



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Projeto e Desenvolvimento de Algoritmos
AD2 2º semestre de 2009.

Nome –

Assinatura –

1ª questão (5.0 pontos): Manipulação de Strings

Tarefa

Escreva um programa que receba um vetor de strings e determine o número de ocorrências de um dado caractere (fornecido pelo usuário) no conjunto de strings.

Parte do código da solução é visto a seguir.

```
início
    imprima 'Entre com o número de strings: '
    leia dim

    para i ← 1 até dim faça
        leia nomes[i]
    próximo i

    imprima 'Entre com o caractere a ser procurado: '
    leia ch

    // o seu código aqui...
fim
```

Exemplo:

Para o conjunto de entrada a seguir, seu programa deveria imprimir o número sete como saída.

```
Entre com o número de strings: 6
'brandon'
'osvaldo'
'buffy'
'jimbo'
'scooby doo'
'bobby'
Entre com o caractere a ser procurado: b
```

Observação:

Para a construção de sua solução, considere a existência das funções **charAt()** e **length()** cuja documentação é mostrada a seguir.

charAt(string, indice)

Retorna o caractere na posição especificada pelo índice. O índice pode variar de 1 até **length(string)**. O primeiro caractere da sequência tem o índice 1, o seguinte o índice 2 e assim por diante.

Exemplo:

```
imprima length('CEDERJ', 2)
```

imprimiria o caractere 'E'

length(string)

Retorna o tamanho da string. O tamanho é igual ao número de caracteres na string.

Exemplo:

```
imprima length('CEDERJ')
```

imprimiria o inteiro 6

2ª questão (5.0 pontos): Convertendo para base 2

Tarefa

Escreva um programa que leia uma sequência de números inteiros na base 10 e imprima cada um dos números convertidos para a base 2 e a maior sequência de bits 1 que o número binário correspondente contém. Considere que o número na base 10 é menor do que $2^{32}-1$, ou seja, ele tem menos de 32 bits.

Entrada:

A entrada contém vários números inteiros positivos menores que $2^{32}-1$. Cada número será fornecido em uma linha separada. O programa termina quando o usuário fornecer um número inteiro negativo.

Saída:

Para cada número da entrada seu programa deve produzir quatro linhas:

- Na primeira linha o programa deve imprimir o caso de teste no formato **Número i**, onde **i** é a posição do número na sequência de entrada, começando em 1.
- Na segunda linha deve ser impresso o número convertido para a base 2. Zeros à esquerda não devem ser impressos.
- Na terceira linha o seu programa deve imprimir a maior sequência de 1's do número em binário.
- A quarta linha deve ser deixada em branco.

Exemplos de entrada e saída:

Exemplo de Entrada	Saída para o exemplo de Entrada
6	Número 1
25	110
123456	2
14	
-1	Número 2
	11001
	2
	Número 3
	11110001001000000
	4
	Número 4
	1110
	3

Dica:

Uma versão inicial do algoritmo para converter um número inteiro positivo da base 10 para a base 2 é mostrada a seguir. Observe que o algoritmo está escrito em uma linguagem de pseudo código livre, mas sua solução terá de usar PETEQS.

```
leia numero
enquanto numero / 2 for diferente de 0 faça
    Resto = numero mod 2
    Guarde o Resto
    numero = numero / 2
fim do enquanto
imprima os restos na ordem inversa a que foram produzidos
```

Você pode obter mais detalhes em

http://pt.wikipedia.org/wiki/Convers%C3%A3o_de_base_num%C3%A9rica



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Projeto e Desenvolvimento de Algoritmos
AD2 2º semestre de 2009.

GABARITO

1ª questão (5.0 pontos): Manipulação de Strings

```
início
  imprima 'Entre com o número de strings: '
  leia dim

  para i ← 1 até dim faça
    leia nomes[i]
  próximo i

  imprima 'Entre com o caractere a ser procurado: '
  leia ch

  conta ← 0
  para i ← 1 até dim faça
    para j ← 1 até length(nomes[i]) faça
      se charAt(nomes[i], j) = ch então
        conta ← conta + 1
      fim se
    próximo j
  próximo i

  imprima 'Número de ocorrências ', conta
fim
```

2ª questão (5.0 pontos): Convertendo para base 2

```
inicio
  contaNumeros ← 0
  acabou ← falso
  enquanto não acabou faça
    leia numero
    se numero < 0 então
      acabou ← verdadeiro
    senão
      contals ← 0
      maiorSequencia ← 0
      bit ← 32
      enquanto bit >= 1 faça
        se numero <> 0 então
          q ← numero/2
          r ← numero mod 2
          resultado[bit] ← r
          numero ← q
          se r=1 então
            contals ← contals + 1
          senão
            se contals > maiorSequencia então
              maiorSequencia ← contals
            fim se
            contals ← 0
          fim se
        senão
          resultado[bit] ← 0
        fim se
        bit ← bit -1
      fim enquanto
      imprima 'Numero', contaNumeros
      para bit ← 1 até 32 faça
        imprima resultado[bit]
      proximo bit
      se contals > maiorSequencia então
        maiorSequencia ← contals
      fim se
      imprima maiorSequencia
    fim se
  fim enquanto
fim
```