**UNIVERSIDADE ESTADUAL DO PARANÁ**

**CIÊNCIA DA COMPUTAÇÃO**

**CICLO DE INSTRUÇÃO DE UM PROCESSADOR:**

**CÓDIGO FONTE**

GUILHERME HENRIQUE DE SOUZA

APUCARANA – PR

2017

GUILHERME HENRIQUE DE SOUZA

**CICLO DE INSTRUÇÃO DE UM PROCESSADOR:**

**CÓDIGO FONTE**

Código fonte apresentado na Disciplina de arquitetura e organização de computadores para o 2º ano de Ciência da Computação.

APUCARANA – PR

2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Author: Guilherme Henrique de Souza

\* Date: 09/07/2017

\* Purpose: Trabalho arquitetura e organizacão de computadores

\* Tectonics: cobc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IDENTIFICATION DIVISION.

PROGRAM-ID. Teste.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

DATA DIVISION.

FILE SECTION.

WORKING-STORAGE SECTION.

\*> AQUI FORAM DECLARADO AS VARIAVEIS DO PROJETO,

\*> ELAS AUXILIAM NOS LOOPS E NO ARMAZENAMENTO DE DADOS.

\*> FOI ESCOLHIDO AS VARIAVEIS DO TIPO 77 POIS NÃO IREMOS

\*> PRECISAR DE SUB-ITENS.

77 WRK-OPCAO PIC 9(6) VALUE 0.

77 WRK-INICIO PIC 9.

77 WRK-OPERACAO PIC 9(6) VALUE 0 OCCURS 250 TIMES.

77 WRK-POSICAO PIC 9(6) VALUE 0 OCCURS 250 TIMES.

77 WRK-DADO PIC S9(6)V9(6) VALUE 0 OCCURS 250 TIMES.

77 WRK-LINHA PIC S9(6) VALUE 0 OCCURS 250 TIMES.

77 WRK-DADOTEMP PIC S9(12)V9(12) VALUE 0 OCCURS 250 TIMES.

77 WRK-VERIFIDIGITADO PIC S9(6) VALUE 0 OCCURS 500 TIMES.

77 WRK-INDICE PIC 9(6) VALUE 1.

77 WRK-INDICEAUX PIC 9(6) VALUE 0.

77 WRK-AUXVISUALIZARMEMORIA PIC 9(6) VALUE 0.

77 WRK-VISUALIZARMEMORIA PIC 9(6) VALUE 0.

77 WRK-PC PIC 9(6) VALUE 1.

77 WRK-MEMORIA PIC S9(12)V9(12) VALUE 0 OCCURS 500 TIMES.

77 WRK-MBR PIC S9(12)V9(12) VALUE 0.

77 WRK-MAR PIC 9(6) VALUE 0.

77 WRK-AUX PIC 9(6) VALUE 0.

77 WRK-NOP PIC 9 VALUE 0.

77 WRK-AUXLINHA PIC 9(6) VALUE 0.

77 WRK-LIMPAR PIC 9(6) VALUE 0.

77 WRK-DEPURACAO PIC 9 VALUE 0.

77 WRK-MOSTRAR PIC 9(6) VALUE 0 OCCURS 251 TIMES.

77 WRK-AUXINCIDENCIA PIC 9(6) VALUE 1.

77 WRK-INCIDENCIA PIC 9 VALUE 0.

77 WRK-CHECK PIC 9(13) VALUE 0.

77 WRK-AUXVISU PIC 9(6) VALUE 1.

77 WRK-VISU PIC 9(6) VALUE 1.

77 WRK-ENTRADADEDADOS PIC 9 OCCURS 250 TIMES.

77 WRK-DEPURACAOAUX PIC 9 VALUE 0.

77 WRK-FLAGZ PIC 9 VALUE 0.

77 WRK-FLAGN PIC 9 VALUE 0.

77 WRK-LIMPARTELA PIC 9(6) VALUE 0.

77 WRK-IND PIC 9(6) VALUE 1.

77 WRK-GUARDAR PIC 9(6) VALUE 0.

77 WRK-AUXGUARDAR PIC 9(6) VALUE 0.

77 WRK-AUXIND PIC 9(6) VALUE 1.

\*> VARIAVEIS DO CALCULO DA RAIZ

77 WRK-N PIC 9(6)V9(6).

77 WRK-RAIZANTERIOR PIC 9(6)V9(6).

77 WRK-T PIC S9(6)V9(6) VALUE 1.

77 WRK-R PIC 9(6)V9(6).

PROCEDURE DIVISION.

MAIN-PROCEDURE.

\*> DENTRO DO MAIN SÓ TEMOS TRES SUB-ROTINAS, A QUE LIMPA A

\*> TELA, A DE BOAS VINDAS E A DO INICIO DE PROGRAMA.

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

MOVE 0 TO WRK-LIMPARTELA

PERFORM BOAS-VINDAS

PERFORM INICIO UNTIL WRK-INICIO = 1

STOP RUN.

INICIO.

\*> NO INICIO DA MINHA FUNCAO "INICIO" TEMOS ALGUMAS OPCOES

\*> QUE O PROJETO PODE FAZER.

DISPLAY "===================================="

DISPLAY "= OPCOES: ="

DISPLAY "===================================="

DISPLAY "= 1 - INSERIR ="

DISPLAY "= 2 - VER INSTRUCOES ="

DISPLAY "= 3 - VER DADOS ="

DISPLAY "= 4 - EXECUTAR ="

DISPLAY "= 5 - LIMPAR ="

IF WRK-DEPURACAO EQUALS 1 THEN

DISPLAY "= 6 - ATIVAR VISUALIZACAO EM LOTE ="

END-IF

IF WRK-DEPURACAO EQUALS 0 THEN

DISPLAY "=6 - DESATIVAR VISUALIZACAO EM LOTE="

END-IF

DISPLAY "= 7 - SAIR ="

DISPLAY "===================================="

\*> SEMPRE QUE O USUARIO DIGITAR UMA OPCAO ERRADA ELE É

\*> OBRIGADO A DIGITAR NOVAMENTE.

ACCEPT WRK-OPCAO

PERFORM OPCAO UNTIL WRK-OPCAO = 000001 OR 000002 OR 000003

OR 000004 OR 000005 OR 000006 OR 000007

\*> A TABELA DE INSTRUÇÕES É MOSTRADA TODA VEZ QUE O USUARIO

\*> DIGITAR A OPÇÃO 1 (INSERIR).

IF WRK-OPCAO EQUAL 1 THEN

\*> É EXECUTADO A SUB-ROTINA LIMPAR-TELA 100 VEZES.

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

MOVE 0 TO WRK-LIMPARTELA

DISPLAY "==================================================="

DISPLAY "= INSTRUCOES: ="

DISPLAY "==================================================="

DISPLAY "= COD - OP1 - OP2 - RESULTADOS ="

DISPLAY "= 000001 - #POS - MBR <- #POS ="

DISPLAY "= 000010 - #POS #DADO - #POS <- #DADO ="

DISPLAY "= 000011 - #POS - MBR <- MBR + #POS ="

DISPLAY "= 000100 - #POS - MBR <- MBR - #POS ="

DISPLAY "= 000101 - #POS - MBR <- MBR \* #POS ="

DISPLAY "= 000110 - #POS - MBR <- MBR / #POS ="

DISPLAY "= 000111 - #LIN - JUMP to #LIN ="

DISPLAY "= 001000 - #LIN - JUMP IF Z to #lin ="

DISPLAY "= 001001 - #LIN - JUMP IF N to #lin ="

DISPLAY "= 001010 - - MBR <- raiz\_quadrada(MBR)="

DISPLAY "= 001011 - - MBR <- - MBR ="

DISPLAY "= 001111 - #POS - #POS <- MBR ="

DISPLAY "= 001100 - - NOP ="

DISPLAY "==================================================="

DISPLAY "A INSTRUCAO SERA SEPARADA PARA MELHOR DIDATICA!"

DISPLAY "DIGITE A OPERACAO: "

\*> LOGO EM SEGUIDA O USUÁRIO ENTRA COM A OPERAÇÃO QUE DEVE

\*> ATENDER AS INSTRUÇÕES DE CIMA.

ACCEPT WRK-OPERACAO(WRK-INDICE)

PERFORM LOOP UNTIL WRK-OPERACAO(WRK-INDICE) = 000001

OR 000010 OR 000011 OR 000100 OR 000101 OR 000110 OR 000111

OR 001000 OR 001001 OR 001010 OR 001011 OR 001111 OR 001100

\*> APÓS A ENTRADA DA OPERAÇÃO TEMOS O EVALUATE, QUE SERIA

\*> COMO FOSSE O SWITCH EM OUTRAS LINGUAGENS, PARA CADA

\*> OPERAÇÃO DIGITADA PELO USUÁRIO É ARMAZENADA UMA POSIÇÃO,

\*> LINHA OU DADO.

EVALUATE WRK-OPERACAO(WRK-INDICE)

WHEN "000001"

DISPLAY "DIGITE A POSICAO(251-500):"

ACCEPT WRK-POSICAO(WRK-INDICE)

\*> A POSIÇÃO NAO PODE SER MENOR QUE 250 E MAIOR QUE 500

\*> CASO CONTRÁRIO O USUÁRIO TEM QUE DIGITAR NOVAMENTE.

PERFORM POSICAO UNTIL (WRK-POSICAO(WRK-INDICE) > 250 AND

WRK-POSICAO(WRK-INDICE) <= 500)

WHEN "000010"

DISPLAY "DIGITE A POSICAO(251-500):"

ACCEPT WRK-POSICAO(WRK-INDICE)

PERFORM POSICAO UNTIL (WRK-POSICAO(WRK-INDICE) > 250 AND

WRK-POSICAO(WRK-INDICE) <= 500)

DISPLAY "DIGITE O DADO: "

ACCEPT WRK-DADOTEMP(WRK-INDICE)

WHEN "000011"

DISPLAY "DIGITE A POSICAO(251-500):"

ACCEPT WRK-POSICAO(WRK-INDICE)

PERFORM POSICAO UNTIL (WRK-POSICAO(WRK-INDICE) > 250 AND

WRK-POSICAO(WRK-INDICE) <= 500)

WHEN "000100"

DISPLAY "DIGITE A POSICAO(251-500)"

ACCEPT WRK-POSICAO(WRK-INDICE)

PERFORM POSICAO UNTIL (WRK-POSICAO(WRK-INDICE) > 250 AND

WRK-POSICAO(WRK-INDICE) <= 500)

WHEN "000101"

DISPLAY "DIGITE A POSICAO(251-500):"

ACCEPT WRK-POSICAO(WRK-INDICE)

PERFORM POSICAO UNTIL (WRK-POSICAO(WRK-INDICE) > 250 AND

WRK-POSICAO(WRK-INDICE) <= 500)

WHEN "000110"

DISPLAY "DIGITE A POSICAO(251-500):"

ACCEPT WRK-POSICAO(WRK-INDICE)

PERFORM POSICAO UNTIL (WRK-POSICAO(WRK-INDICE) > 250 AND

WRK-POSICAO(WRK-INDICE) <= 500)

\*> O MESMO ACONTECE COM A LINHA, ELE NÃO PODE SER MENOR

\*> QUE 0 E MAIOR 250.

WHEN "000111"

DISPLAY "DIGITE A LINHA(1-250):"

ACCEPT WRK-LINHA(WRK-INDICE)

PERFORM LINHA UNTIL (WRK-LINHA(WRK-INDICE) > 0 AND

WRK-LINHA(WRK-INDICE) <= 250)

WHEN "001000"

DISPLAY "DIGITE A LINHA(1-250):"

ACCEPT WRK-LINHA(WRK-INDICE)

PERFORM LINHA UNTIL (WRK-LINHA(WRK-INDICE) > 0 AND

WRK-LINHA(WRK-INDICE) <= 250)

WHEN "001001"

DISPLAY "DIGITE A LINHA(1-250):"

ACCEPT WRK-LINHA(WRK-INDICE)

PERFORM LINHA UNTIL (WRK-LINHA(WRK-INDICE) > 0 AND

WRK-LINHA(WRK-INDICE) <= 250)

WHEN "001010"

DISPLAY "RAIZ QUADRADA DE MBR"

WHEN "001011"

DISPLAY "MBR - MBR"

WHEN "001111"

DISPLAY "DIGITE A POSICAO(251-500):"

ACCEPT WRK-POSICAO(WRK-INDICE)

PERFORM POSICAO UNTIL (WRK-POSICAO(WRK-INDICE) > 250 AND

WRK-POSICAO(WRK-INDICE) <= 500)

WHEN "001100"

DISPLAY "NOP"

END-EVALUATE

COMPUTE WRK-INDICE = WRK-INDICE + 1

\*> É EXECUTADO A SUB-ROTINA LIMPAR-TELA 100 VEZES.

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

MOVE 0 TO WRK-LIMPARTELA

END-IF

\*> QUANDO A OPÇÃO FOR 2 ELE MOSTRA TODAS AS INSTRUÇÕES

\*> DIGITADAS ATÉ O MOMENTO.

IF WRK-OPCAO EQUAL 2 THEN

\*> É EXECUTADO A SUB-ROTINA LIMPAR-TELA 100 VEZES.

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

DISPLAY "==================================="

DISPLAY "= DADOS INSERIDOS ATE O MOMENTO ="

MOVE 1 TO WRK-INDICEAUX

MOVE 0 TO WRK-LIMPARTELA

PERFORM IMPRIMIR UNTIL WRK-INDICEAUX EQUAL WRK-INDICE

END-IF

\*> QUANDO A OPÇÃO FOR 3 ELE MOSTRA OS DADOS ALTERADOS NA

\*> MEMÓRIA.

IF WRK-OPCAO EQUAL 3 THEN

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

MOVE 0 TO WRK-LIMPARTELA

DISPLAY "==================================="

DISPLAY "= DADOS MODIFICADOS ATE O MOMENTO ="

PERFORM VISUALIZAR-MEMORIA UNTIL (WRK-VISUALIZARMEMORIA

EQUAL 250)

MOVE 0 TO WRK-VISUALIZARMEMORIA

PERFORM ORDENAR UNTIL WRK-IND EQUALS 251

MOVE 1 TO WRK-IND

PERFORM IMPRIMIR-MEMORIA UNTIL WRK-VISU EQUALS 252

MOVE 0 TO WRK-VISU

END-IF

\*> QUANDO A OPÇÃO FOR 4 ELE VERIFICA SE O WRK-INDICE É

\*> MAIOR QUE 1, ISSO SIGNIFICA QUE TEM ALGUMA INSTRUÇÃO

\*> E PODERÁ SER EXECUTADO SEM PROBLEMAS. O NOP É SETADO PARA

\*> 0 PARA QUE TODA VEZ QUE FOR EXECUTADO ELE POSSA RECOMECAR,

\*> CASO O USUÁRIO NAO INSIRA O NOP ELE EXECUTA ATÉ O ESTOURO.

IF WRK-OPCAO EQUAL 4 THEN

\*> É EXECUTADO A SUB-ROTINA LIMPAR-TELA 100 VEZES.

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

MOVE 0 TO WRK-LIMPARTELA

IF WRK-INDICE EQUAL 1 THEN

DISPLAY "======================================"

DISPLAY "= NAO POSSUI INSTRUCOES ="

DISPLAY "= PARA SER EXECUTADA ="

DISPLAY "======================================"

END-IF

IF WRK-INDICE > 1 THEN

DISPLAY "==================================="

DISPLAY "= EXECUTANDO ="

DISPLAY "==================================="

MOVE 1 TO WRK-PC

MOVE 0 TO WRK-NOP

MOVE 0 TO WRK-MBR

MOVE 1 TO WRK-LIMPAR

PERFORM LIMPAR-EXECUTANDO UNTIL WRK-LIMPAR EQUAL 250

PERFORM EXECUTAR UNTIL WRK-NOP EQUAL 1

END-IF

END-IF

\*> QUANDO A OPÇÃO FOR 5 ELE LIMPA TODOS OS INDÍCES PREENCHI

\*> DOS ATÉ O MOMENTO, FAZENDO COM QUE O USUÁRIO NÃO PRECISE

\*> FECHAR O PROJETO.

IF WRK-OPCAO EQUAL 5 THEN

MOVE 1 TO WRK-LIMPAR

PERFORM LIMPAR UNTIL WRK-LIMPAR EQUAL 250

MOVE 0 TO WRK-NOP

MOVE 0 TO WRK-INDICEAUX

MOVE 0 TO WRK-MBR

MOVE 0 TO WRK-MOSTRAR(251)

MOVE 0 TO WRK-MOSTRAR(250)

MOVE 1 TO WRK-INDICE

\*> É EXECUTADO A SUB-ROTINA LIMPAR-TELA 100 VEZES.

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

DISPLAY "==================================="

DISPLAY "= TODOS OS DADOS FORAM LIMPOS! ="

DISPLAY "==================================="

MOVE 0 TO WRK-LIMPARTELA

END-IF

\*> CASO O USUARIO DESEJA MUDAR O MODO A VISUALIZACAO BASTA

\*> ELE INSIRIR A OPCAO 6 NO MENU, A UNICA DIFERENCA SERIA NA

\*> HORA DE EXECUTAR E VISUALIZAR, QUE ELE AGUARDARIA UM COMA

\*> NDO DO USUARIO PARA PASSAR PARA A PRÓXIMA INSTRUÇÃO.

IF WRK-OPCAO EQUAL 6 THEN

\*> É EXECUTADO A SUB-ROTINA LIMPAR-TELA 100 VEZES.

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

MOVE 0 TO WRK-LIMPARTELA

EVALUATE WRK-DEPURACAO

WHEN 1

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

DISPLAY "======================================"

DISPLAY "=MODO DE VISUALIZACAO EM LOTE ATIVADO="

DISPLAY "======================================"

MOVE 0 TO WRK-DEPURACAO

MOVE 0 TO WRK-LIMPARTELA

WHEN 0

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

DISPLAY "========================================="

DISPLAY "=MODO DE VISUALIZACAO EM LOTE DESATIVADO="

DISPLAY "= A CADA INSTRUCAO O USUARIO DEVERA ="

DISPLAY "= APERTAR UMA TECLA PARA CONTINUAR! ="

DISPLAY "========================================="

MOVE 1 TO WRK-DEPURACAO

MOVE 0 TO WRK-LIMPARTELA

END-EVALUATE

END-IF

\*> QUANDO A OPÇÃO FOR 7 A VARÍAVEL WRK-INICIO RECEBE 1 PARA

\*> QUE O LOOP QUE ESTÁ DENTRO DO MEU MAIN PARE E SAIA DO

\*> PROJETO.

IF WRK-OPCAO EQUAL 7 THEN

\*> É EXECUTADO A SUB-ROTINA LIMPAR-TELA 100 VEZES.

PERFORM LIMPAR-TELA UNTIL WRK-LIMPARTELA EQUAL 100

MOVE 0 TO WRK-LIMPARTELA

PERFORM BOAS-VINDAS

DISPLAY "==================================="

DISPLAY "= O SIMULADOR SERA FECHADO! ="

DISPLAY "==================================="

ACCEPT WRK-INICIO

MOVE 1 TO WRK-INICIO

END-IF

.

\*> ESSA SUB-ROTINA É CHAMADA DENTRO DE OUTRA SUB-ROTINA,

\*> A SUB-ROTINA QUE RECEBE A WRK-OPCAO IGUAL A 2.

IMPRIMIR.

MOVE WRK-DADOTEMP(WRK-INDICEAUX) TO WRK-DADO(WRK-INDICEAUX)

DISPLAY "==================================="

DISPLAY "INDICE"

DISPLAY WRK-INDICEAUX

DISPLAY "OPERACAO UTILIZADA"

DISPLAY WRK-OPERACAO(WRK-INDICEAUX)

DISPLAY "POSICAO UTILIZADA"

DISPLAY WRK-POSICAO(WRK-INDICEAUX)

DISPLAY "DADO UTILIZADO"

DISPLAY WRK-DADO(WRK-INDICEAUX)

DISPLAY "LINHA UTILIZADA"

DISPLAY WRK-LINHA(WRK-INDICEAUX)

COMPUTE WRK-INDICEAUX = WRK-INDICEAUX + 1

IF WRK-DEPURACAO EQUAL 1 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

.

\*> ESSA SUB-ROTINA É CHAMADA DENTRO DE OUTRA SUB-ROTINA,

\*> A SUB-ROTINA QUE RECEBE A WRK-OPCAO IGUAL A 4.

EXECUTAR.

\*> AQUI TEMOS O PROCESSAMENTO DE TODOS OS DADOS INSERIDOS

\*> ACIMA.

EVALUATE WRK-OPERACAO(WRK-PC)

WHEN "000001"

\*> PARA MELHOR SIMULAÇÃO É EXIBIDO ALGUMAS OPERAÇÕES E

\*> DADOS CONFORME FIGURA 3.6 DO LIVRO.

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-POSICAO(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "MBR <- #POS"

DISPLAY WRK-MBR " <- #" WRK-POSICAO(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "EXECUTANDO OPERACAO"

DISPLAY "VALOR DO MBR: "

DISPLAY WRK-MBR

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

MOVE WRK-MEMORIA(WRK-AUX) TO WRK-MBR

DISPLAY "O VALOR NA MEMORIA: "

DISPLAY WRK-MEMORIA(WRK-AUX)

DISPLAY "VALOR DO MBR APOS A OPERACAO: "

DISPLAY WRK-MBR

DISPLAY "O VALOR FOI ARMAZENADO!"

PERFORM FLAGS

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "000010"

MOVE WRK-DADOTEMP(WRK-PC) TO

WRK-DADO(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-POSICAO(WRK-PC)

DISPLAY "RI <OP2>: " WRK-DADO(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "#POS <- #DADO"

DISPLAY WRK-POSICAO(WRK-PC) " <- " WRK-DADO(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "CALCULO DO ENDERECO DO SEGUNDO OPERANDO: "

DISPLAY "ENDERECO: " WRK-DADO(WRK-PC)

MOVE WRK-DADO(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O SEGUNDO OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "ARMAZENANDO "

DISPLAY WRK-DADO(WRK-PC)

DISPLAY "NA POSICAO "

DISPLAY WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

MOVE WRK-DADO(WRK-PC) TO WRK-MEMORIA(WRK-AUX)

IF WRK-DADO(WRK-PC) < 0 THEN

MOVE 0 TO WRK-FLAGZ

MOVE 1 TO WRK-FLAGN

END-IF

IF WRK-DADO(WRK-PC) EQUALS 0 THEN

MOVE 1 TO WRK-FLAGZ

MOVE 0 TO WRK-FLAGN

END-IF

IF WRK-DADO(WRK-PC) > 0 THEN

MOVE 0 TO WRK-FLAGN

MOVE 0 TO WRK-FLAGZ

END-IF

DISPLAY "CALCULANDO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-MAR

DISPLAY "ARMAZENANDO O OPERANDO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "O VALOR FOI ARMAZENADO!"

MOVE 1 TO WRK-VERIFIDIGITADO(WRK-POSICAO(WRK-PC))

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "000011"

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-POSICAO(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "MBR <- MBR + #POS"

DISPLAY "MBR <- " WRK-MBR " + #" WRK-POSICAO(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "VALOR DO MBR "

DISPLAY WRK-MBR

DISPLAY "VALOR DO CONTEUDO DA POSICAO "

DISPLAY WRK-MEMORIA(WRK-AUX)

COMPUTE WRK-CHECK = WRK-MBR + WRK-MEMORIA(WRK-AUX)

IF WRK-CHECK > 999999999999 THEN

PERFORM CHECK-ESTOURO

ELSE

COMPUTE WRK-MBR = WRK-MBR + WRK-MEMORIA(WRK-AUX)

PERFORM FLAGS

DISPLAY "VALOR DO MBR APOS A OPERACAO "

DISPLAY WRK-MBR

DISPLAY "O VALOR FOI ARMAZENADO!"

END-IF

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "000100"

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-POSICAO(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "MBR <- MBR - #POS"

DISPLAY "MBR <- " WRK-MBR " - #" WRK-POSICAO(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "VALOR DO MBR "

DISPLAY WRK-MBR

DISPLAY "VALOR DO CONTEUDO DA POSICAO "

DISPLAY WRK-MEMORIA(WRK-AUX)

COMPUTE WRK-CHECK = WRK-MBR - WRK-MEMORIA(WRK-AUX)

IF WRK-CHECK > 999999999999 THEN

PERFORM CHECK-ESTOURO

ELSE

COMPUTE WRK-MBR = WRK-MBR - WRK-MEMORIA(WRK-AUX)

PERFORM FLAGS

DISPLAY "VALOR DO MBR APOS A OPERACAO "

DISPLAY WRK-MBR

DISPLAY "O VALOR FOI ARMAZENADO!"

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

END-IF

WHEN "000101"

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-POSICAO(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "MBR <- MBR \* #POS"

DISPLAY "MBR <- " WRK-MBR " \* #" WRK-POSICAO(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "VALOR DO MBR "

DISPLAY WRK-MBR

DISPLAY "VALOR DO CONTEUDO DA POSICAO "

DISPLAY WRK-MEMORIA(WRK-AUX)

COMPUTE WRK-CHECK = WRK-MBR \* WRK-MEMORIA(WRK-AUX)

IF WRK-CHECK > 999999999999 THEN

PERFORM CHECK-ESTOURO

ELSE

COMPUTE WRK-MBR = WRK-MBR \* WRK-MEMORIA(WRK-AUX)

PERFORM FLAGS

DISPLAY "VALOR DO MBR APOS A OPERACAO "

DISPLAY WRK-MBR

DISPLAY "O VALOR FOI ARMAZENADO!"

END-IF

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "000110"

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-POSICAO(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "MBR <- MBR / #POS"

DISPLAY "MBR <- " WRK-MBR " / #" WRK-POSICAO(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "VALOR DO MBR "

DISPLAY WRK-MBR

DISPLAY "VALOR DO CONTEUDO DA POSICAO "

DISPLAY WRK-MEMORIA(WRK-AUX)

COMPUTE WRK-CHECK = WRK-MBR / WRK-MEMORIA(WRK-AUX)

IF WRK-CHECK > 999999999999 OR WRK-MEMORIA(WRK-AUX)

EQUALS 0 THEN

PERFORM CHECK-ESTOURO

ELSE

COMPUTE WRK-MBR = WRK-MBR / WRK-MEMORIA(WRK-AUX)

PERFORM FLAGS

DISPLAY "VALOR DO MBR APOS A OPERACAO "

DISPLAY WRK-MBR

DISPLAY "O VALOR FOI ARMAZENADO!"

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

END-IF

WHEN "000111"

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-LINHA(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "JUMP to #LIN"

DISPLAY "JUMP to " WRK-LINHA(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-LINHA(WRK-PC)

MOVE WRK-LINHA(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "LINHA SENDO PULADA PARA: "

DISPLAY WRK-LINHA(WRK-PC)

COMPUTE WRK-AUXLINHA = WRK-LINHA(WRK-PC) - 1

IF WRK-OPERACAO(WRK-AUXLINHA) NOT EQUAL 0

MOVE WRK-AUXLINHA TO WRK-PC

DISPLAY "OPERACAO REALIZADA COM SUCESSO!"

DISPLAY "==================================="

END-IF

IF WRK-OPERACAO(WRK-AUXLINHA) EQUAL 0

DISPLAY "===================================="

DISPLAY "= ERRO NA OPERACAO! ="

DISPLAY "=NAO E POSSIVEL REALIZAR A OPERACAO="

DISPLAY "= 000111!! ="

DISPLAY "===================================="

MOVE 1 TO WRK-NOP

END-IF

MOVE 0 TO WRK-FLAGN

MOVE 0 TO WRK-FLAGZ

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "001000"

IF WRK-FLAGZ EQUALS 0 THEN

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-LINHA(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "JUMP IF Z to #lin"

DISPLAY "JUMP IF Z to " WRK-LINHA(WRK-PC)

DISPLAY "OPERACAO NAO PODE SER EXECUTADA!"

DISPLAY "==================================="

END-IF

IF WRK-FLAGZ EQUALS 1 THEN

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-LINHA(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "JUMP IF Z to #lin"

DISPLAY "JUMP IF Z to " WRK-LINHA(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-LINHA(WRK-PC)

MOVE WRK-LINHA(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "LINHA SENDO PULADA PARA: "

DISPLAY WRK-LINHA(WRK-PC)

COMPUTE WRK-AUXLINHA = WRK-LINHA(WRK-PC) - 1

IF WRK-OPERACAO(WRK-AUXLINHA) NOT EQUAL 0

MOVE WRK-AUXLINHA TO WRK-PC

DISPLAY "OPERACAO REALIZADA COM SUCESSO!"

DISPLAY "==================================="

END-IF

IF WRK-OPERACAO(WRK-AUXLINHA) EQUAL 0

DISPLAY "===================================="

DISPLAY "= ERRO NA OPERACAO! ="

DISPLAY "=NAO E POSSIVEL REALIZAR A OPERACAO="

DISPLAY "= 001000!! ="

DISPLAY "===================================="

MOVE 1 TO WRK-NOP

END-IF

MOVE 0 TO WRK-FLAGN

MOVE 0 TO WRK-FLAGZ

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "001001"

IF WRK-FLAGN EQUALS 0 THEN

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-LINHA(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "JUMP IF N to #lin"

DISPLAY "JUMP IF N to " WRK-LINHA(WRK-PC)

DISPLAY "OPERACAO NAO PODE SER EXECUTADA!"

DISPLAY "==================================="

END-IF

IF WRK-FLAGN EQUALS 1 THEN

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-LINHA(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "JUMP IF N to #lin"

DISPLAY "JUMP IF N to " WRK-LINHA(WRK-PC)

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-LINHA(WRK-PC)

MOVE WRK-LINHA(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "LINHA SENDO PULADA PARA: "

DISPLAY WRK-LINHA(WRK-PC)

COMPUTE WRK-AUXLINHA = WRK-LINHA(WRK-PC) - 1

MOVE WRK-AUXLINHA TO WRK-PC

IF WRK-OPERACAO(WRK-AUXLINHA) NOT EQUAL 0

MOVE WRK-AUXLINHA TO WRK-PC

DISPLAY "OPERACAO REALIZADA COM SUCESSO!"

DISPLAY "==================================="

END-IF

IF WRK-OPERACAO(WRK-AUXLINHA) EQUAL 0

DISPLAY "===================================="

DISPLAY "= ERRO NA OPERACAO! ="

DISPLAY "=NAO E POSSIVEL REALIZAR A OPERACAO="

DISPLAY "= 001001!! ="

DISPLAY "===================================="

MOVE 1 TO WRK-NOP

END-IF

MOVE 0 TO WRK-FLAGN

MOVE 0 TO WRK-FLAGZ

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "001010"

IF WRK-MBR >= 0 THEN

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-MBR

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "MBR <- raiz\_quadrada(MBR)"

DISPLAY "MBR <- raiz\_quadrada("WRK-MBR")"

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-MBR

MOVE WRK-MBR TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "VALOR DO MBR "

DISPLAY WRK-MBR

MOVE WRK-MBR TO WRK-N

COMPUTE WRK-R = WRK-N / 2

PERFORM CALCULO-RAIZ UNTIL (WRK-T < 0.00001)

MOVE 1 TO WRK-T

IF WRK-R < 0 THEN

MOVE 0 TO WRK-FLAGZ

MOVE 1 TO WRK-FLAGN

END-IF

IF WRK-R EQUALS 0 THEN

MOVE 1 TO WRK-FLAGZ

MOVE 0 TO WRK-FLAGN

END-IF

IF WRK-R > 0 THEN

MOVE 0 TO WRK-FLAGN

MOVE 0 TO WRK-FLAGZ

END-IF

DISPLAY "VALOR DO MBR APOS A OPERACAO"

DISPLAY WRK-R

DISPLAY "ARMAZENANDO O OPERANDO: "

MOVE WRK-R TO WRK-MBR

DISPLAY "O VALOR FOI ARMAZENADO!"

DISPLAY "==================================="

END-IF

IF WRK-MBR < 0 THEN

DISPLAY "===================================="

DISPLAY "= ERRO NA OPERACAO! ="

DISPLAY "=NAO E POSSIVEL REALIZAR A OPERACAO="

DISPLAY "= 001010 COM NUMERO NEGATIVO! ="

DISPLAY "===================================="

MOVE 1 TO WRK-NOP

END-IF

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "001011"

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-MBR

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "MBR <- - MBR"

DISPLAY WRK-MBR " <- - " WRK-MBR

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-MBR

MOVE WRK-MBR TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO NA POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "VALOR DO MBR "

DISPLAY WRK-MBR

COMPUTE WRK-MBR = WRK-MBR \* -1

PERFORM FLAGS

DISPLAY "VALOR DO MBR APOS A OPERACAO"

DISPLAY WRK-MBR

DISPLAY "O VALOR FOI ARMAZENADO!"

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "001111"

MOVE WRK-POSICAO(WRK-PC) TO WRK-AUX

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "RI <OP1>: " WRK-POSICAO(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "#POS <- MBR"

DISPLAY WRK-POSICAO(WRK-PC) "<- " WRK-MBR

DISPLAY "CALCULO DO ENDERECO DO OPERANDO: "

DISPLAY "ENDERECO: " WRK-POSICAO(WRK-PC)

MOVE WRK-POSICAO(WRK-PC) TO WRK-MAR

DISPLAY "BUSCANDO O OPERANDO POSICAO: "

DISPLAY "MAR: " WRK-MAR

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "VALOR DO MBR "

DISPLAY WRK-MBR

MOVE WRK-MBR TO WRK-MEMORIA(WRK-AUX)

PERFORM FLAGS

DISPLAY "VALOR DO ENDERECO APOS A OPERACAO"

DISPLAY WRK-MEMORIA(WRK-AUX)

DISPLAY "O VALOR FOI ARMAZENADO!"

MOVE 1 TO WRK-VERIFIDIGITADO(WRK-POSICAO(WRK-PC))

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

WHEN "001100"

DISPLAY "CALCULO DO ENDERECO DA INSTRUCAO: "

DISPLAY "PC: " WRK-PC

DISPLAY "BUSCANDO A INSTRUCAO: "

DISPLAY "RI <OPCODE>: " WRK-OPERACAO(WRK-PC)

DISPLAY "DECODIFICANDO A INSTRUCAO: "

DISPLAY "NOP"

DISPLAY "OPERACAO DE DADOS: "

DISPLAY "ENCERRANDO OPERACOES!"

MOVE 1 TO WRK-NOP

DISPLAY "OPERACOES ENCERRADAS!"

DISPLAY "==================================="

IF WRK-DEPURACAO EQUAL 1 THEN

IF WRK-OPERACAO(WRK-PC) NOT EQUAL 0 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

END-EVALUATE

\*> QUANDO SE CHEGA NO FINAL DO EVALUATE ELE É SOMADO

\*> UM, INDEPEDENTE SE HOUVE UM PULO OU NÃO, POR ESSE

\*> MOTIVO TODA VEZ QUE TEMOS UM PULO ELE "PULA" PARA

\*> UMA LINHA A MENOS. POR EXEMPLO: SE ELE TIVESSE QUE

\*> PULAR PARA A LINHA 6, NO NOSSO PULO ELE IRIA PARA O 5.

COMPUTE WRK-PC = WRK-PC + 1

IF WRK-PC > 250 THEN

DISPLAY "O NOP NAO FOI ENCONTRADO, PARANDO DE EXECUTAR."

MOVE 1 TO WRK-NOP

END-IF

.

\*> ESSA SUB-ROTINA CHAMADA "LOOP" SERIA PARA TODA VEZ QUE O

\*> USUÁRIO ERRAR A OPERACAO.

LOOP.

DISPLAY "DIGITE NOVAMENTE A OPERACAO:"

ACCEPT WRK-OPERACAO(WRK-INDICE)

.

\*> ESSA SUB-ROTINA CHAMADA "OPCAO" SERIA PARA TODA VEZ QUE O

\*> USUÁRIO ERRAR A OPCAO.

OPCAO.

DISPLAY "DIGITE NOVAMENTE A OPCAO:"

ACCEPT WRK-OPCAO

.

\*> ESSA SUB-ROTINA CHAMADA "POSICAO" SERIA PARA TODA VEZ QUE

\*> O USUÁRIO ERRAR A POSICAO.

POSICAO.

DISPLAY "DIGITE NOVAMENTE A POSICAO(251-500):"

ACCEPT WRK-POSICAO(WRK-INDICE)

.

\*> ESSA SUB-ROTINA CHAMADA "LINHA" SERIA PARA TODA VEZ QUE O

\*> USUÁRIO ERRAR A LINHA.

LINHA.

DISPLAY "DIGITE NOVAMENTE A LINHA(1-250):"

ACCEPT WRK-LINHA(WRK-INDICE)

.

\*> ESSA FUNÇÃO CHAMADA "LIMPAR" LIMPA TODO OS DADOS ARMAZE

\*> NADOS NO PROGRAMA.

LIMPAR.

MOVE 0 TO WRK-VERIFIDIGITADO(WRK-LIMPAR)

MOVE 0 TO WRK-OPERACAO(WRK-LIMPAR)

MOVE 0 TO WRK-POSICAO(WRK-LIMPAR)

MOVE 0 TO WRK-DADO(WRK-LIMPAR)

MOVE 0 TO WRK-LINHA(WRK-LIMPAR)

MOVE 0 TO WRK-DADOTEMP(WRK-LIMPAR)

MOVE 0 TO WRK-MOSTRAR(WRK-LIMPAR)

MOVE 0 TO WRK-ENTRADADEDADOS(WRK-LIMPAR)

MOVE 0 TO WRK-MEMORIA(WRK-LIMPAR)

MOVE 0 TO WRK-MEMORIA(WRK-LIMPAR + 250)

COMPUTE WRK-LIMPAR = WRK-LIMPAR + 1

.

\*> ESSA SUB-ROTINA É UTILIZADA PARA LIMPAR OS DADOS NA MÉMO

\*> RIA PARA QUE ELA POSSA SER USADA MAIS DE UMA VEZ SEM A NE

\*> CESSIDADE DE SE FECHAR O PROJETO.

LIMPAR-EXECUTANDO.

MOVE 0 TO WRK-MEMORIA(WRK-LIMPAR + 250)

COMPUTE WRK-LIMPAR = WRK-LIMPAR + 1

.

\*> COMO NÃO FOI USADO BIBLIOTECAS PARA AUXILIAR NO DESENVOL

\*> VIMENTO DO PROJETO FOI CRIADO UMA SUB-ROTINA CHAMADA

\*> "CALCULO-RAIZ", ESSE CALCULO É FEITO ATRAVÉS DO MÉTODO DE

\*> NEWTON RAMPHSON COM PRECISÃO DE 10^-5.

CALCULO-RAIZ.

MOVE WRK-R TO WRK-RAIZANTERIOR

COMPUTE WRK-R = (((WRK-R \* WRK-R) + WRK-N) / (2 \* WRK-R))

COMPUTE WRK-T = WRK-R - WRK-RAIZANTERIOR

IF WRK-T < 0

COMPUTE WRK-T = WRK-T + (-2 \* WRK-T)

END-IF

.

\*> AQUI FOI ADOTADO UMA SOLUCAO MULTI-PLATAFORMA PARA LIMPAR

\*> A TELA, TODA VEZ QUE FOR NECESSARIO É CHAMADA ESSA FUNCAO

LIMPAR-TELA.

DISPLAY " "

COMPUTE WRK-LIMPARTELA = WRK-LIMPARTELA + 1

.

\*> ESSA SUB-ROTINA É PARA SETAR AS FLAGS DE ACORDO COM AS

\*> OPERAÇÕES EXECUTADAS.

FLAGS.

IF WRK-MBR < 0 THEN

MOVE 0 TO WRK-FLAGZ

MOVE 1 TO WRK-FLAGN

END-IF

IF WRK-MBR EQUALS 0 THEN

MOVE 1 TO WRK-FLAGZ

MOVE 0 TO WRK-FLAGN

END-IF

IF WRK-MBR > 0 THEN

MOVE 0 TO WRK-FLAGN

MOVE 0 TO WRK-FLAGZ

END-IF

.

\*> ESSA SUB-ROTINA É PARA VERIFICAR SE NÃO HOUVE ESTOURO

\*> DE DADOS OU SE HOUVE UMA DIVISÃO POR 0, COMO EM COBOL

\*> QUANDO HÁ UM ESTOURO ELE IGNORA O SINAL PODEMOS USAR

\*> SOMENTE UM PARAMETRO.

CHECK-ESTOURO.

IF WRK-CHECK > 999999999999 THEN

DISPLAY "===================================="

DISPLAY "= ESTOURO DE DADOS! ="

DISPLAY "= PARANDO A EXECUCAO! ="

DISPLAY "===================================="

MOVE 0 TO WRK-CHECK

MOVE 1 TO WRK-NOP

ELSE

DISPLAY "===================================="

DISPLAY "= OPERACAO IMPOSSIVEL! ="

DISPLAY "= PARANDO A EXECUCAO! ="

DISPLAY "===================================="

MOVE 0 TO WRK-CHECK

MOVE 1 TO WRK-NOP

.

\*> ESSA SUB-ROTINA SERIA SOMENTE PARA IMPRIMIR AS INFROMAÇÕES

\*> DO TRABALHO.

BOAS-VINDAS.

DISPLAY "==================================================="

DISPLAY "=SIMULADOR DO CICLO DE INSTRUCAO DE UM PROCESSADOR="

DISPLAY "= ARQUITETURA E ORGANIZACAO DE COMPUTADORES ="

DISPLAY "= LINGUAGEM: COBOL COMPILADOR: OPENCOBOLIDE 4.7.6 ="

DISPLAY "= ALUNO: GUILHERME HENRIQUE DE SOUZA ="

DISPLAY "= EMAIL: GUIILHERMEHSOUZA@GMAIL.COM ="

DISPLAY "==================================================="

.

\*> NESSA SUB-ROTINA É VERIFICADO SE A VERIFIDIGITADO DA POSI

\*> CAO FOI MARCADO COM 0 OU 1, CASO SEJA IGUAL A 1 ELE

\*> ELE CHAMA A SUB-ROTINA VERIFICAR-INCIDENCIA, CASO ELE

\*> SEJE "ACEITO" ELE É ARMAZENADO NO VETOR WRK-MOSTRAR PARA

\*> SER IMPRESSO DEPOIS.

VISUALIZAR-MEMORIA.

IF WRK-VERIFIDIGITADO(WRK-POSICAO(WRK-VISUALIZARMEMORIA))

NOT EQUAL 0

PERFORM VERIFICAR-INCIDENCIA UNTIL WRK-AUXINCIDENCIA EQUALS

500

IF WRK-INCIDENCIA EQUALS 0 THEN

MOVE WRK-POSICAO(WRK-VISUALIZARMEMORIA) TO

WRK-MOSTRAR(WRK-AUXVISU)

MOVE 1 TO WRK-ENTRADADEDADOS(WRK-AUXVISU)

COMPUTE WRK-AUXVISU = WRK-AUXVISU + 1

END-IF

END-IF

MOVE 1 TO WRK-AUXINCIDENCIA

MOVE 0 TO WRK-INCIDENCIA

COMPUTE WRK-VISUALIZARMEMORIA = WRK-VISUALIZARMEMORIA + 1

.

\*> NESSA SUB-ROTINA ELE SÓ VERIFICA SE NÃO EXISTE A MESMA

\*> POSIÇÃO DUAS OU MAIS VEZES NO ARRAY WRK-MOSTRAR, CASO

\*> TENHA ELE MOVE 1 PARA O WRK-INCIDENCIA.

VERIFICAR-INCIDENCIA.

IF WRK-POSICAO(WRK-VISUALIZARMEMORIA) EQUALS

WRK-MOSTRAR(WRK-AUXINCIDENCIA) THEN

MOVE 1 TO WRK-INCIDENCIA

END-IF.

COMPUTE WRK-AUXINCIDENCIA = WRK-AUXINCIDENCIA + 1

.

\*> AQUI ELE IMPRIME OS DADOS ALTERADOS NA MEMORIA,

\*> TODOS OS DADOS QUE ESTAVAM ARMAZENADOS NO ARRAY

\*> WRK-MOSTRAR E SEUS RESPECTIVOS CONTEUDOS.

IMPRIMIR-MEMORIA.

IF WRK-MOSTRAR(WRK-VISU) NOT EQUALS 0

DISPLAY "============================="

DISPLAY "DADOS ALTERADOS NA MEMORIA:"

DISPLAY WRK-MOSTRAR(WRK-VISU)

DISPLAY "VALORES ALTERADOS NA MEMORIA:"

DISPLAY WRK-MEMORIA(WRK-MOSTRAR(WRK-VISU))

DISPLAY "============================="

IF WRK-DEPURACAO EQUAL 1 THEN

ACCEPT WRK-DEPURACAOAUX

END-IF

END-IF

COMPUTE WRK-VISU = WRK-VISU + 1

.

\*> ESSA SUB-ROTINA SERVE PARA ORDENAR OS DADOS IMPRESSOS,

\*> PARA QUE SEMPRE SEJAM IMPRESSOS DO MENOR PARA O MAIOR.

ORDENAR.

IF WRK-MOSTRAR(WRK-IND) > WRK-MOSTRAR(WRK-IND + 1) THEN

MOVE WRK-MOSTRAR(WRK-IND + 1) TO WRK-GUARDAR

MOVE WRK-MOSTRAR(WRK-IND) TO WRK-MOSTRAR(WRK-IND + 1)

MOVE WRK-GUARDAR TO WRK-MOSTRAR(WRK-IND)

MOVE 0 TO WRK-IND

END-IF

COMPUTE WRK-IND = WRK-IND + 1

.

END PROGRAM Teste.