

1) Veja o programa abaixo e o resultado.

```
#include<stdio.h>

int main()
{
    int d = 0, s = 0; // d: qtd digitos, s = soma
    int n, r;

    printf("\ndigite um número inteiro: ");
    scanf("%d", &n);
    printf("\nn%d tem ", n);

    while (n > (1) )
    {
        r = n % (2) ;
        s = s + r;
        n = n / (3) ;
        d++;
    }
    printf("%d digitos cuja soma eh %d\n", d, s);
}
```

```
~/Unisantos$ gcc somadigitos.c -o somadigitos
```

```
~/Unisantos$ ./somadigitos
```

```
digite um número inteiro: 123
```

```
123 tem 3 digitos cuja soma eh 6
```

```
~/Unisantos$ ./somadigitos
```

```
digite um número inteiro: 1050
```

```
1050 tem 4 digitos cuja soma eh 6
```

1.a) Preencha as lacunas (valor 1,5):

(1): 0

(2): 10

(3): 10

1.b) Reescreva apenas um printf para apresentar o quadrado da soma dos dígitos (valor 0,5):

```
printf("%d digitos cujo quadrado da soma eh %d\n", d, s * s);
```

1.c) Marque a alternativa correta. Para somar apenas os dígitos pares precisa modificar (valor 0,5):

EX: n = 21263. A soma apenas dos dígitos pares será $2 + 2 + 6 = 10$

- ☒ de "s = s + r;" para "if (r % 2 == 0) s = s + r;"
- ☐ de "s = s + r;" para "if (r % 2 == 1) s = s + r;"
- ☐ de "d++;" para "d = d + 1;"
- ☐ de "d++;" para "if (r % 2 == 0) d++;"
- ☐ de "printf("%d digitos cuja soma eh %d\n", d, s);" para "printf("if digito eh par, entao a soma e %d", s);"

1.d) Reescreva apenas uma instrução para apresentar a soma dos quadrados dos dígitos (valor 0,5):

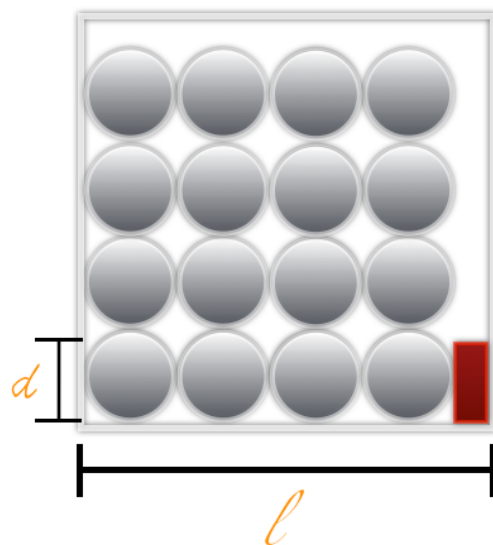
```
s = s + r*r;
```

1.d) Marque a alternativa correta. Para calcular a media dos digitos devemos (valor 0,5): **CANCELADO**

- ☐ declarar "float m;" e calcular "m = (float)s/(float)n;"
- ☐ declarar "float m;" e calcular "m = (float)n/(float)s;"
- ☐ declarar "int m;" e calcular "m = s/n;"
- ☐ declarar "float m " e calcular "m = (float)media;"

DICA: Se a variavel "x" é int, "(float)x" a transforma em um float.

2) Veja o diagrama e a figura.



O diagrama apresenta uma caixa quadrada de lado l , e o primeiro nível (base) de esferas de diâmetro d . As esferas devem ser empilhadas conforme as laranjas na figura.

2.a) Se $l = 10$ e $d = 3$, quantos níveis teremos (valor 0,5):

3 (tres) níveis

2.b) Se $l = 10$ e $d = 3$, quantas esferas teremos em cada nível (valor 0,5):

9 (nove)
4 (quatro)
1 (um)

2.c) Se $l = 10$ e $d = 3$, quantas esferas teremos na pilha (valor 0,5):

14 esferas

2.d) Complete o programa abaixo para calcular a quantidade de esferas na pilha (valor 1,0):

```
#include <stdio.h>

int main()
{
    /*****
    Dados de entrada (input) do problema:
        l: largura da caixa
        d: diametro da esfera

    Variaveis de suporte:
        n: quantidade de esferas numa lateral da base

    Dados de saida (output) do problema:
        t: numero total de esferas da pilha
    *****/
    int d, n, l, t;
    int i;

    l = 13;
    d = 3;

    n = l / d ;

    t = 0;

    for (i = 1; i <= n; i++)
    {
        t = t + i * i ;
    }

    printf("Quantidade de esferas: %d\n", t);
}
```

```
~/Unisantos$ gcc esferas.c -o esferas
~/Unisantos$ ./esferas
Quantidade de esferas: 30
```

3) Responda os testes

3.a) `m = 15; p = 3; q = 5; Qual(is) condiçõe(s) retorna(m) verdadeiro?`
Marcar todas as alternativas corretas. (valor 0,25):

☒ `if ((m % 3) == 0) && ((m % 5) == 0)) ...`
☒ `if (((m % p) == 0) || ((p % m) == 0)) ...`
☒ `if (((p % 3) == 0) && ((m % 5) == 0)) ...`
☒ `if (((m % p) == 0) || ((m % q) == 0)) ...`
☒ `if (((p % p) == 0) && ((q % q) == 0)) ...`

3.b) O que faz o fragmento do programa a seguir? Marcar todas as alternativas corretas. (valor 0,5):

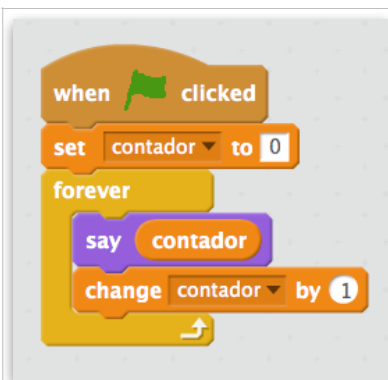
```
int a[10][10];
for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        if (i == j)
            a[i][j] = 1;
        else
            a[i][j] = 0;
    }
}
```

Cria uma matriz 10x10:

☐ de zero's, todos os elementos iguais a 0.
☐ de um's, todos os elementos iguais a 1.
☐ com metade dos elementos iguais a 1 e metade igual a 0.
☒ **que tem 100 elementos.**
☒ **com 10 elementos iguais a 1, e 90 elementos iguais a 0.**

3.c) Quais desses fragmentos de código apresentam erro de compilação ou execução. Marcar todos. (valor 0,25):

☐ `int a[3]; a[0] = 3;`
☒ `int a[3]; a[3] = 3;`
☒ `int a[2]; a[-1] = 3;`
☐ `int a[1000]; a[0] = 3;`
☒ `int a[-1]; a[-1] = 3;`
☐ `int a[3]; a[2] = 3;`



3.d) O que faz o script acima? Marcar todas as alternativas corretas. (valor 0,5):

- (☒) Conta e apresenta em sequencia os números inteiros começando em 0.
 (☐) Conta e apresenta em sequencia os números inteiros começando em 1.
 (☒) Inicializa a variavel "contador" com valor 0 .
 (☐) Apresenta apenas os 10 primeiros inteiros começando em 0.
 (☒) Possui um loop infinito.

3.e) Veja o fragmento do programa abaixo, e a tabela de descontos. Marcar a alternativa correta: (valor 1,0):

```

if (qtd > 10000)
    desconto = (0.1 * (qtd - 10000) + 0.05 * 9000)/qtd;
else if (qtd > 1000)
    desconto = (0.05 * (qtd - 1000))/qtd;
else
    desconto = 0.0;
  
```

faixa	quantidade	desconto
A	até 1000	(a)
B	entre 1001 até 10000	(b)
C	acima de 10001	(c)

- (☐) (a) = 0%, (b) = 5%, (c) = 100%
 (☐) (a) = 10%, (b) = 5%, (c) = 0%
 (☐) (a) = 100%, (b) = 5%, (c) = 0%
 (☒) (a) = 0%, (b) = 5%, (c) = 10%
 (☐) (a) = 1%, (b) = 5%, (c) = 10%

4.a) Escreva um programa em C para calcular o desvio padrão de n=5 amostras, armazenadas no array x[0], x[1], x[2], x[3], x[4], cuja média é zero.
(valor 1,0): DICA1: para raiz quadrada utilizar a função sqrt().

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2}$$

```
#include <stdio.h>

int main()
{
    float media = 0.0;
    float n = 5.0;
    float desvio;
    float x[5];
    x[0] = -2.0;
    x[1] = -1.0;
    x[2] = 0.0;
    x[3] = 1.0;
    x[4] = 2.0;

    int i;
    float ss = 0.0; // ss: sum of squares (soma dos quadrados)

    for ( i = 0; i < 5; i++ )
    {
        ss = ss + x[i] * x[i];
    }

    desvio = sqrt( ss/(n - 1.0) );

    printf("desvio padrao = %f\n", desvio);
}
```