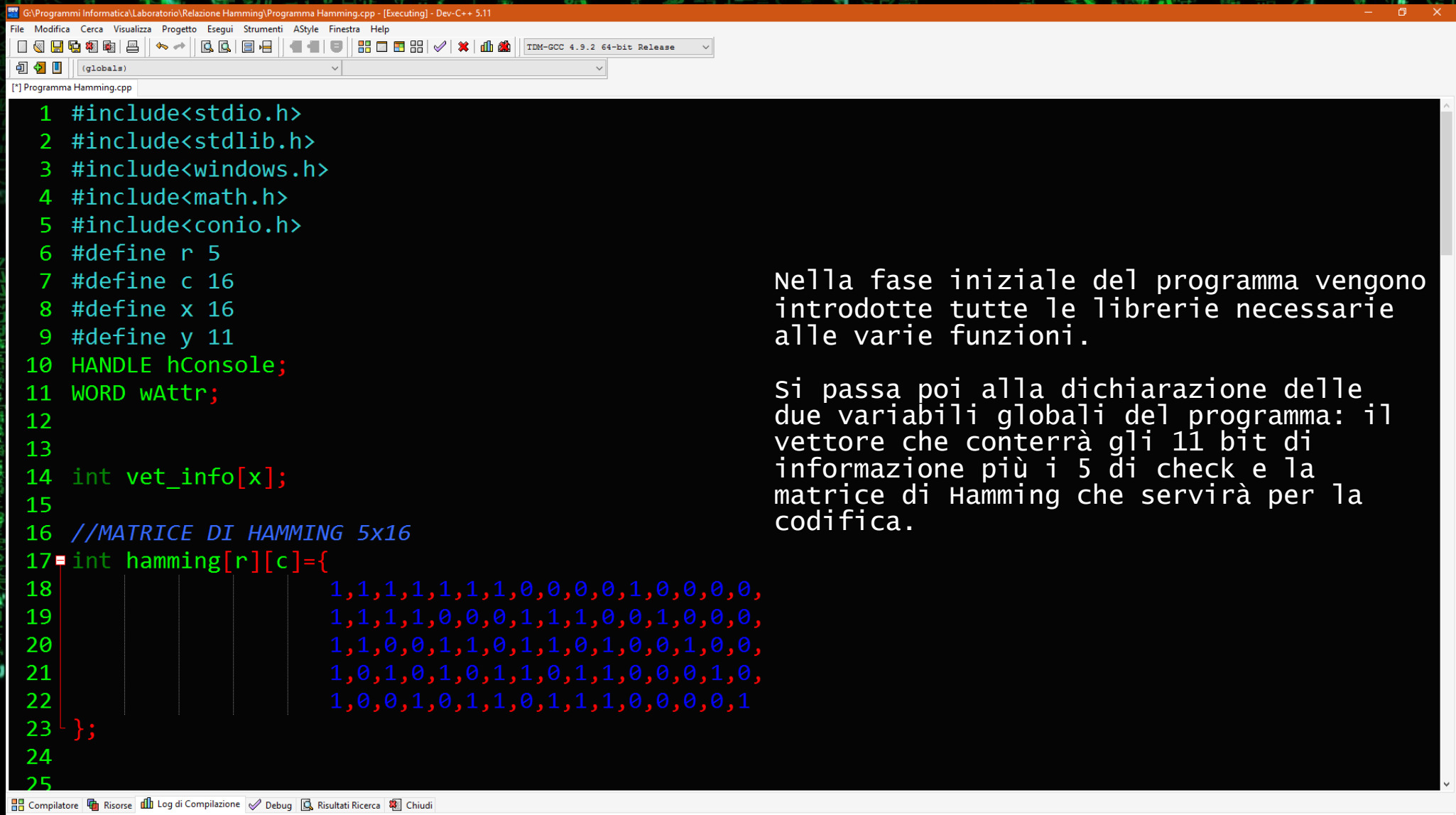
# Relazione – Tecnica di Hamming

La tecnica di Hamming, che prende il nome dal suo ideatore Richard Hamming (1915-1998), consiste in un algoritmo che ha la caratteristica di rilevare uno o più bit errati durante una trasmissione di 11 bit e di correggerli nella maggior parte dei casi servendosi di bit di ridondanza, creati appositamente grazie a una matrice, per il controllo del messaggio.

Nelle seguenti slide ne verrà illustrato il funzionamento tramite un programma scritto in linguaggio C che simula la codifica del messaggio da parte del mittente.



# Codice del Mittente – Librerie e Variabili Globali



```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<windows.h>
4 #include<math.h>
5 #include<conio.h>
6 #define r 5
7 #define c 16
8 #define x 16
9 #define y 11
10 HANDLE hConsole;
11 WORD wAttr;
12
13
14 int vet_info[x];
15
16 //MATRICE DI HAMMING 5x16
17 int hamming[r][c]={
18     1,1,1,1,1,1,1,0,0,0,0,1,0,0,0,0,
19     1,1,1,1,0,0,0,1,1,1,0,0,1,0,0,0,
20     1,1,0,0,1,1,0,1,1,0,1,0,0,1,0,0,
21     1,0,1,0,1,0,1,1,0,1,1,0,0,0,1,0,
22     1,0,0,1,0,1,1,0,1,1,1,0,0,0,0,1
23 };
24
25
```

Nella fase iniziale del programma vengono introdotte tutte le librerie necessarie alle varie funzioni.

Si passa poi alla dichiarazione delle due variabili globali del programma: il vettore che conterrà gli 11 bit di informazione più i 5 di check e la matrice di Hamming che servirà per la codifica.

# Codice del Mittente – Funzioni I

```
G:\Programmi Informatica\Laboratorio\Relazione Hamming\Programma Hamming.cpp - Dev-C++ 5.11
File Modifica Cerca Visualizza Progetto Esegui Strumenti AStyle Finestra Help
TDM-GCC 4.9.2 64-bit Release
[global]
[*] Programma Hamming.cpp
25
26
27 // FUNZIONI
28
29 void rosso()
30 {
31     hConsole=GetStdHandle(STD_OUTPUT_HANDLE);
32     wAttr=FOREGROUND_RED;
33     SetConsoleTextAttribute(hConsole,wAttr);
34 }
35
36 void verde()
37 {
38     hConsole=GetStdHandle(STD_OUTPUT_HANDLE);
39     wAttr=FOREGROUND_GREEN;
40     SetConsoleTextAttribute(hConsole,wAttr);
41 }
42
43 void blu()
44 {
45     hConsole=GetStdHandle(STD_OUTPUT_HANDLE);
46     wAttr=FOREGROUND_BLUE;
47     SetConsoleTextAttribute(hConsole,wAttr);
48 }
49
Compilatore Risorse Log di Compilazione Debug Risultati Ricerca Chiudi
```

Prima di presentare il programma vero e proprio, osserviamo le funzioni di cui è composto.

In questa sezione si trovano delle semplici funzioni che servono a cambiare il colore del testo all'occorrenza per pura estetica.



# Codice del Mittente – Funzioni II

G:\Programmi Informatica\Laboratorio\Relazione Hamming\Programma Hamming.cpp - Dev-C++ 5.11

File Modifica Cerca Visualizza Progetto Esegui Strumenti AStyle Finestra Help

TDM-GCC 4.9.2 64-bit Release

(globals)

[\*] Programma Hamming.cpp

```
49
50 void visualizza()
51 {
52     int i,j;
53     for(i=0;i<r;i++)
54     {
55         for(j=0;j<c;j++)
56             printf("%d",hamming[i][j]);
57
58         printf("\n");
59     }
60 }
61
62 void carica_info()
63 {
64     int i;
65     for(i=0;i<y;i++)
66     {
67         do
68         {
69             printf("inserire il bit di informazione %d: \n",i+1);
70             scanf("%d",&vet_info[i]);
71             if(vet_info[i]>1)
72                 printf("inserimento non valido\n reinserire il valore:\n");
73         }
74         while(vet_info[i]>1);
75     }
76 }
77
```

Funzioni: “visualizza” e  
“carica\_info”.

La prima ha il compito di  
stampare a video la matrice di  
Hamming all'occorrenza, mentre  
la seconda si occupa  
dell'inserimento degli 11 bit  
di informazione.

Compilatore Risorse Log di Compilazione Debug Risultati Ricerca Chiudi

# Codice del Mittente – Funzioni III

G:\Programmi Informatica\Laboratorio\Relazione Hamming\Programma Hamming.cpp - Dev-C++ 5.11

File Modifica Cerca Visualizza Progetto Esegui Strumenti AStyle Finestra Help

TDM-GCC 4.9.2 64-bit Release

(globals)

[\*] Programma Hamming.cpp

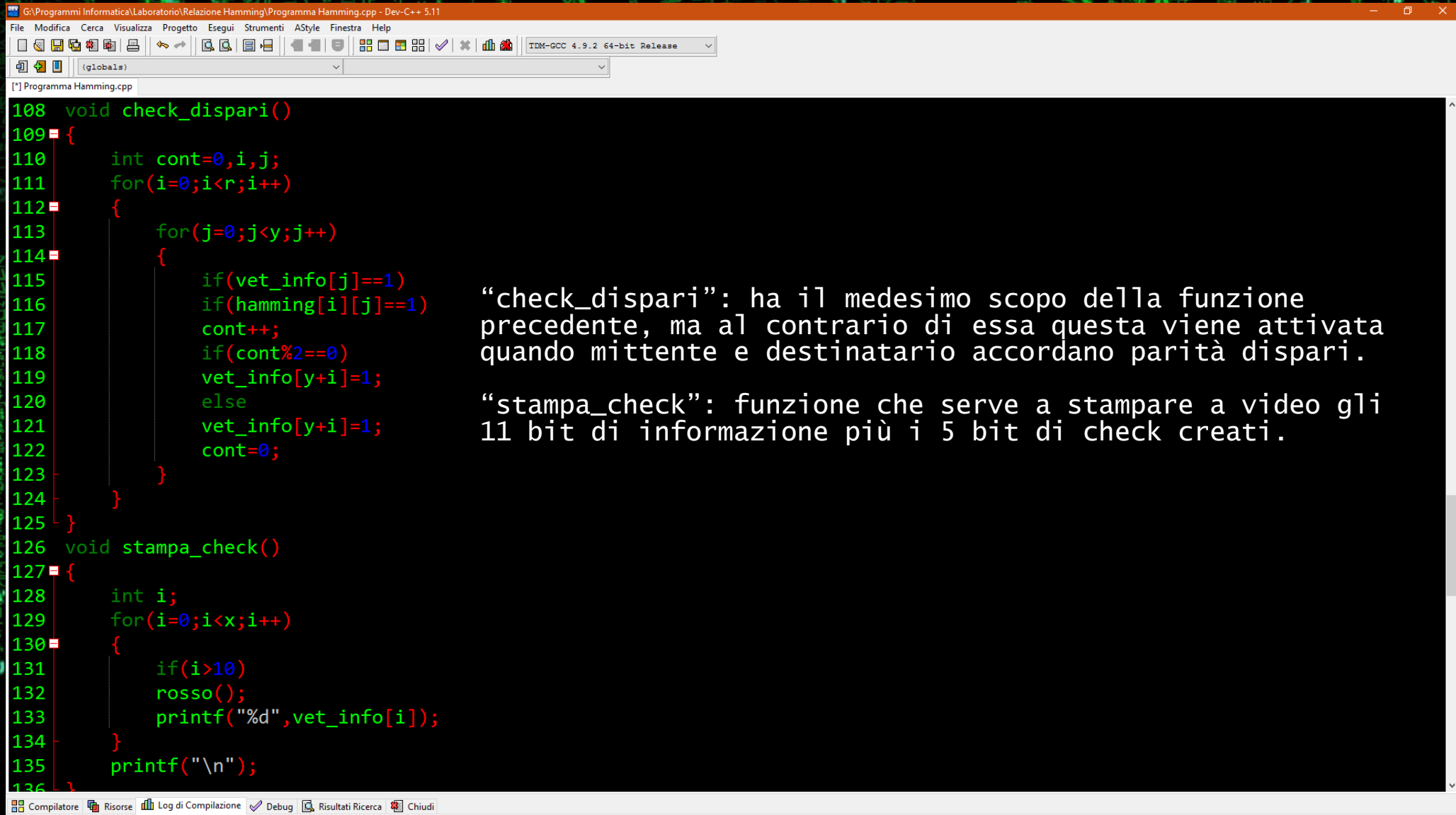
```
79
80 void stampa_info()
81 {
82     int i;
83     for(i=0;i<y;i++)
84         printf("%d",vet_info[i]);
85     printf("\n");
86 }
87
88 void check_pari()
89 {
90     int cont=0,i,j;
91     for(i=0;i<r;i++)
92     {
93         for(j=0;j<y;j++)
94         {
95             if(vet_info[j]==1)
96             if(hamming[i][j]==1)
97                 cont++;
98             if(cont%2==0)
99                 vet_info[y+i]=0;
100             else
101                 vet_info[y+i]=1;
102             cont=0;
103         }
104     }
105 }
106
107
```

“stampa\_info”: ha lo scopo di stampare a video gli 11 bit di informazione.

“check\_pari”: serve a creare il bit di check corretto in base alla parità del numero di bit valenti 1 accordata fra mittente e destinatario.  
In caso sia stata accordata parità pari, la funzione si occupa di creare un bit valente 1 o 0 in modo che la somma dei bit valenti 1, presenti negli 11 risultanti dall'operazione AND fra gli 11 bit di informazione e i corrispettivi incolonnati della riga della matrice, sia pari.

Compilatore Risorse Log di Compilazione Debug Risultati Ricerca Chiudi

# Codice del Mittente – Funzioni IV



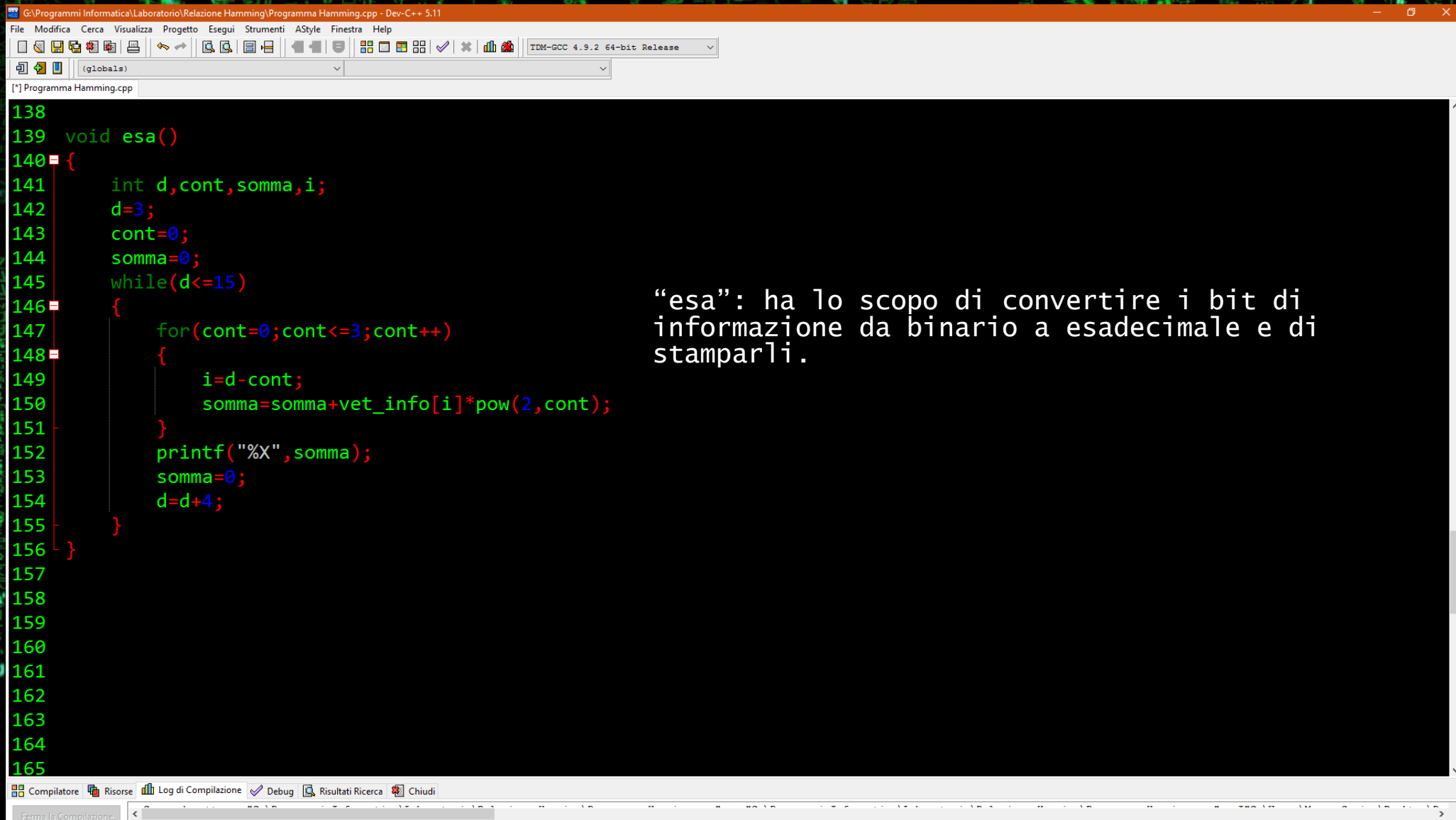
```
108 void check_dispari()
109 {
110     int cont=0,i,j;
111     for(i=0;i<r;i++)
112     {
113         for(j=0;j<y;j++)
114         {
115             if(vet_info[j]==1)
116             if(hamming[i][j]==1)
117                 cont++;
118             if(cont%2==0)
119                 vet_info[y+i]=1;
120             else
121                 vet_info[y+i]=1;
122             cont=0;
123         }
124     }
125 }
126 void stampa_check()
127 {
128     int i;
129     for(i=0;i<x;i++)
130     {
131         if(i>10)
132             rosso();
133         printf("%d",vet_info[i]);
134     }
135     printf("\n");
136 }
```

“check\_dispari”: ha il medesimo scopo della funzione precedente, ma al contrario di essa questa viene attivata quando mittente e destinatario accordano parità dispari.

“stampa\_check”: funzione che serve a stampare a video gli 11 bit di informazione più i 5 bit di check creati.



# Codice del Mittente – Funzioni V



The image shows a screenshot of a C++ IDE window titled "G:\Programmi Informatica\Laboratorio\Relazione Hamming\Programma Hamming.cpp - Dev-C++ 5.11". The window displays the source code for a function named `esa()`. The code is as follows:

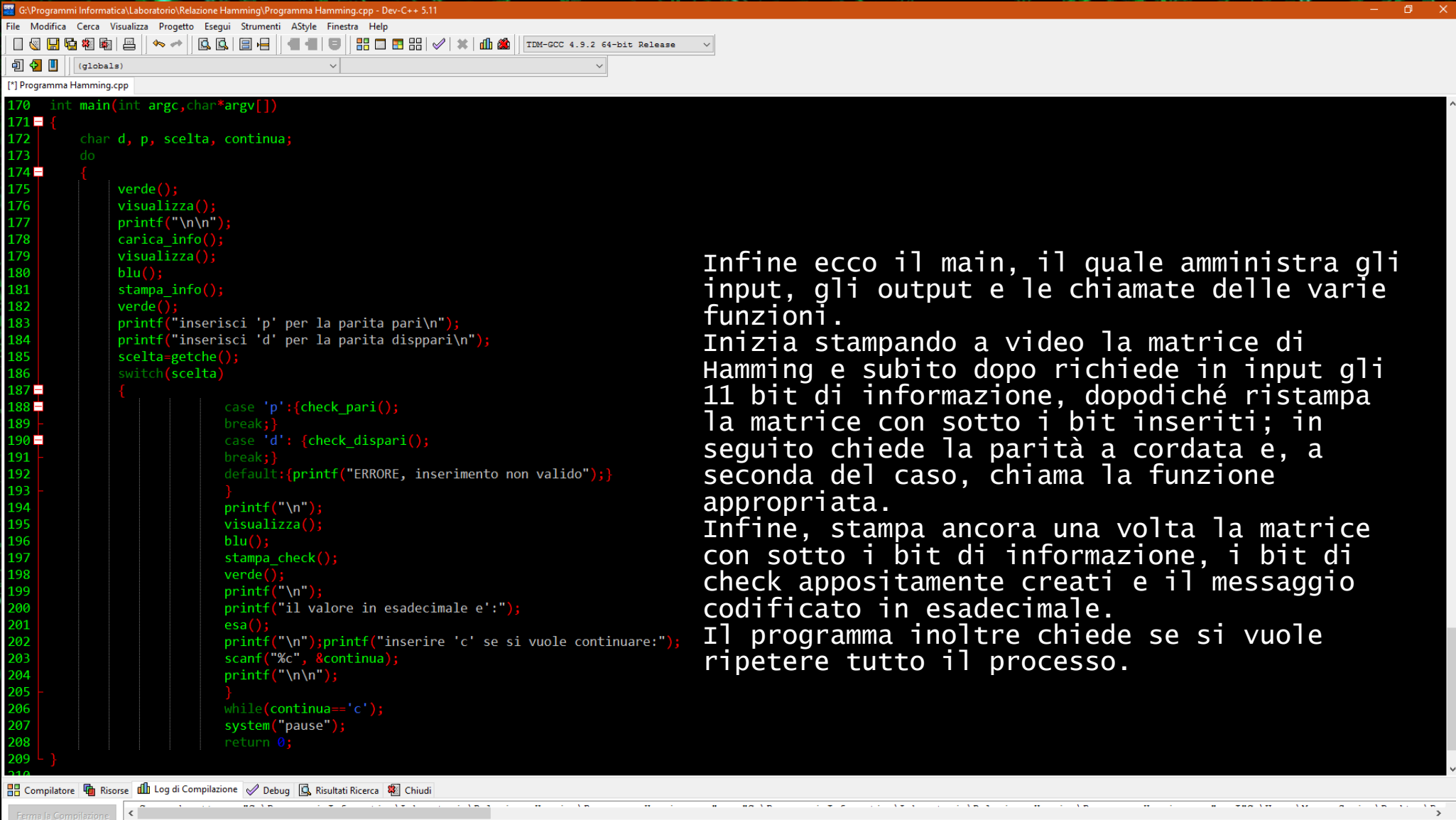
```
138
139 void esa()
140 {
141     int d, cont, somma, i;
142     d=3;
143     cont=0;
144     somma=0;
145     while(d<=15)
146     {
147         for(cont=0; cont<=3; cont++)
148         {
149             i=d-cont;
150             somma=somma+vet_info[i]*pow(2, cont);
151         }
152         printf("%X", somma);
153         somma=0;
154         d=d+4;
155     }
156 }
```

Below the code, there is a text box with the following explanation:

“esa”: ha lo scopo di convertire i bit di informazione da binario a esadecimale e di stamparli.

The IDE interface includes a menu bar (File, Modifica, Cerca, Visualizza, Progetto, Esegui, Strumenti, AStyle, Finestra, Help), a toolbar, and a status bar at the bottom with buttons for Compilatore, Risorse, Log di Compilazione, Debug, Risultati Ricerca, and Chiudi. A "Ferma la Compilazione" button is also visible in the bottom left corner.

# Codice del Mittente – Main



```
170 int main(int argc, char* argv[])
171 {
172     char d, p, scelta, continua;
173     do
174     {
175         verde();
176         visualizza();
177         printf("\n\n");
178         carica_info();
179         visualizza();
180         blu();
181         stampa_info();
182         verde();
183         printf("inserisci 'p' per la parita pari\n");
184         printf("inserisci 'd' per la parita dispari\n");
185         scelta=getche();
186         switch(scelta)
187         {
188             case 'p':{check_pari();
189             break;}
190             case 'd': {check_dispari();
191             break;}
192             default:{printf("ERRORE, inserimento non valido");}
193             }
194             printf("\n");
195             visualizza();
196             blu();
197             stampa_check();
198             verde();
199             printf("\n");
200             printf("il valore in esadecimale e':");
201             esa();
202             printf("\n");printf("inserire 'c' se si vuole continuare:");
203             scanf("%c", &continua);
204             printf("\n\n");
205         }
206         while(continua=='c');
207         system("pause");
208         return 0;
209     }
```

Infine ecco il main, il quale amministra gli input, gli output e le chiamate delle varie funzioni.

Inizia stampando a video la matrice di Hamming e subito dopo richiede in input gli 11 bit di informazione, dopodiché ristampa la matrice con sotto i bit inseriti; in seguito chiede la parità a cordata e, a seconda del caso, chiama la funzione appropriata.

Infine, stampa ancora una volta la matrice con sotto i bit di informazione, i bit di check appositamente creati e il messaggio codificato in esadecimale.

Il programma inoltre chiede se si vuole ripetere tutto il processo.



# Programma del Mittente

G:\Programmi\Informatica\Laboratorio\Relazione Hamming\Programma Hamming.exe

```
1111111000010000
11110001111001000
1100110110100100
1010101101100010
1001011011100001

inserire il bit di informazione 1:
1
inserire il bit di informazione 2:
1
inserire il bit di informazione 3:
1
inserire il bit di informazione 4:
1
inserire il bit di informazione 5:
1
inserire il bit di informazione 6:
1
inserire il bit di informazione 7:
1
inserire il bit di informazione 8:
1
inserire il bit di informazione 9:
1
inserire il bit di informazione 10:
1
inserire il bit di informazione 11:
1
1111111000010000
11110001111001000
1100110110100100
1010101101100010
1001011011100001
11111111111
inserisci 'p' per la parita pari
inserisci 'd' per la parita disppari
p
1111111000010000
11110001111001000
1100110110100100
1010101101100010
1001011011100001
1111111111100111

il valore in esadecimale e':FFE7
inserire 'c' se si vuole continuare:

Premere un tasto per continuare . . .
```

```
-----
Process exited after 9.489 seconds with return value 0
Premere un tasto per continuare
```

Possiamo infine osservare il funzionamento completo del programma, come descritto della slide precedente.

Il messaggio ora può essere trasmesso con i bit sovrabbondanti per il controllo da parte del destinatario di eventuali errori durante la comunicazione del messaggio.