

**17.09.2024**

**Laboratório 7**

**Theo Canuto - 2311293**

**Professor Raúl Renteria**

**INF1018 - 3WA**

2.

Código em C:

```
#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){
        if (*pc != 123 && *pc != 125) printf ("%c", *pc);
        pc++;
    }
    printf("\n");
    return 0;
}
```

Alterações feitas em Assembly:

```
    cmpl    $123, %eax
    je      Lx
    cmpl    $125, %eax
    je      Lx
    movq     $Sf, %rdi    /* primeiro parametro (ponteiro)*/
    movl     %eax, %esi   /* segundo parametro (inteiro) */
    movl     $0, %eax
    call     printf       /* chama a funcao da biblioteca */
/*****/
Lx:
    addq     $1, %r12     /* r12 += 1; */
    jmp      L1           /* goto L1; */
```

3.

Código em C:

```
#include <stdio.h>

int main(void){
    for(int i=1; i<11; i++) printf("%d ", i*i);
    printf("\n");
    return 0;
}
```

Código em Assembly:

```
.data
Sf: .string "%d "    # string de formato para imprimir valores
Sf2: .string "\n"    # string de formato para imprimir nova linha
```

```
    movl  $1, %ebx    /* ebx = 1; */

L1:
    cmpl  $11, %ebx   /* if (ebx == 11) ? */
    je    L2          /* goto L2 */

    movl  %ebx, %eax
    imull %ebx, %eax

    /*****
    /* este trecho imprime o valor de %eax (estraga %eax) */
    movq  $Sf, %rdi    /* primeiro parametro (ponteiro)*/
    movl  %eax, %esi    /* segundo parametro (inteiro) */
    movl  $0, %eax
    call  printf        /* chama a funcao da biblioteca */
    /*****/

    addl  $1, %ebx     /* ebx += 1; */
    jmp   L1
```

```
L2:
    /*****
    /* este trecho imprime o \n (estraga %eax) */
    movq  $Sf2, %rdi   /* primeiro parametro (ponteiro)*/
    movl  %ebx, %esi
    call  printf        /* chama a funcao da biblioteca */
    /*****/
```

4.

```

.data

nums: .long 65, -105, 111, 34
Sf: .string "soma = %d\n"

.text
.globl main
main:

/*****/
/* mantenha este trecho aqui e nao mexa - prologo !!! */
    pushq   %rbp
    movq    %rsp, %rbp
    subq    $16, %rsp
    movq    %rbx, -8(%rbp)
    movq    %r12, -16(%rbp)
/*****/

    movl    $0, %ebx /* ebx = 0; */
    movl    $0, %eax /* eax = 0; */

L1:
    cmpl    $4, %ebx
    je      L2
    movl    %ebx, %edx
    leaq    nums(,%rdx,4), %rcx
    movl    (%rcx), %edx
    addl    %edx, %eax
    addl    $1, %ebx
    jmp     L1

L2:
    movl    %eax, %esi
    movq    $Sf, %rdi
    movl    $0, %eax
    call    printf

```