1. Escreva um programa C que imprima o seu nome e os parâmetros de linha de comando, um por linha, na ordem inversa.

$ ./invcmd.c um 2 tres 4

#include <stdio.h>

int main (int argc, char \*argv[]) {

for (int cont = argc - 1; cont >= 0; cont--)

printf ("%s\n", argv[cont]);

return 0;

}

2. Escreva um programa C que receba um inteiro como parâmetro na linha de comando, e mostre quantos bits 1 tem a representação binária desse número.

$ ./contabits 20

exemplos : 20 = 2 (10100 em binário) / 65535 = 16 (1111111111111111 em binário)

Printf (“O numero %d possui %d bits em 1”, entrada, qtda);

#include <stdio.h>

int main() {

int n, r, cont;

cont = 0;

printf("Digite o numero: ");

scanf("%d", &n);

int num = n;

// Verifica se é o número 1 e imprime ele mesmo

if(n == 1) {

printf("%d\n", n);

}

// Caso o número seja maior do que 1, realiza divisões sucessivas por 2, imprimindo o seu resto para construção da forma binária.

else {

printf("Forma binaria: ");

do {

r = n % 2;

//printf("%d", r);

if(r == 1) cont++;

n = n / 2;

} while(n >= 1);

//printf("\n");

}

printf("O numero %d possui %d bits em 1.\n", num, cont);

}

**O QUE FAZER?** Converter o número de entrada para binário e contar a quantidade de números 1.