Aluno: Reynan Da Silva Dias Paiva
N° Matricula: 221115751

MATA49 – PROGRAMAÇÃO DE SOFTWARE BÁSICO

1- Considerar o código “Writes Hello World to the output” do link acima.

- Comentar e identificar o que cada linha faz, ou seja, descrever o código.

```
; Simple example
; Writes Hello World to the output

        JMP start
hello: DB "Hello World!" ; Variable
       DB 0     ; String terminator

start:
        MOV C, hello    ; Point to var
        MOV D, 232      ; Point to output
        CALL print
        HLT             ; Stop execution

print:                  ; print(C:*from, D:*to)
        PUSH A
        PUSH B
        MOV B, 0
.loop:
        MOV A, [C]      ; Get char from var
        MOV [D], A      ; Write to output
        INC C
        INC D
        CMP B, [C]      ; Check if end
        JNZ .loop       ; jump if not

        POP B
        POP A
        RET
```

JPM start; transfere o controle de execução para a instruçção especificada, neste caso, o start. Quando o programa for iniciado, a primeira instrução executada é a que virá após o start.

hello: DB “Hello World!”; declara uma variavel (hello) iniciada com uma string.

DB 0; DB define um ou mais bytes em uma posição de memória especificada. Neste caso, é usada quando a função de impressão percorre a string, ele para de imprimir quando encontra o valor ‘0’.

start: ; Marca o inicio de um bloco de código.

MOV C, hello; Registrador C é usado para armazenar um endereço de memória de 16 bits. Carrega o endereço de memória da string que está na variável hello no registrador C.

MOV D, 232; Carrega o valor 232 no registrador D. Por padrão 232 é o endereço de memória do dispositivo de saída de caracteres. ASCII. Permite que o programa exiba a saída “hello world”.

CALL print; CALL é usada para chamar uma função dentro do programa. Ao chamar o CALL print, o programa começa a executar a função que foi chamada, neste caso, o print.

HLT; usada para indicar que o programa terminou de executar com sucesso. Sem essa instrução, o programa continuará executando na memória.

Print: ; o inicio da funcao print

PUSH A; empilha o valor do registrador A na pilha do processador.

PUSH B; empilha o valor do registrador B na pilha do processo

MOV B, 0; ATRIBUI O VALOR 0 PARA O REGISTRADOR B

.loop: ; Marca o início de um loop no programa

MOV A, [C]; usa o valor do processador C como um endereço de memória e lê o bite armazenado no endereço, apos isso, o valor do byte é armazenado no registrador A

MOV [D], A; escreve o conteúdo especificado pelo registrador A em um endereço de memória especificado pelo registrador D.

INC C; avançar para o próximo endereço de memória na sequência, ou seja, será apontará para a próxima posição na string.

INC D; Incrementa o valor armazenado no registrador D em 1. Isso é feito para que o registrador D agora aponte para a próxima posição no output.

CMP B, [C]; Compara o valor do registrador B com o valor apontado por C

JNZ .loop; se forem diferentes, pula de volta para a label .loop

POP B; desempilha o valor do registrador B

POP A; desempilha o valor do registrador A

RET; retorna o controle para o ponto onde a função foi chamada