

Nome: Lucas Demarco Cambraia Lemos
Matrícula: 2110013

1ª questão

O código está percorrendo um vetor de char, estendendo cada um de seus valores para inteiro, imprimindo, em seguida, cada um desses valores inteiros como char. Isso faz com que a saída seja o caractere correspondente na tabela ASCII.

2ª questão

.S

```
movq  $S2, %r12 /* r12 = &S2 */

L1:
    cmpb  $0, (%r12)
    je  L3

    cmpb  $123, (%r12)
    je  L2
    cmpb  $125, (%r12)
    je  L2
    movsbl (%r12), %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 */
    /******

L2:
    addq  $1, %r12 /* r12 += 1; */
    jmp  L1        /* goto L1; */

L3:
    /******
    /* este trecho imprime o \n (estraga %eax) */
    movq  $Sf2, %rdi /* primeiro parametro (ponteiro)*/
    movl  $0, %eax
    call  printf    /* chama a funcao da biblioteca */
    /******
```

.C

```
1  #include <stdio.h>
2
3  char S2[] = {65, 108, 111, 32, 123, 103, 97, 108, 101, 114, 97, 125, 33, 0};
4  int main (void) {
5      char *pc = S2;
6      while (*pc){
7          if (*pc != 123 && *pc != 125){
8              printf ("%c", *pc++);
9          }
10         else{
11             pc++;
12         }
13     }
14     printf("\n");
15     return 0;
16 }
```

3ª questão

.C

```
1  #include <stdio.h>
2
3  int main(void){
4      int valor = 1;
5
6      for(; valor <= 10; valor++){
7          printf("Quadrado de %d: %d\n", valor, (valor*valor));
8      }
9
10     return 0;
11 }
```

```
Quadrado de 1: 1
Quadrado de 2: 4
Quadrado de 3: 9
Quadrado de 4: 16
Quadrado de 5: 25
Quadrado de 6: 36
Quadrado de 7: 49
Quadrado de 8: 64
Quadrado de 9: 81
Quadrado de 10: 100
```

.S

```
1  .data
2  Sf: .string "%d\n"
3
4  .text
5  .globl main
6  main:
7  /******
8  /* mantenha este trecho aqui e nao mexa - prologo !!! */
9  |   pushq   %rbp
10 |   movq    %rsp, %rbp
11 |   subq    $16, %rsp
12 /******
13
14 |   movl    $1, %ebx
15
16 L1:
17 |   cmpl    $11, %ebx
18 |   je      L2
19 |   movl    %ebx, %ecx
20 |   imull   %ecx, %ecx
21
22 |   movl    %ecx, %eax
23 /******
24 /* este trecho imprime o valor de %eax (estraga %eax) */
25 |   movq    $Sf, %rdi /* primeiro parametro (ponteiro)*/
26 |   movl    %eax, %esi /* segundo parametro (inteiro) */
27 |   movl    $0, %eax
28 |   call    printf    /* chama a funcao da biblioteca */
29 /******
30
31 |   incl    %ebx
32 |   jmp     L1
33
34 L2:
35 /******
36 /* mantenha este trecho aqui e nao mexa - finalizacao!!!! */
37 |   movq    $0, %rax /* rax = 0 (valor de retorno) */
38 |   leave
39 |   ret
40 /******
```

```
1
4
9
16
25
36
49
64
81
100
```

4ª questão

```
1  .data
2  nums:  .int 65, -105, 111, 34
3  Sf:  .string "soma = %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 /*****/
15
16 | movq   $nums, %r12
17 | movl   $0, %ebx    /*i*/
18 | movl   $0, %eax    /*soma*/
19
20 L1:
21 | cml     $4, %ebx
22 | je     L2
23
24 | movl   %ebx, %edx  /* preserva i */
25 | imull  $4, %edx
26 | movq   %r12, %r13 /* preserva nums */
27
28 | movslq %edx, %rcx
29 | addq   %rcx, %r12
30 | addl   (%r12), %eax
31 | movq   %r13, %r12
32
33 | incl   %ebx
34 | jmp    L1
35
36 L2:
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 | movl   $0, %eax
42 | call   printf      /* chama a funcao da biblioteca */
43 /*****/
44 /*****/
45 /* mantenha este trecho aqui e nao mexa - finalizacao!!!! */
46 | movq   $0, %rax    /* rax = 0 (valor de retorno) */
47 | leave
48 | ret
49 /*****/
```