

# Teorija informacije

# Laboratorijska vježba 2020/2021

# Korisničke upute - Koder za Hammingove kodove

Završiva datoteka se zove **hammingCoder** i namijenjena je za operacijski sustav Linux, Glibc libc, i računalnu arhitekturu AMD64 (još poznatu i kao x86-64).

Program prima dva argumenta naredbenog retka,  $n$  i  $k$ , tim redom (to su parametri Hammingovog koda); nakon pokretanja programa moguć je unos proizvoljno dugačkog niza bitova koji treba biti kodiran u kodne riječi koda zadanog parametrima  $n$  i  $k$ . Kontrolni bitovi se u kodnoj riječi nalaze na pozicijama 1, 2, 4, 8, i tako dalje; gdje brojanje pozicija bitova počinje od jedinice. Unošenjem parametra  $k$  koji ne odgovara parametru  $n$ , uzrokovan je uranjeni završetak programa, ali uz ispis parametra  $k$  koji odgovara danom  $n$ ; tako da nije potrebno na papiru ili u glavi računati odgovarajući  $k$ .

Ulazni niz bitova predstavljen je znakovima **0** i **1** koji mogu biti odvojeni s po volji mnogo znakova ASCII whitespacea (tab, razmak ili novi redak). Prekid unosa se može postići unosom bilo kojeg slova ili niza slova (na primjer, "**gotovo**") prije novog retka.

Nakon toga program prikazuje generirajuću matricu odabranog koda, te redom kodne riječi (uz izvorne bitove od kojih je svaka kodna riječ sačinjena)..

# Primjeri izvršavanja

Korisnički unos je **podebljan**.

Počnimo od najjednostavnijeg Hammingovog koda:

```

$ ./hammingCoder 3 1
Linear block code [n = 3, k = 1] (n = code word length) (k = number of source bits in each code word)
code rate = R(K) = 0.333333

Enter a message in bits (possibly separated by whitespace) to be Hamming coded using the chosen code parameters:

```

```
1 0 11 0 111 0
done
```

Input source message:  
101101110

The generator matrix for the code:

111

To encode the entire source input string into codewords, we divide the input string into parts of  $k$  or less bits, where the last part's last bits are padded with zeros. Each input part is multiplied with the generator to produce the corresponding codeword.

```

Input      1 bits: 1
Output: 111
Input      1 bits: 0
Output: 000
Input      1 bits: 1
Output: 111
Input      1 bits: 1
Output: 111
Input      1 bits: 0
Output: 000
Input      1 bits: 1
Output: 111
Input      1 bits: 1
Output: 111
Input      1 bits: 1
Output: 111
Input      1 bits: 0
Output: 000

```

Isprobajmo neke pogrešne načine zadavanja parametara, kako bismo vidjeli kako program na to reagira:

```
$ ./hammingCoder
coder: wrong number of arguments, start the program with two arguments, both natural numbers
```

```
$ ./hammingCoder 25 5
coder: wrong input for second argument (k), try 20
```

```
$ ./hammingCoder 25 20
Linear block code [n = 25, k = 20] (n = code word length) (k = number of source bits in each code word)
code rate = R(K) = 0.8
```

Enter a message in bits (possibly separated by whitespace) to be Hamming coded using the chosen code parameters:

[illegible]

Input source message:



[illegible]

To encode the entire source input string into codewords, we divide the input string into parts of  $k$  or less bits, where the last part's last bits are padded with zeros. Each input part is multiplied with the generator to produce the corresponding codeword.

[illegible]