

Projecto de Programação Imperativa

2020/2021

Introdução

Este trabalho tem como objectivo a implementação de três problemas distintos com foco na manipulação de strings. Pretende-se com este projecto a implementação de um programa que dispõe de opções para execução da componente que resolve um determinado problema.

Interface da aplicação com o utilizador

A aplicação deve ter o nome de **projc** e pode ser invocada para execução, a partir da linha de comando do *linux*, e só pode ser executado caso for indicado opções no acto da execução, segue as opções permitidas:

- -h | -help: imprime a documentação do programa, incluindo a lista das opções de execução e descrição do projecto.
- -prob <n>: executa um determinado problema, onde <n> indica o número do problema. Sabendo que apenas existe 3 problemas, então <n> só deve ser permitido de 1 à 3.
- -group: imprime a lista dos integrantes do grupo.

Em situações de ser indicada opções invalidas no acto da execução, o programa de informar a lista de opções correctas para execução do mesmo e terminar.

Desenvolvimento da aplicação

O programa deve ser desenvolvido na linguagem de programação C, norma ANSI, devendo ser organizado em vários módulos, factor que será devidamente apreciado na avaliação. A cada módulo deve estar associada uma determinada funcionalidade bem explícita. Cada módulo deve constituído por um ficheiro de extensão .c, com a definição de funções, e outro de extensão .h, com os protótipos dessas funções. O programa deve ser assim constituído por pelo menos dois módulos, sendo que o número de módulos depende da implementação efectuada.

Avaliação

O trabalho é desenvolvido em grupo, cada grupo. A nota do programa é atribuída individualmente a cada elemento do grupo numa escala de 0 a 20 valores. O seu peso na nota final do projecto é 40%, sendo os 60% reservado para a discussão.

- Um programa que não compile, tem erros, implica a reprovação de todos os integrantes do grupo;
- Um programa com avisos no acto da compilação terá no máximo 12 valores;
- Um programa cuja execução aborte ou bloqueie terá no máximo 14 valores

A nota do programa é também função de:

- Algoritmos empregues;
- Organização do código em módulos;
- Clareza e apresentação do código;

Bom trabalho.

SEGUIE OS ENUNCIADOS DOS PROBLEMAS

Problema 01

Para este problema, considere que uma string é rotacionada quando uma quantidade N de caracteres é movida do final para o início da string. Por exemplo, as rotações possíveis da palavra **linguagem** são:

inguageml, nguagemli, guagemlin, uagemling, agemlingu, gemlingua, emlinguag, mlinguage, linguagem

Observe que a própria palavra **linguagem** também é considerada como uma rotação.

Faça um programa para verificar se uma string S2 pode ser obtida pela rotação da string S1. O programa deve ler a string S1 e a string S2 e retornar 1 caso S2 possa ser obtida através da rotação de S1. Caso não seja possível, deve-se retornar 0.

Exemplo 1

Entrada

**programacao
programacao**

Saída

1

Exemplo 2

Entrada

**programacao
rogramacaop**

Saída

1

O mesmo resultado é esperado para: ogramacaopr, gramacaopro, ramacaopro, amacaoprogr, macaoprogra, acaoprogram, caoprograma, aoprogramac, oprogramaca e macaoprogra.

Exemplo 3

Entrada

papagaio
opapagai

Saída

1

O mesmo resultado é esperado para: apagaiop, pagaiopa, agaiopap, gaiopapa, aiopapag, iopapaga e papagaio.

Exemplo 4

Entrada

papagaio
opapagia

Saída

0

Problema 02

Deseja-se criar um programa capaz de identificar uma mensagem inimiga que está sendo transmitida em ondas de rádio acima de 100Mhz. O programa de computador espião DevAA já captou a transmissão e é necessário que seja construído outro software capaz de interpretar e extrair a mensagem.

O DevAA dá como saída uma cadeia como a seguinte:

```
90c87esd67uj,./';*&^120lin87uj101gu87km102a77jh150gem..&
```

Onde, da esquerda para direita:

- 90 é a frequência em Mhz.
- c é o código lido na frequência de 90Mhz.
- 87 é a frequência do próximo código.
- esd é o código lido na frequência de 87Mhz
- 67 é a frequência do próximo código.
- uj é o código lido na frequência de 67Mhz
- ,./';*&^ foi uma interferência que ocorreu quando lia-se o código da frequência de 67Mhz.
- ...

Assim, no fragmento acima, a mensagem transmitida acima de 100Mhz foi: **linguagem**. Pois, **lin** foi transmitido a 120Mhz **gu** a 101Mhz, **a** a 102Mhz e **gem** a 150Mhz.

Construa um programa capaz de recebendo uma cadeia de no máximo 250 caracteres retornar a mensagem transmitida acima de 100Mhz.

Considere que:

- a frequência estará sempre entre 1 e 200Mhz.
- toda a interferência deverá ser ignorada. Deve-se considerar interferência todo caractere diferente de uma letra ou um número.
- não existirá espaços na cadeia de entrada (produzida pelo DevAA).
- o tamanho máximo da mensagem será de 100 caracteres.

Exemplo 1

Entrada

```
90c87esd67uj,./';*&^120lin87uj101gu87km102a77jh150gem..&
```

Saída

linguagem

Exemplo 2

Entrada

```
*(12*23qualquer130i120n87j102t87ejh104er*&^_)(105n7k122e33kw140t**
```

Saída

internet

Problema 03

Geralmente um processador de textos utiliza algum algoritmo para fazer a hifenização das palavras. Neste algoritmo são consideradas posições onde a palavra pode ser dividida. Por exemplo, a palavra programação têm as seguintes possibilidades para a divisão silábica:

pro-gramação

progra-mação

programa-ção

Faça um programa que, recebendo uma cadeia de caracteres (máximo de 50 caracteres) no formato abaixo, mostre todas as divisões silábicas possíveis (uma por linha) em ordem de preferência da palavra representada.

Formato de entrada e instruções gerais

<letra0><dígito0><letra1><dígito1>...<letran><dígito n>

Onde:

- <letra_n> é uma letra minúscula do alfabeto.
- <dígito_n> pode ser omitido e é um dígito de inteiro positivo no intervalo fechado entre 1 e 9.

Caso <dígito_n> seja um número par o ponto não pode sofrer divisão silábica, caso seja ímpar poderá sofrer a divisão. Valores maiores são aqueles que indicam pontos onde há a preferência pela divisão.

Caso exista mais de um ponto com a mesma preferência, todas as opções devem ser exibidas, uma opção linha mostrando primeiro aquelas que o hífen aparece mais a esquerda.

Exemplo 1

Entrada

p2r4o5g2r4a5ma7ção

Saída

programa-ção

pro-gramação

progra-mação

Exemplo 2

Entrada

pro7gra9ma7ção

Saída

progra-mação

pro-gramação

programa-ção

Exemplo 3

Entrada

i2n3c4o8n2s7t6i9t8u7c2i4o5n6a4l

Saída

inconsti-tucional

incons-titucional

inconstitu-cional

inconstitucio-nal

in-constitucional