

Nome: Lucas Lemos

Matrícula: 2110013

2ª questão

```
11 .data
12 nums: .int 10, -21, -30, 45
13 Sf: .string "%d\n" # string de formato para printf
14
15 .text
16 .globl main
17 main:
18
19 /*****
20 /* mantenha este trecho aqui e nao mexa - prologo !!! */
21 | pushq %rbp
22 | movq %rsp, %rbp
23 | subq $16, %rsp
24 | movq %rbx, -8(%rbp)
25 | movq %r12, -16(%rbp)
26 /*****/
27
28 | movl $0, %ebx /* ebx = 0; */
29 | movq $nums, %r12 /* r12 = &nums */
30 | movl $0, %ecx /* sum = 0 */
31
32 L1:
33 | cmpl $4, %ebx /* if (ebx == 4) ? */
34 | je L2 /* goto L2 */
35
36 | addl (%r12), %ecx /* sum = sum + *r12 */
37
38 | addl $1, %ebx /* ebx += 1; */
39 | addq $4, %r12 /* r12 += 4; */
40 | jmp L1 /* goto L1; */
41
42 L2:
43 | movl %ecx, %eax
44
45 /*****/
46 /* este trecho imprime o valor de %eax (estraga %eax) */
47 | movq $Sf, %rdi /* primeiro parametro (ponteiro)*/
48 | movl %eax, %esi /* segundo parametro (inteiro) */
49 | call printf /* chama a funcao da biblioteca */
50 /*****/
51
52 /*****/
53 /* mantenha este trecho aqui e nao mexa - finalizacao!!!! */
54 | movq $0, %rax /* rax = 0 (valor de retorno) */
55 | movq -8(%rbp), %rbx
56 | movq -16(%rbp), %r12
57 | leave
58 | ret
59 /*****/
```

3ª questão

```
1  .data
2  nums: .int 10, -21, -30, 45
3  Sf: .string "%d\n"
4
5  .text
6  .globl main
7  main:
8
9  /******
10 /* mantenha este trecho aqui e nao mexa - prologo !!! */
11  pushq %rbp
12  movq %rsp, %rbp
13  subq $16, %rsp
14  movq %rbx, -8(%rbp)
15  movq %r12, -16(%rbp)
16  /******
17
18  movl $0, %ebx /* ebx (i) = 0 */
19  movq $nums, %r12 /* r12 = &nums */
20
21  FOR:
22  cmpl $4, %ebx
23  je RET
24
25  movl (%r12), %ecx
26  andl $1, %ecx
27  cmpl $1, %ecx
28  je PRINT
29
30  addl $1, %ebx
31  addq $4, %r12
32  jmp FOR
33
34  PRINT:
35  movl (%r12), %eax
36
37  /******
38 /* este trecho imprime o valor de %eax (estraga %eax) */
39  movq $Sf, %rdi /* primeiro parametro (ponteiro)*/
40  movl %eax, %esi /* segundo parametro (inteiro) */
41  call printf /* chama a funcao da biblioteca */
42  /******
43  addl $1, %ebx
44  addq $4, %r12
45  jmp FOR
46
47  RET:
48
49  /******
50 /* mantenha este trecho aqui e nao mexa - finalizacao!!!! */
51  movq $0, %rax /* rax = 0 (valor de retorno) */
52  movq -8(%rbp), %rbx
53  movq -16(%rbp), %r12
54  leave
55  ret
```

4ª questão

```

1  /*
2  int nums[] = {10, -21, -30, 45};
3  int main() {
4      int i, *p;
5      for (i = 0, p = nums; i != 4; i++, p++)
6          printf("%d\n", *p);
7      return 0;
8  }
9  */
10
11 .data
12 nums: .byte 10, -21, -30, 45
13 Sf: .string "%d\n" # string de formato para printf
14
15 .text
16 .globl main
17 main:
18
19 /*****
20 /* mantenha este trecho aqui e nao mexa - prologo !!! */
21     pushq    %rbp
22     movq     %rsp, %rbp
23     subq     $16, %rsp
24     movq     %rbx, -8(%rbp)
25     movq     %r12, -16(%rbp)
26 /*****/
27
28     movl     $0, %ebx /* ebx = 0; */
29     movq     $nums, %r12 /* r12 = &nums */
30
31 L1:
32     cmpl     $4, %ebx /* if (ebx == 4) ? */
33     je       L2      /* goto L2 */
34
35     movsbl   (%r12), %eax /* eax = *r12 */
36
37 /*****/
38 /* este trecho imprime o valor de %eax (estraga %eax) */
39     movq     $$Sf, %rdi /* primeiro parametro (ponteiro)*/
40     movl     %eax, %esi /* segundo parametro (inteiro) */
41     call     printf /* chama a funcao da biblioteca */
42 /*****/
43
44     addl     $1, %ebx /* ebx += 1; */
45     addq     $1, %r12 /* r12 += 4; */
46     jmp      L1      /* goto L1; */
47
48 L2:
49 /*****/
50 /* mantenha este trecho aqui e nao mexa - finalizacao!!!! */
51     movq     $0, %rax /* rax = 0 (valor de retorno) */
52     movq     -8(%rbp), %rbx
53     movq     -16(%rbp), %r12
54     leave
55     ret
56 /*****/

```