

## CÁLCULO NUMÉRICO - Gabarito Lista No. 1

1. a)  $(5,987)_{10} \times 10^0$
- b)  $(6,30)_{10} \times 10^2$
- c)  $(5,12)_{10} \times 10^{-7}$
- d)  $(1,111)_2 \times 2^3$
- e)  $(1,011)_2 \times 2^1$
- f)  $(1,000)_2 \times 2^{-4}$

2. Truncamento      Arredondamento

- a)  $(5,98)_{10} \times 10^0$        $(5,99)_{10} \times 10^0$
- b)  $(6,30)_{10} \times 10^2$        $(6,30)_{10} \times 10^2$
- c)  $(5,12)_{10} \times 10^{-7}$        $(5,12)_{10} \times 10^{-7}$
- d)  $(1,11)_2 \times 2^3$        $(1,00)_2 \times 2^4$
- e)  $(1,01)_2 \times 2^1$        $(1,10)_2 \times 2^1$
- f)  $(1,00)_2 \times 2^{-4}$        $(1,00)_2 \times 2^{-4}$

3. a) menor número:  $(1,00000)_{10} \times 10^{-6} = 10^{-6}$ , maior número:  $(9,99999)_{10} \times 10^6 = 9999990$
- b) truncamento:  $(1,89247)_{10} \times 10^6$ , arredondamento:  $(1,89248)_{10} \times 10^6$ ,
- c) truncamento:  $a + b \rightarrow 2,57820 \cdot 10^5$ , arredondamento:  $a + b \rightarrow 2,57821 \cdot 10^5$

d)  $a + \sum_{i=1}^{10} b \rightarrow 2,57820 \cdot 10^5$

e)  $\sum_{i=1}^{10} b + a \rightarrow 2,57826 \cdot 10^5$

f) Os itens d) e e) mostram que **a adição de números em aritmética de ponto flutuante não é associativa**, mas esta operação é comutativa e tem o zero como elemento neutro.

4. b) Cálculos feitos no Octave (formato de saída short e long)

$q$	$x_1$	$x_2$ (format short)	$x_2$ (format long)
5	$1.4 \cdot 10^5$	2.5000	2.500000000000528
11	$1.4 \cdot 10^{11}$	2.5000	2.50000020685093
12	$1.4 \cdot 10^{12}$	2.5000	2.50000020685093
15	$1.4 \cdot 10^{15}$	2.4425	2.44249065417534

e)

$q$	$x_1$	$x_2$ (format long)
5	$1.4 \cdot 10^5$	2.500000000000000
11	$1.4 \cdot 10^{11}$	2.500000000000000
12	$1.4 \cdot 10^{12}$	2.500000000000000
15	$1.4 \cdot 10^{15}$	2.500000000000000

5. Código em Linguagem C.

```
#include <stdio.h>
#include <conio.h>
```

```

int fatorial (int); /* prototipo da funcao */

int main ( )
{
    int i, k;

    printf ("Escolha um numero natural:");
    scanf ("%d", &k);
    printf ("k  k!\n\n");

    for(i = 1; i <= k; i++)
    {
        printf("%d %ld\n", i, fatorial(i));
    }
    getch ( );

    return 0;
}

/* Funcao fatorial*/

int fatorial (int num) /* definicao da funcao */
{
    if (num == 0)
    return 1;
    else
    return num * fatorial(num - 1);
}

```

Na saída do programa, ocorre um overflow ao calcular 17!.

6.

```

#include <stdio.h>
#include <conio.h>

int main ( )
{
    double a, s, prec; /* com precisao dupla */
    /* troque double por float para usar precisao simples */

    a = 1; /* valor inicial de a */
    s = 1 + a;

    printf ("%A s\n\n");
}

```

```

printf (".25f % .25f\n", a, s);

do {
a = a/2;
s = 1 + a;
printf (".25f %.25f\n", a, s);
} while (s > 1);

printf ("\n");

prec = 2 * a; /* precisao da maquina */

printf ("Precisao da maquina = % .25f", prec);
getch ( );

return 0;
}

```

7. a)

```

#include <stdio.h>
#include <conio.h>

float exponencial (float, int); /* prototipo da funcao */

int main ( )
{
float x;
int n;

printf ("Digite o valor de x: ");
scanf ("%f", &x);

printf ("Digite o numero de termos da serie de Taylor: ");
scanf ("%d", &n);

printf("O valor da exp(%.8f) com %d termos e: % .30f", x, n, exponencial(x,n) );
getch ( );

return 0;
}

/* Definicao da funcao */
float exponencial (float x, int n)
{

```

```
float aux = 1.0;
float soma = 1.0;
int k;

for (k =1; k <= n; k++)
{
aux = aux * (x/k);
soma = soma + aux;
}

return soma;
}
```