Exercício 4 - AC 2 - João Comini César de Andrade

Atividade 1

1) Respostas:
a) O que representa o rótulo main:?
É um nome simbólico para o endereço de memória onde a execução principal do programa começa.
b) Por que usar rótulos é melhor que endereços numéricos?
Rótulos tornam o código mais legível (j loop vs j 0x400028) e facilitam a manutenção, pois o montador recalcula os endereços automaticamente se o código for alterado.
c) Qual a diferença entre um símbolo e um endereço numérico?
O símbolo (ex: main) é um nome usado pelo programador no código-fonte. O endereço numérico (ex: 0x400000) é o valor real que o processador usa. O montador converte os símbolos em endereços.
2) Código

```
.globl main
.text

main:
  # Le um numero do teclado e armazena em a0
  addi t0, zero, 4
  ecall

  # Soma 2 ao valor de a0
  addi a0, a0, 2

  # Imprime o valor de a0 na tela
  addi t0, zero, 1
  ecall

  # Va novamente para o main e recomece a execucao
  j main
```

Atividade 2

```
.globl main
.text

main:
  # Le um numero do teclado e armazena em a0.
```

```
  addi t0, zero, 4
  ecall

  # Salva o numero lido (que esta em a0) em s0, conforme a dica.
  add s0, a0, zero

  # Compara o valor salvo em s0 com zero.
  # Se forem iguais, salta para o rotulo 'fim'.
  beq s0, zero, fim

  # Soma 2 ao valor original (que esta em s0) e coloca em a0 para impressao.
  addi a0, s0, 2

  # Imprime o resultado da soma.
  addi t0, zero, 1
  ecall

  # Volta ao inicio para continuar o loop.
  j main

fim:
  # Encerra a execucao do programa.
  addi t0, zero, 10
  ecall
```

Atividade 3

```
.globl main
.text

main:
  # Le um numero do teclado.
  addi t0, zero, 4
  ecall

  # Salva o numero em s0 para uso posterior.
  add s0, a0, zero

  # Verifica a condicao de parada ANTES de qualquer calculo.
  # Se o numero digitado (em s0) for zero, o programa salta
  # diretamente para 'fim', IGNORANDO a soma e a impressao.
  beq s0, zero, fim
```

```
  # O trecho abaixo so executa se o numero for diferente de zero.

  # Soma 2 ao numero original e prepara para impressao em a0.
  addi a0, s0, 2

  # Imprime o resultado.
  addi t0, zero, 1
  ecall

  # Volta ao inicio do loop.
  j main

fim:
  # Rotulo de destino para encerrar o programa de forma limpa.
  addi t0, zero, 10
  ecall
```

Atividade 4

```
.globl main
.text

main:
  # Inicializacao do contador do loop.
  # Vamos usar s1 como nosso contador 'i'.
  # i = 10.
  addi s1, zero, 10

loop:
  # --- Corpo do Loop ---
  # O codigo abaixo sera executado 10 vezes.

  # Le um numero do teclado e o armazena em a0.
  addi t0, zero, 4
  ecall

  # Soma 2 ao valor lido que esta em a0.
  addi a0, a0, 2

  # Imprime o resultado da soma.
```

```
  addi t0, zero, 1
  ecall

  # --- Controle do Loop ---

  # Decrementa o contador: i = i - 1.
  addi s1, s1, -1

  # Condicao de permanencia no loop.
  # A instrucao BGE salta se o primeiro operando for MAIOR ou IGUAL ao segundo.
  # Enquanto o contador (s1) for maior ou igual a 1, ele salta de volta para 'loop'.
  # Quando s1 se tornar 0, a condicao (0 >= 1) sera falsa e o loop termina.
  addi t1, zero, 1      # Coloca o valor 1 em um registrador temporario para a
comparacao
  bge s1, t1, loop

fim:
  # Encerra a execucao do programa.
  addi t0, zero, 10
  ecall
```

Atividade 5

```
.globl main
.text

main:
loop_inicio:
  # --- Leitura e Verificacao do Primeiro Numero ---

  # Le o primeiro numero do teclado.
  addi t0, zero, 4
  ecall

  # Verifica a condicao de saida: se o primeiro numero for 0, encerra.
  beq a0, zero, fim

  # Salva o primeiro numero em s0, pois a proxima leitura vai sobrescrever a0.
  add s0, a0, zero

  # --- Leitura e Verificacao do Segundo Numero ---

  # Le o segundo numero do teclado.
```

```
  addi t0, zero, 4
  ecall

  # Verifica a condicao de saida: se o segundo numero for 0, encerra.
  beq a0, zero, fim

  # --- Comparacao e Impressao ---

  # Compara os dois numeros (s0 e a0).
  # Se o primeiro (s0) for MAIOR ou IGUAL ao segundo (a0),
  # salta para a secao que imprime o primeiro.
  bge s0, a0, imprime_primeiro

  # Se o programa chegou aqui, o segundo numero (em a0) e o maior.
  # O valor ja esta em a0, pronto para ser impresso.
  addi t0, zero, 1
  ecall

  # Apos imprimir, salta de volta para o inicio do loop.
  j loop_inicio

imprime_primeiro:
  # Este bloco executa se o primeiro numero for o maior ou igual.
  # Move o primeiro numero (de s0) para a0 para poder imprimi-lo.
  add a0, s0, zero

  # Imprime o valor.
  addi t0, zero, 1
  ecall

  # Apos imprimir, salta de volta para o inicio do loop.
  j loop_inicio

fim:
  # Encerra a execucao do programa.
  addi t0, zero, 10
  ecall
```

Atividade 6

```
.globl main
.text
```

```
main:
  # Le um numero do teclado.
  addi t0, zero, 4
  ecall

  # Subtrai 10 do numero lido. O resultado fica em a0.
  addi a0, a0, -10

  # Verifica se o resultado (a0) e negativo.
  # A instrucao BLT (Branch if Less Than) salta se a0 for menor que zero.
  blt a0, zero, trata_negativo

# --- Bloco para Numeros Positivos (ou Zero) ---
# Este trecho so executa se o numero em a0 for >= 0.
imprime_positivo:
  # Imprime o numero normalmente.
  addi t0, zero, 1  # Chamada de sistema 1: imprimir inteiro
  ecall
  # Pula para o final para nao executar o codigo do numero negativo.
  j fim

# --- Bloco para Numeros Negativos ---
trata_negativo:
  # Salva o numero negativo (que esta em a0) em s0 para nao o perder.
  add s0, a0, zero

  # Passo 1: Imprimir o sinal de menos '-'.
  addi a0, zero, 45 # Coloca o codigo ASCII de '-' (45) em a0.
  addi t0, zero, 2  # Chamada de sistema 2: imprimir caractere.
  ecall

  # Passo 2: Converter o numero para positivo.
  # A forma mais simples de obter o valor positivo de um numero
  # negativo e subtrai-lo de zero (ex: 0 - (-5) = 5).
  sub a0, zero, s0  # a0 = 0 - s0

  # Passo 3: Imprimir o numero (agora positivo).
  addi t0, zero, 1  # Chamada de sistema 1: imprimir inteiro.
  ecall

fim:
  # Encerra a execucao do programa.
  addi t0, zero, 10
  ecall
```