

 **UNISINOS** - UNIVERSIDADE DO VALE DO RIO DOS SINOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS (C6/6) – Curso: Informática

LABORATÓRIO II

Disciplina: Linguagem de Programação PASCAL
Professor responsável: *Fernando Santos Osório*
Semestre: 2000/2
Horário: 31

E-mail: osorio@exatas.unisinos.br
Web:
<http://www.inf.unisinos.br/~osorio/lab2.html>
Xerox : Pasta 54 – LAB. II (Xerox do C6/6)

TRABALHO PRÁTICO 2000/2 – GRAU A (Versão 1.0)

Faça um programa para “simular” uma calculadora programável baseada em pilhas de dados numéricos e comandos/operações (similar ao modo de funcionar das calculadoras HP). O programa deve possuir um menu com as seguintes opções:

1. Entrar dados através de um arquivo em disco: Esta opção permite ler do disco uma seqüência de dados e comandos de programação da calculadora. Estes dados serão todos lidos a partir de um arquivo texto (cujo nome é fornecido pelo usuário), sendo transferidos para a memória, onde são armazenados em uma *FILA DE ENTRADA de comandos* (máximo: 40 comandos). Antes de ler o arquivo a fila deve ser inicializada.
2. Entrar dados pelo teclado: Esta opção permite que sejam lidos a partir do teclado (informações fornecidas pelo usuário) comandos que serão adicionados ao final da *fila de comandos*, dados estes que esperam o momento de serem executados.
3. Executar comandos: Esta opção irá disparar o processo de interpretação dos comandos contidos na *fila de comandos*. Os comandos devem ser retirados da fila um-à-um, sendo executados, e prosseguindo até que não tenhamos mais comandos para retirar da *fila de comandos*. Ao terminar a execução dos comandos, reinicializar a fila.
4. Mostrar o estado da *PILHA DA CALCULADORA*: Esta opção irá mostrar o conteúdo da *pilha da calculadora* (dados que se encontram armazenados na pilha do sistema). Esta opção não deve afetar a pilha, servindo apenas para que possamos visualizar o conteúdo da *pilha da calculadora* (pilha: máximo 20 dados).
5. Limpa a pilha: Esta opção permite esvaziar a pilha da calculadora, resultando em uma pilha vazia.
6. Sair do programa.

Os dados fornecidos para a calculadora na fila de comandos podem ser de dois tipos: dados numéricos ou comandos. Cada dado é sempre fornecido em uma nova linha do arquivo, ou seja, temos um dado sempre seguido do “enter” (veja mais abaixo as dicas de como ler e interpretar os dados de entrada). A lista de comandos aceitos pela calculadora deve ser a seguinte:

- ⇒ Os dados numéricos representam apenas valores do tipo real, e são armazenados na pilha.
- ⇒ Os comandos são representados por um único caracter (não numérico).

- <valor> => Os dados numéricos ao serem lidos da fila de comandos são colocados no topo da pilha. O valor é um valor numérico real, com ou sem sinal. Exemplo: 10 (empilha o valor 10), -15.25 (empilha o valor -15.25).
- A** => **Adiciona os dois valores** contidos no topo da pilha, armazenando o resultado da soma no topo da pilha. Comando ‘A’: desempilha dois valores, soma e empilha o resultado.
- S** => **Subtrai os dois valores** contidos no topo da pilha, armazenando o resultado da subtração no topo da pilha. Comando ‘S’: desempilha dois valores, subtraindo o segundo valor do primeiro valor desempilhado (atenção a ordem do valores: primeiro menos o segundo) e empilha o resultado.
- M** => **Multiplica os dois valores** contidos no topo da pilha, armazenando o resultado da multiplicação no topo da pilha. Comando ‘M’: desempilha dois valores, multiplica e empilha o resultado.
- D** => **Divide os dois valores** contidos no topo da pilha, armazenando o resultado da divisão no topo da pilha. Comando ‘D’: desempilha dois valores, fazendo a *divisão do segundo valor pelo primeiro valor desempilhado* (atenção a ordem!) e empilha o resultado.
- R** => **Calcula o resto da divisão** entre os dois números. Comando ‘R’: desempilha dois valores, subtraindo sucessivamente o *segundo valor do primeiro valor* (atenção a ordem!) e empilhando o resultado, que representa o resto da divisão. Este comando deverá ser **implementado de maneira recursiva!**
- P** => **Calcula um percentual** em relação a um certo valor (X por cento de Y). Comando ‘P’: desempilha dois valores, calculando quanto vale a *porcentagem indicada pelo primeiro valor em relação ao valor total indicado pelo segundo valor desempilhado* (atenção a ordem!) e empilha o resultado.
- E** => **Desempilha o valor contido no topo da pilha e exhibe** este valor na tela. Exibir o valor na tela com no máximo 6 casas após a vírgula.
- T** => **Lê um valor do teclado**, colocando o valor lido no topo da pilha. Comando ‘T’: pede para o usuário digitar um valor inteiro, lê este valor, e armazena o valor lido no topo da pilha.
- I** => **Inverte a posição dos dois primeiros valores** contidos no topo da pilha, ou seja, desempilha dois valores, empilhando na ordem inversa em que foram desempilhados. Comando ‘I’: se no topo eu tinha X seguido de Y, após este comando, deverei ter Y seguido de X.
- V** => Desempilha TODOS os dados contidos na pilha, deixando a **pilha vazia**.
- X** => Realiza uma **pausa na execução** dos comandos. Comando ‘X’: espera até que o usuário pressione uma tecla para continuar a execução dos comandos contidos na fila de comandos que estavam sendo executados.
- G** => **Grava os dados** contidos na pilha em um arquivo binário de reais (Pilha.bin).
- L** => **Lê do disco os dados** contidos em um arquivo binário, colocando-os na pilha. Este comando deve poder ler os dados que foram salvos com o comando ‘G’, restaurando a pilha que havia sido salva (os valores devem estar na ordem em que se encontravam originalmente). Nome do arquivo: ‘Pilha.bin’.

Este programa deverá simular uma calculadora, baseando-se em duas estruturas de dados principais:

- FILA DE COMANDOS
- PILHA DA CALCULADORA

ATENÇÃO: Estas estruturas de dados devem ser baseadas (similares) as rotinas de manipulação de estruturas de dados que estudamos e implementamos na nossa disciplina. Fazer um programa MODULAR e SEM USAR VARIÁVEIS GLOBAIS.

*** Dicas:**

- O arquivo de comandos lido do disco (opção 1 do menu) poderá ser manipulado como um arquivo que contem apenas texto. Posteriormente os dados numéricos que forem lidos na forma textual poderão ser convertidos para um valor numérico.
 - Uma string pode ser facilmente convertida em um valor numérico inteiro ou real, usando a função “val” do Pascal.
 - Para testar se uma string possui um valor numérico ou não em seu interior, basta testar para ver se o primeiro caracter da string é um dígito entre 0 e 9, ou se for um valor negativo, o primeiro caracter será o sinal. Exemplo:
IF Texto[1] in ['0'..'9', '-'] THEN writeln ('Número') ELSE writeln ('Comandos');
- Exemplos de programas usados na calculadora descrita acima:

Soma de dois números:

1.0
2.0
A
E

Média de dois números (inteiros)

T
T
A
2.0
D
E

- O programa de simulação da calculadora deve ter um mínimo de consistência, detectando erros do tipo: pilha vazia (impossível de calcular uma operação), pilha ou fila cheia (estouro da capacidade de armazenamento de uma estrutura de dados), comando inexistente/inválido, etc. Entretanto, nós assumiremos um “sistema bem comportado”, ou seja, assumimos que o usuário não tem a intenção de fornecer dados inválidos ao sistema, sendo obrigação deste prever principalmente as situações de erro ligadas próprio sistema e erros “acidentais”.

BOM TRABALHO!

Exemplo de funcionamento dos componentes (fila e pilha) da calculadora programável:

1. O arquivo original é lido do disco, obtendo-se assim a fila de entrada.
2. Esta fila de entrada armazena os dados lidos do arquivo (1.0, 2.0, 'A' e 'E'), que começam a ser retirados um a um, sendo então analisado e executados.
3. Os dois primeiros dados são valores numéricos, e portanto devem ser empilhados na pilha (1.0 e 2.0).
4. Os demais dados da fila são dados do tipo caracter, logo estes dados são comandos a serem executados. Inicialmente o comando 'A' realiza uma adição: desempilha os 2 valores do topo da pilha, soma e empilha o resultado da soma. Depois, o comando 'E' é executado desempilhando o valor armazenado no topo da pilha e exibindo ele na tela.

