



QUESTÃO I - 3.0

Escreva um programa em C que seja capaz de gerar o email de uma instituição, conforme as seguintes regras. O administrador do recurso deve informar o domínio da empresa e o nome completo do usuário. O email deverá ser composto pelo primeiro nome do usuário, seguido de um ponto, último nome do usuário, caractere at ('@') e domínio da empresa (tudo em minúsculo). Por exemplo, se o administrador informar como entrada o domínio **inf027.ifba.edu.br** e como nome do usuário **Alan Mathison Turing**, o programa deverá gerar como saída **alan.turing@inf027.ifba.edu.br**.

QUESTÃO II - 4.0

Escreva um programa em C que seja capaz de 'compactar' uma string de até 32.000 caracteres. Para isso, o algoritmo deve contar o número de ocorrências consecutivas de um caractere e substituir estas ocorrências pelo caractere, seguido desta quantidade. Assim, por exemplo, uma string "AAAAAABCCCCC", deveria gerar uma string que represente os pares {<A,6>;<B,1>; <C;5>}. Assim, o algoritmo de compactação deve gerar como saída uma (também) string que seja composta por uma sequência de substrings formadas pelo caractere e o equivalente ASCII de sua quantidade de ocorrências. No caso em questão a string gerada deveria ter os seguintes caracteres:

'A'	ACK	'B'	SOH	'C'	ENQ	'\0'
-----	-----	-----	-----	-----	-----	------

O seu algoritmo deve considerar que o maior valor ASCII de um caractere é 255. Assim, para sequências de um mesmo caractere, cujo tamanho seja maior que 255, o algoritmo deverá gerar dois pares com o mesmo carácter consecutivamente. Por exemplo, caso exista uma sequência de 320, numerais '5' ("555555... -repete o numeral 5 trezentos e dez vezes- ...5555", a saída deveria ser a string {<5,255>;<5,65>;}, o que equivale à string:

'5'	'\y'	'5'	'A'	'\0'
-----	------	-----	-----	------

A string gerada deverá ser impressa.

QUESTÃO III - 3.0

No contexto das telecomunicações, **paridade** refere-se ao número de bits '1' de um determinado número binário. Para assinalar a paridade, é adicionado, no final ou no início de uma sequência binária, um dígito binário de paridade.

Por exemplo, considere a sequência "0000". Esta sequência, já possui um número par de bits 1 e assim para manter a paridade, acrescenta-se um bit 0, a sequência: "00000". Por sua vez a sequência "1011", com número três bits 1, precisa de um novo bit ligado para obter a paridade: "10111".

A paridade é vulgarmente utilizada para detectar erros nas transmissões, já que o seu cálculo é extremamente simples. Por exemplo, se for anexado um **bit de paridade** extra a cada byte transmitido, um erro pode ser detectado se a paridade do byte não coincidir com o bit de paridade. Para um determinado protocolo de telecomunicações, se definiu um quadro composto de uma matriz de bits (4x4). Contudo, para a transmissão, o protocolo faz uso de uma matriz 5x5, que acrescenta um bit de

paridade a cada linha, e a cada coluna. Ao final acrescenta-se um bit de paridade, a linha / coluna extra (posição 4x4). Veja a tabela abaixo.

Escreva um programa em C que leia uma matriz 4x4 de bits, calcule e imprima a matriz a ser transmitida acrescida com os bits de paridade.

0	0	0	0	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1