**Compito #32: ripasso**

**A)** Data la sequenza S = {$, \*, $, !, @, !, !, @, \*, $, $, \*, @, !, \*, @}

1. calcolare l’entropia H(S) svolgendo tutti i passaggi

$ = 4/16 = 0,25

\* = 4/16 = 0,25

! = 4/16 = 0,25

@ = 4/16 = 0,25

H(Si) = -4(0,25\*(log (2)0,25)= -0.5\*(-4)=2 bit/sym

1. creare una codifica FLC con il numero minimo possibile

| $ | 00 |
| --- | --- |
| \* | 10 |
| ! | 01 |
| @ | 11 |

1. calcolare la distanza di Hamming tra tutti i simboli

$-\*=1

$-!=1

$-@=2

\*-!=2

\*-@=1

!-@=1

1. assumendo la codifica 2) e calcolata la codeword o stringa di bit che rappresenta l’intera sequenza S calcolare il bit di parità pari e dispari

codeword=00100001110101111000001011011011

Il bit di parità pari è 0

Il bit di parità dispari è 1

1. calcolare la checksum di S considerando la codifica 2) svolgendo tutti i passaggi

11 xor @ = 11 xor 11 = 00

00 xor ! = 00 xor 01 = 01

01 xor ! = 01 xor 01 = 00

11 xor \* = 11 xor 10 = 01

01 xor $ = 01 xor 00 = 01

01 xor $ = 01 xor 00 = 01

01 xor \* = 01 xor 10 = 11

11 xor @ = 11 xor 11 = 00

checksum=00

1. creare una codifica VLC di Huffman della sequenza S svolgendo tutti i passaggi
2. calcolare il rapporto di compressione tra il messaggio codificato con 2) e con 6)

compressione

**B)** Data la sequenza S = {!, @, @, %, $, @, \*, $, @, $, \*, \*, !, \*, !, $}

1. calcolare l’entropia H(S) svolgendo tutti i passaggi

! = 3/16 = 0,1875

@ = 4/16 = 0,25

% = 1/16 = 0,0625

$ = 4/16 = 0,25

\* = 4/16 = 0,25

H(Si) = -3(0,25\*(log (2)0,25) -(0.1875\*(log(2)0,1875) -(0.0625\*(log(2)0,0625)= 1,5+0,452+0.25=2,202 bit/sym

1. creare una codifica FLC con il numero minimo possibile di bit

| ! | 000 |
| --- | --- |
| @ | 001 |
| % | 010 |
| $ | 100 |
| \* | 111 |

1. calcolare la distanza di Hamming tra tutti i simboli

!-@=1

!-%=1

!-$=1

!-\*=3

@-%=2

@-$=2

@-\*=2

%-$=2

%-$=2

$-\*=2

1. assumendo la codifica 2) e data la codeword o stringa di bit che rappresenta l’intera sequenza S calcolare il bit di parità pari e dispari

codeword=0000010010100001111100001100001100111111000111000100

Il bit di parità pari è 1

Il bit di parità dispari è 0

1. calcolare la checksum di S considerando la codifica 2) svolgendo tutti i passaggi

! xor @ = 000 xor 001 = 001

001 xor @ = 001 xor 001 = 000

000 xor % = 000 xor 010 = 010

010 xor $ = 010 xor 100 = 110

110 xor @ = 110 xor 001 = 111

111 xor \* = 111 xor 111 = 000

000 xor $ = 000 xor 100 = 100

100 xor @ = 100 xor 001 = 101

101 xor $ = 101 xor 100 = 001

001 xor \* = 001 xor 111 = 110

110 xor 111 = 110 xor 11 = 001

001 xor ! = 001 xor 000 = 001

001 xor \* = 001 xor 111 = 110

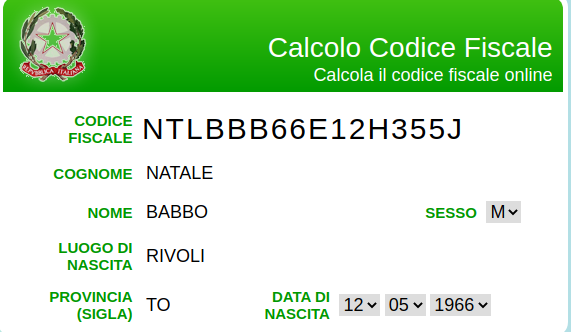
111 xor ! = 110 xor 000 = 110

110 xor $ = 110 xor 100 = 010

checksum=010

1. creare una codifica VLC di Huffman della sequenza S svolgendo tutti i passaggi
2. calcolare il rapporto di compressione tra il messaggio codificato con 2) e con 6)

**C)** Dato il seguente codice fiscale: NTLBBB66E12H355 calcolare il check digit e riportare tutti i passaggi



char\_dispari(NLB6E235)=(20,4,0,15,9,5,7,13)

char\_pari(TBB61H5)=(7,9,17,19,5,0,7,13)

char\_dispari(20,4,0,15,9,5,7,13)+ char\_pari(19,1,1,6,1,7,5)=113%26=9

checkdigit 9= J