

```

/* MDB2020
Data una stringa di 1024 bit generati casualmente e una sequenza di 8 bit (byte) inserita in input, contare
quante volte la sequenza è contenuta nella stringa (numero di occorrenze)
*/
#include <string>
#include <iostream>
#include <stdlib.h> // per la funzione rand()
#include <ctime> // per la funzione srand() che inilializza il seme della funzione rand()
#define n 1024
using namespace std;
bool sequenzaok(string s) // funzione che restituisce true se la sequenza è corretta
{
    if(s.length() !=8) return false;
    for(int i=0;i<8;i++)
    {
        if (s[i]!='0' && s[i]!='1') return false;
    }
    return true;
}
int main()
{
    char continui='s';
    while(1) // ciclo generale per ripetere eventualmente l'elaborazione-----
    {
        system("cls");
        cout<<"Riconoscitore di sequenze";
        // costruisco la stringa di 1024 bit generati casualmente
        string stringarandom="";
        srand(time(NULL));
        for(int i=0;i<n;i++)
        {
            int bitrandom=rand()%2; //genera un valore random compreso tra 0 e 1
            if(bitrandom==0)
            {
                stringarandom+="0";
            }
        }
    }
}

```

```

else
{
    stringarandom+="1";
}
}
cout<<"\n\nStringa di "<<n<<" bit"; //display stringa random
cout<<"\n\n"<<stringarandom;
string sequenza="";
do
{
    cout<<"\n\nInserisci la sequenza di 8 bit (X per uscire): ";
    cin>>sequenza;
    if(sequenza=="X" || sequenza=="x")break;
}while(!sequenzaok(sequenza)); // se la sequenza non è corretta, ovvero non è un byte...
int k=0; // contatore delle occorrenze
for(int i=0;i<n-8;i++)
{
    if(stringarandom.substr(i,8)==sequenza)k++;
}
cout<<"\n\nNumero occorrenze della sequenza "<<sequenza<<" :"<<k;
cout<<"\n\nContinui (S/N)? ";
cin>>continui;
if(continui=='N' || continui=='n') break;
} //chiude while(1)-----
} //chiude main()-----

```

```

Riconoscitore di sequenze

Stringa di 1024 bit

10101000100111011111001100000111011110111111111010010110000100010001001011101010010011001010010100100000101001010010101
11000001111010100101010001111000000001110110010001000111110111010010100010110010000010001000100011010111110100111101001
101001000011011100101100010001100110001001000100011011001010010011011001000001100100111100111100100101011000101001111001
001101101111101000100000110101001101001000111010011000000011101100110001000000110000011101110111010100010001000010101111
10110001010011001110100111010010011011011110010101011010110011111100101001110010001110110000111011001010010101110000001
1001111000100011100100111110011111010011110110001101110110100011111101110011101011010001100001100011111111010000011000
10010101001101101010001100011011100011000100010010110010101000100110100110000000101010100001010101010110000000100001001
110001100100110001100010110000110001100110000110001000111001111110011001101100111001100000001010011101100111000100010001
00110001000010111010110100101100001000001010010111011111011001

Inserisci la sequenza di 8 bit (X per uscire): 11010101

Numero occorrenze della sequenza 11010101 :1

Continui (S/N)?

```