**FÁCEIS**

**1.** Considerando que strings sempre terminam num caractere nulo, determinar o comprimento de uma string dada como entrada.

**2.** Dada uma string como entrada determinar a soma do código ascii de seus caracteres.

**3.** Escreva um programa que solicita dois números ponto flutuante do usuário e uma operação (caractere) que ele deseja realizar. O programa deve realizar a operação selecionada pelo usuário e apresentar na tela. Cuidado com divisões por zero.

**4.**Escreva um programa que exibe um quadrado sólido de tamanho **n** formado por um símbolo **s**. O tamanho do lado do quadrado e o símbolo devem ser informados pelo usuário.

|  |
| --- |
| **Exemplo: n=2, s= \*** |
| **\*\***  **\*\*** |

**5.** Escreva um programa que recebe lê um número inteiro, representando uma nota quantitativa de um aluno (entre 0 e 20), e imprime uma cadeia de caracteres que corresponde a uma nota qualitativa, de acordo com a tabela abaixo:

|  |  |
| --- | --- |
| Nota  quantitativa | Nota  qualitativa |
| 18 a 20 | Muito Bom |
| 14 a 17 | Bom |
| 10 a 13 | Suficiente |
| 5 a 9 | Mediocre |
| 0 a 4 | Ruim |

**6.** Faça um programa que receba palavras separadas por vários espaços em branco e apresente essas palavras separadas por somente um espaço.

|  |  |
| --- | --- |
| **Exemplo** | |
| **Entrada** | LABORATORIO DE PROGRAMACAO I |
| **Saída** | LABORATORIO DE PROGRAMACAO I |

**7.** Faça um programa que receba uma frase, calcule e mostre a quantidade de vogais da frase digitada. Também deve ser calculado e exibido a quantidade de palavras.

**8.** Faça um programa para criptografar uma frase, trocando as consoantes por # e

invertendo a frase.

Ex: Entrada: Sua casa

Saída: a#a# au#

**9.** Faça um programa que receba um caractere e informe se ele é **letra maiúscula**, **letra minúscula** ou **não é uma letra**. Lembre-se que caracteres são representados internamente por números.

**10.** Converter em caixa alta (caracteres alfabéticos em maiúsculo) uma string dada como entrada.

**MÉDIAS**

**11.** Faça um programa que receba uma frase e imprima a mesma frase com as palavras escritas ao contrário.

|  |  |
| --- | --- |
| **Exemplo** | |
| **Entrada** | LABORATORIO DE PROGRAMACAO I |
| **Saída** | OIROTAROBAL ED OACAMARGORP I |

**12.** Dada uma string ascii como entrada, substituir todos os caracteres que não sejam letras pelo caractere "\_".

**13.**Faça um programa que receba uma frase e imprima a mesma frase com as palavras repetidas.

|  |  |
| --- | --- |
| **Exemplo** | |
| **Entrada** | PROGRAMAR EH LEGAL |
| **Saída** | PROGRAMAR PROGRAMAR EH EH LEGAL LEGAL |

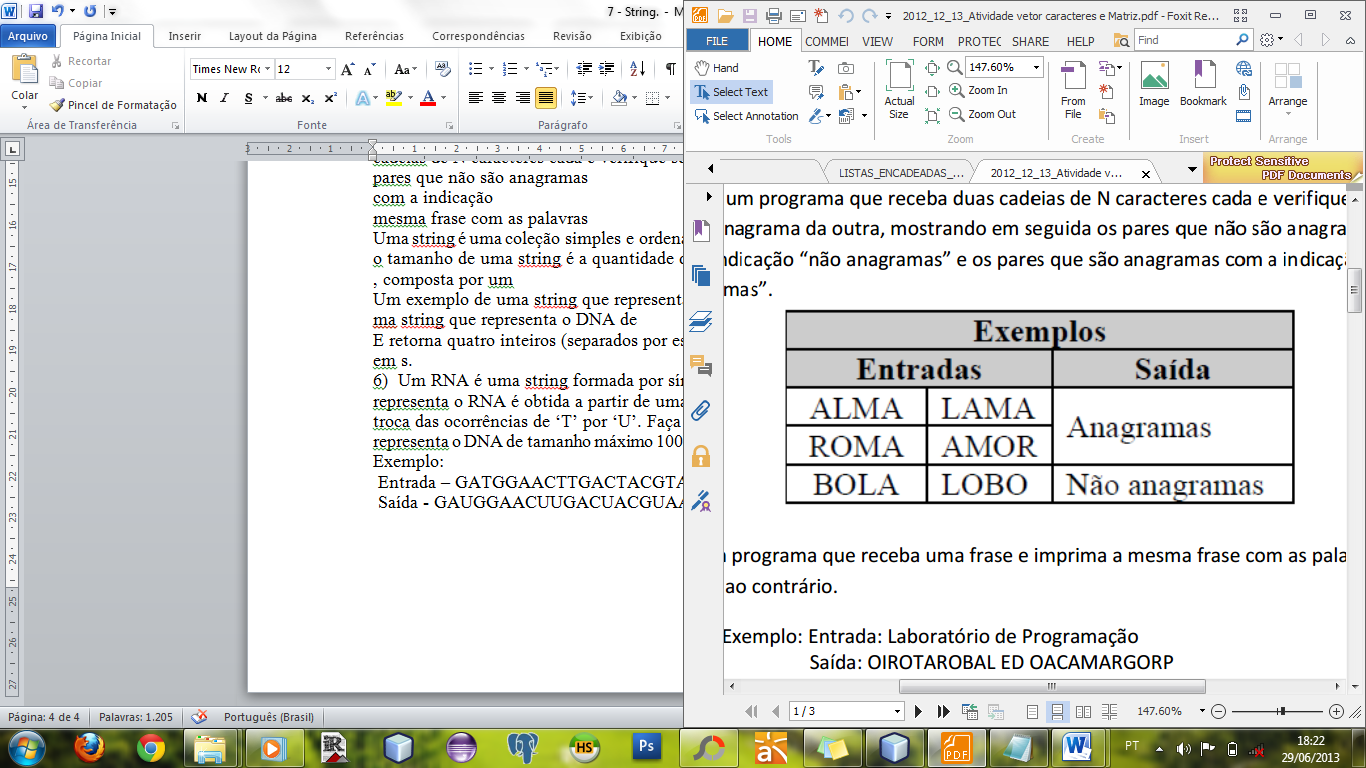
**14.** Faça um programa que receba palavras e separe suas silabas levando em consideração o seguinte critério: se após uma vogal existir uma não-vogal, deve-se separar a sílaba.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exemplos** | | | | |
| **Entradas** | LABORATORIO | PIAUI | SOUZA | QUIPROQUO |
| **Saídas** | LA-BO-RA-TO-RIO | PIAUI | SOU-ZA | QUI-PRO-QUO |

**15.** Faça um programa que receba uma frase e altere esta frase trocando as posições de duas letras consecutivas em uma palavra. Caso a palavra tenha número ímpar de letras, a última não será alterada.

|  |  |
| --- | --- |
| **Exemplo** | |
| **Entrada** | ESTA FRASE FICARA UMA BAGUNÇA SO |
| **Saída** | SEAT RFSAE IFACAR MUA ABUGÇNA OS |

**16.** Elabore um programa que receba duas strings uma é anagrama da outra, mostrando em seguida a indicação “não anagramas” e os pares que são “anagramas”.



**17.** Faça o programa do Jogo da Forca. Para cada letra que ainda não foi decifrada, o programa imprime a letra X. Caso o usuário acerte uma letra, o programa deve apresentar na palavra as ocorrências da letra e as demais permanecem com X. Se o usuário acertar a palavra, o programa apresenta “Parabéns!”, caso o usuário erre 10 vezes, o programa apresenta “Você Perdeu!” e a palavra original. (Inicie a palavra em código).

|  |
| --- |
| **Dicas** |
| Mantenha sempre o número de acertos para saber quando a palavra foi descoberta, e o número de erros para saber quando o usuário perdeu |
| Utilize uma cadeia de caracteres em separado, inicialmente composta de X, para mostrar na tela e adicione nela as letras encontradas pelo usuário |

**18.** Uma string é um palíndromo se a sequência dos caracteres da esquerda para direita é igual à seqüência da direita para a esquerda. Por exemplo,são palíndromos "ARARA", "RADAR", "AKASAKA" e "ANA". Determinar se uma string de entrada é ou não um palíndromo.

**19.** Dados como entrada uma string S e um caractere ch, ambos ascii, eliminar todas as aparições de ch em S. Por exemplo, se S = "a\_casa"e ch = ’a’, então S deverá se tornar "\_cs".

**20.** Um número é um palíndromo quando a ordem natural e reversa de dígitos revela o mesmo número. Por exemplo, 121, 234432. O número 585 é palíndromo nas bases 10 e base 2 (1001001001). Determinar a soma de todos os números com esta propriedade que sejam menores que um milhão.

**21.** Dadas duas strings de entrada S e X, determinar quantas vezes X ocorre dentro de S. Por exemplo, se S = ”zzabaxdcabasabaqwertabar” e X = ”aba” então a saída deverá ser 4.

**22.**Criar um programa capaz de ler uma frase de 4 palavras, sendo que cada palavra deve ser lida em uma variável string diferente. Depois os valores das variáveis devem ser trocados de modo que a ordem das palavras na frase fique de trás para frente, por fim as variáveis devem ser impressas (Defina somente variáveis globais).

**DIFÍCEIS**

**23.** Elabore um programa que receba duas cadeias de N caracteres cada e verifique se uma é anagrama da outra, mostrando em seguida os pares que não são anagramas com a indicação “não anagramas” e os pares que são anagramas com a indicação “anagramas”.

|  |  |  |
| --- | --- | --- |
| **Exemplos** | | |
| **Entradas** | | **Saída** |
| ALMA | LAMA | Anagramas |
| ROMA | AMOR |
| BOLA | LOBO | Não anagramas |

**24.** Faça um programa que receba uma frase e imprima a mesma frase codificada pelo método TENIS-POLAR: Para cada letra da frase que aparece nas palavras, deve-se substituí-la pela letra correspondente na outra palavra. Troca-se P por T, T por P, E por O, O por E etc. As demais permanecem como no original.

|  |  |
| --- | --- |
| **Exemplo** | |
| **Entrada** | LABORATORIO DE PROGRAMACAO I |
| **Saída** | NIBESIPESAE DO TSEGSIMICIE A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Letras que trocam** | | | | |
| T | E | N | I | S |
| P | O | L | A | R |

**25.**Um RNA é uma string formada por símbolos 'A', 'C', 'G', e 'U'. Uma string que representa o RNA é obtida a partir de uma string que representa o DNA através da troca das ocorrências de ‘T’ por ‘U’. Faça um programa que receba uma string que representa o DNA de tamanho máximo 1000 e exiba a conversão deste DNA para RNA.

Exemplo:

Entrada – GATGGAACTTGACTACGTAAATT

Saída – GAUGGAACUUGACUACGUAAAUU

**26.** Construir funções que recebam um inteiro positivo como argumento e imprimam respectivamente suas representações binária e hexadecimal.

**27.** A comparação lexicográfica de duas strings é feita caractere a caractere e as comparações ocorrem da esquerda para a direita entre caracteres de mesma posição nas strings comparadas. Se as strings possuem mesmo comprimento e todas as comparações obtém sucesso então as duas strings são consideradas lexicograficamente idênticas. Se uma é prefixo da outra então a mais curta é lexicograficamente menor (por ex.,”casa” < ”casado”). Se durante a comparação surge um caractere distinto, será lexicograficamente maior a string que possuir o caractere de maior código ascii (por ex.,”caso” > ”casa” pois ’o’ > ’a’). Considerando que strings sempre terminam num caractere nulo, comparar lexicograficamente duas strings dadas como entrada e reimprimí-las em ordem crescente.