

NOME/RA : .....

### INSTRUÇÕES

1. Escreva com caneta o seu nome completo e o RA;
2. As respostas devem ser transcritas com caneta esferográfica;
3. Cada questão vale 03 (três) pontos;
4. 01 ponto será dado caso siga os padrões descritos em sala de aula;
5. As questões serão corrigidas considerando corretude, padronização e estruturação do código.

### QUESTÕES

**Questão 1.** Estruture o código abaixo de acordo com os conceitos aprendidos na disciplina [2,0 pts] e otimize-o [1,0 pt]:

```
1 #include <stdio.h>
3 /* Encontre a soma dos pares de numeros amigaveis entre 1 e 10000
4  * dois numeros a e b sao amigaveis se a soma dos divisores de "a" eh igual a "b" e
5  * a soma dos divisores de "b" eh igual a "a"
6  */
7 int soma_amigaveis()
8 {
9     int a, b, soma_div_a, soma_div_b, k, soma;
10
11     soma = 0;
12
13     for (a=1; a<=10000; a++) {
14         soma_div_a = 0;
15         for (k=1;k<a;k++) {
16             if ((a%k)==0) soma_div_a += k;
17         }
18         b = soma_div_a;
19         soma_div_b = 0;
20         for (k=1;k<b;k++) {
21             if ((b%k)==0) soma_div_b += k;
22         }
23         if (soma_div_a > a && soma_div_b == a) {
24             soma += a+b;
25         }
26     }
27     return soma;
28 }
29
30 int main()
31 {
32     printf("%d\n", soma_amigaveis());
33     return 0;
34 }
```

```
1 #include <stdio.h>
2
3 int soma_divisores (int x)
4 {
5     int i, soma = 1; /* nao preciso testar o 1 */
6     for (i=2;i<x;i++) {
```

```

8     if ((x%i) == 0) soma += i;
10  }
12  return soma;
14  }
16  /* Encontre a soma dos pares de numeros amigaveis entre 1 e 10000
18  * dois numeros a e b sao amigaveis se a soma dos divisores de "a" eh igual a "b" e
20  * a soma dos divisores de "b" eh igual a "a"
22  */
24  int soma_amigaveis()
26  {
28  int a, b, soma_div_a, soma_div_b, k, soma;
30  soma = 0;
32  /* eh facil perceber que os numeros de 1 a 10 nao podem ser amigaveis
34  * pois a soma dos divisores deles eh 1.
36  */
38  for (a=11; a<=10000; a++) {
40  soma_div_a = soma_divisores(a);
    b = soma_div_a;
    soma_div_b = soma_divisores(b);
    if (soma_div_a > a && soma_div_b == a) {
        soma += a+b;
    }
}
return soma;
}

int main()
{
printf("%d\n", soma_amigaveis());
return 0;
}

```

**Questão 2.** Uma forma de realizar a multiplicação  $m \cdot n$  é:

$$m \cdot n = \begin{cases} n, & \text{para } m==1 \\ (m/2) \cdot (2n), & \text{para } m \text{ par} \\ n + (m/2) \cdot (2n), & \text{para } m \text{ impar} \end{cases} \quad (1)$$

Implemente o algoritmo em sua versão de recursão caudal [3,0 pts].

```

1  #include <stdio.h>
3  int multiplica(int m, int n)
4  {
5  if (m==1) return n;
6  if (m%2) return n + multiplica(m/2, 2*n);
7  return multiplica(m/2, 2*n);
8  }
9
10 int multiplicaTR(int m, int n, int resultado)
11 {
12 if (m==1) return n + resultado;
13 if (m%2) return multiplicaTR(m/2, 2*n, resultado + n);
14 return multiplicaTR(m/2, 2*n, resultado);
15 }
16
17 int main ()
18 {
19 printf("%d\n", multiplica(12,14));
20 printf("%d\n", multiplicaTR(12,14,0));
21
22 return 0;
23 }

```

**Questão 3.** Corrija o código abaixo para que retorne o que é esperado [3.0 pts]:

```

1 #include <stdio.h>
3 /* copia o conteudo de s2 para s1 */
4 char * strcpy (char *s1, char *s2)
5 {
6     char *dst = s1;
7     char *src = s2;
8
9     while (*src != '\0') { /* correcao: *dst != \0 */
10        *dst = *src;
11        dst++;
12        src++;
13    }
14
15    return s1;
16 }
17
18 int main(void)
19 {
20     char s1[100] = "Ola\0jasdakjshdkjahsd";
21     char s2[100] = "Ola Mundo\0asdafasdas";
22
23     printf("s1 = %s\ns2 = %s\n", s1, s2);
24
25     printf("s1 == s2? %d\n", strcpy(s1,s2));
26
27     return 0;
28 }

```