

1º Exercício de C – Base 91 (trabalho em grupo)

O formato de representação de caracteres conhecido como Base91 é utilizado para permitir que arquivos binários como fotos e executáveis sejam transferidos por e-mail, entre computadores com processadores incompatíveis, sistemas operacionais diferentes ou protocolos de comunicação com 8 ou 7 bits. O formato Base91 usa 91 caracteres visíveis da tabela ASCII, apenas não sendo utilizados os caracteres “-“ (hífen), “\” (contra barra) e “'” (apostrofe). O algoritmo de codificação divide os dados de entrada em blocos de 13 bits. Cada bloco, denominado X, é mapeado em um caracter na base 91, denominado Y, concatenando esses bits e gerando um inteiro de 13 bits. Esse caracter Y é mapeado como dois caracteres ASCII, segundo a fórmula:

$X \rightarrow (Y1, Y2)$ , onde  $X \in [1..8191]$

$Y1 = X // 91$ , onde // significa divisão inteira;

$Y2 = X \% 91$ , onde % significa módulo

Uma codificação para Y1 e Y2 pode obtida entrando com esses valores como subscritos na tabela a seguir. Por exemplo, seja  $X = 3274$  então  $Y1 = 35$  e  $Y2 = 89$ . Portanto Y1 e Y2 serão mapeados, respectivamente como “j” e “<”.

0	<b>A</b>	13	<b>N</b>	26	<b>a</b>	39	<b>n</b>	52	<b>0</b>	65	<b>&gt;</b>	78	<b>%</b>
1	<b>B</b>	14	<b>O</b>	27	<b>b</b>	40	<b>o</b>	53	<b>1</b>	66	<b>?</b>	79	<b>&amp;</b>
2	<b>C</b>	15	<b>P</b>	28	<b>c</b>	41	<b>p</b>	54	<b>2</b>	67	<b>@</b>	80	<b>(</b>
3	<b>D</b>	16	<b>Q</b>	29	<b>d</b>	42	<b>q</b>	55	<b>3</b>	68	<b>[</b>	81	<b>)</b>
4	<b>E</b>	17	<b>R</b>	30	<b>e</b>	43	<b>r</b>	56	<b>4</b>	69	<b>]</b>	82	<b>*</b>
5	<b>F</b>	18	<b>S</b>	31	<b>f</b>	44	<b>s</b>	57	<b>5</b>	70	<b>^</b>	83	<b>+</b>
6	<b>G</b>	19	<b>T</b>	32	<b>g</b>	45	<b>t</b>	58	<b>6</b>	71	<b>_</b>	84	<b>,</b>
7	<b>H</b>	20	<b>U</b>	33	<b>h</b>	46	<b>u</b>	59	<b>7</b>	72	<b>`</b>	85	<b>.</b>
8	<b>I</b>	21	<b>V</b>	34	<b>i</b>	47	<b>v</b>	60	<b>8</b>	73	<b>{</b>	86	<b>/</b>
9	<b>J</b>	22	<b>W</b>	35	<b>j</b>	48	<b>w</b>	61	<b>9</b>	74	<b> </b>	87	<b>:</b>
10	<b>K</b>	23	<b>X</b>	36	<b>k</b>	49	<b>x</b>	62	<b>!</b>	75	<b>}</b>	88	<b>;</b>
11	<b>L</b>	24	<b>Y</b>	37	<b>l</b>	50	<b>y</b>	63	<b>#</b>	76	<b>~</b>	89	<b>&lt;</b>
12	<b>M</b>	25	<b>Z</b>	38	<b>m</b>	51	<b>z</b>	64	<b>\$</b>	77	<b>"</b>	90	<b>=</b>

Se o arquivo não possuir um número múltiplo de 13 bits, o último bloco de bits deve ser completado com n bits com valor zero, onde  $n = 1, \dots, 12$ , gerando um caracter na base 91, denotado, digamos  $Y_f$ . Após  $Y_f$  deve ser gerado mais um caracter base 91, denotado  $\phi_n$ , onde  $\phi_n = 8191 + n$ . Tanto  $Y_f$  quanto  $\phi_n$  vão gerar, cada um deles, dois caracteres Y1 e Y2, segundo a fórmula acima. O final do arquivo deve ser marcado com a sequência “=” e “=”.

**1:** Faça um DCM para uma programa capaz de, a partir da leitura de um arquivo binário, converte-lo para o formato Base91, e vice versa (converter um Base91 em um arquivo binário) (2,0)

**2:** implemente em C99 o programa especificado a) para gerar Base91 corretamente (4,0) a partir de um arquivo binário; b) decodifique um arquivo texto em Base91 e gere o arquivo binário correspondente (2,0); c) receba parâmetros via linha do comando com operação a ser realizada (codificar ou decodificar e nomes dos arquivos) ou, caso não sejam passados, solicite os parâmetros aos usuários (1,0) ; e d) possua documentação interna (1,0). Ao programa que apresente o melhor algoritmo (considerando performance e escalabilidade) será atribuído um bônus de até 2,0 pontos.