Laboratório 08
Assembly: Estruturas de Controle

1. Considere o programa abaixo:

```
#include <stdio.h>

char S2[] = { 65, 108, 111, 32, 123, 103, 97,
              108, 101, 114, 97, 125,  33,  0};

int main (void) {
  char *pc = S2;
  while (*pc)
     printf ("%c", *pc++);
  printf("\n");
  return 0;
}
```

Baixe do Moodle o arquivo “prog-01.s”, que é uma tradução desse programa para Assembly.

Agora, compare o código Assembly com o código C, e veja se você consegue entender a correspondência entre eles.

Para os exercícios abaixo, escreva primeiro o código C para resolver o que se pede, faça testes, e em seguida traduza esse código C para Assembly.

2. Modifique o programa C do exercício anterior para imprimir somente as letras diferentes de 'a'. Teste sua modificação em C e depois faça a mesma modificação no código Assembly.

3. Escreva em C um programa que imprima os quadrados dos números de 1 a 10 e depois traduza-o para Assembly.

Esse programa não deve utilizar um array global de inteiros. Utilize uma variável inteira para armazenas os valores de 1 a 10. Para calcular o quadrado de um valor, você pode multiplicá-lo por si mesmo.

Note que agora você vai precisar de uma string de formatação para imprimir um valor inteiro:

```
Sf:  .string  "%d\n"
```

4. Traduza agora para Assembly o programa abaixo, lembrando o que foi visto em sala sobre o cálculo do endereço de cada posição de um array:

```
addr(A[i]) = addr(A) + i * sizeof(T); onde T é o tipo dos elementos de "A"
```

Atenção: esse programa não é igual ao visto no laboratório de array e estruturas. Nesse laboratório, percorremos os arrays com o uso de ponteiros. Você deve, agora, traduzir a operação de indexação do array usada no código C (aritmética normal, não de ponteiro).

```
#include <stdio.h>

int nums[4] = {65, -105, 111, 34};

int main (void) {
  int i;
  int s = 0;

  for (i = 0; i < 4; i++)
    s = s + nums[i];

  printf ("soma = %d\n", s);

  return 0;
}
```

Dica: Você pode usar um registrador auxiliar, por exemplo "%rcx", para fazer o cálculo do endereço do elemento do array. Lembre-se que ele deve ser um registrador de 64 bits, para que você possa fazer a soma com o endereço do início do array.

5. Voltando ao programa do exercício 3 do laboratório arrays e estruturas, com uma modificação: agora o programa imprime os elementos pares e maiores que zero do array:

```
#include <stdio.h>

int nums[] = {10, -21, -30, 45};

int main() {
  int i, *p;
  for (i = 0, p = nums; i != 4; i++, p++)
    if (((*p % 2) == 0) && (*p > 0))
      printf("%d\n", *p);
  return 0;
}
```

Traduza essa modificação. Repare que, depois de testar uma das condições, dependendo do resultado desse teste, não é necessário testar a segunda condição.

Este tipo de avaliação de expressões com operadores lógicos é chamada curto-circuito, e ocorre quando a avaliação do segundo operando da expressão somente é realizada se o resultado da avaliação do primeiro operando não é suficiente para determinar o valor da expressão.