

Esercizio "Calcolo checksum"

Un checksum è un valore utilizzato per verificare se un blocco di dati che viene trasferito tra due dispositivi subisce variazioni durante la trasmissione. Un checksum può essere calcolato utilizzando diversi algoritmi, alcuni particolarmente efficaci e sofisticati.

In questo esercizio si utilizza un algoritmo semplice e si fa riferimento alle operazioni eseguite nel dispositivo che trasmette:

1. Viene generato in modo random un blocco di dati di 1024 bit
2. prima di essere trasmesso, il blocco dati viene suddiviso in 128 parole di 1 byte (8 bit) ciascuna
3. per ciascun byte si calcola il numero decimale corrispondente (conversione da binario a decimale)
4. si calcola la *somma* tutti i numeri decimali
5. si calcola il checksum come $somma \% 256$, ovvero come resto della divisione della somma/256
6. il checksum calcolato è un numero compreso tra 0 e 255 che viene convertito in un byte (conversione da decimale a binario). Tale byte viene trasmesso in aggiunta agli altri 128 byte del blocco dati
7. nel dispositivo che riceve, viene calcolato il checksum relativo ai primi 128 byte. Se risulta uguale al checksum ricevuto significa che non ci state variazioni, se è diverso significa che uno o più bit sono variati durante la trasmissione

```
# CheckSum di un blocco di 1024 bit generato in modo random
import random
def conv_tobase10(byte):
    # conversione binario-->decimale
    somma = 0
    for i in range(0, 8, 1):
        if byte[i] == "1":
            somma += pow(2, 8 - i - 1)
    return somma
#
def conv_frombase10(numero):
    # conversione decimale-->binario
    stringabinaria = ""
    while numero > 0:
        stringabinaria = str(numero%2) + stringabinaria
        numero = int(numero/2)
    if len(stringabinaria)<8:
        zeri="00000000"
        stringabinaria=zeri[0:8-len(stringabinaria)]+stringabinaria
    return stringabinaria
#
print("Checksum di un blocco dati di 1024 bit")
# Genero 1024 numeri interi random compresi tra 0 e 1
# e costruisco quindi un blocco dati di 1024 bit
n = 1024
n1 = 0
n2 = 1
bloccodati = ""
for i in range(n):
    numerogenerato = random.randint(n1, n2)
    bloccodati += str(numerogenerato)
```

```

# display del blocco dati generato su 8 righe (128 bit per riga)
print()
print("Blocco di 1024 bit generato in modo random:")
for i in range(0, n, 128):
    print(bloccodati[i:i + 128])
# Suddivido il blocco dati in byte (8 bit) e converto ciascun byte
# nel corrispondente numero decimale. Sommo i valori decimali dei 128 byte
somma = 0
for i in range(0, n, 8):
    byte = bloccodati[i:i + 8]
    valoredecimale = conv_tobase10(byte)
    somma += valoredecimale
print()
print("Somma dei valori decimali dei 1024 bit: " + str(somma))
# Calcolo il checksum ovvero il resto della divisione della somma/256
checksum = somma % 256
print("Checksum: " + str(checksum))
# Converto il checksum decimale in binario e costruisco il byte di controllo
# da trasmettere insieme con gli altri 128 byte
print("Byte di Checksum: " + conv_frombase10(checksum))

```

Checksum di un blocco dati di 1024 bit

Blocco di 1024 bit generato in modo random:

```

10001001010110100010011101101111000000100000011100101001100001111001110010111100011010111000001010011001101111111011000101010011
1011100000110000110010110110010001111010000001111000010011110001011100010101011111001101110100100110100010000111010110101
00110001110000110000010010100010100001100010011000100110000011100110110000110010011010010010001111111000001101101011001000100110
0110010110000111010101101111101011111011110011100010001011000011000011110101110001010111010100110100111111000100001011000101
00101000110111011110011001100011010111010101010011000101001001001001001011010111001001110001101011111010011111100110110
011001101101001011111001101101010010101000111111001001100010010111000101001010111110010011000011101101010011100010100110100111
111110111001111001001101010001010111110100001000000000000100110011100011001000100111100101100110100011010000011011101111101011
111011101011100001101111010100000110001100010001111001100111001110100100000111011110010100110011010110000100111100001001010011

```

Somma dei valori decimali dei 1024 bit: 15705

Checksum: 89

Byte di Checksum: 01011001