

\*\*\*\*\* TROCA \*\*\*\*\*

1. Dados dois copos marcados como A e B. Copo A contem leite e Copo B limonada. Sugira uma maneira de trocar os conteudos dos copos A e B.

Pegar um copo C vazio.

Passar o conteúdo de A para C.

B -> A e C -> B.

2. Crie um algoritmo que faça as seguintes trocas: a -> b -> c -> a

(b assume o valor de a, c o valor de b, etc)

```
troca(a,b,c) //fazendo ao contrario para que os valores originais
              //sejam trocados
```

```
aux = a;
a=c;
c=b;
b=aux;
```

```
troca(a,b,c) //acabou que deu na mesma
```

```
aux1 =b;
aux2=c;
b=a;
c=aux1;
a=aux2;
```

3.Crie um algoritmo que faça as seguintes trocas: a <- b <- c <- d <- a

```
troca(a,b,c,d)
```

```
aux1=d;
d=a;
a=b;
b=c;
c=aux1;
```

4. Dadas duas variaveis inteiras a e b, troque os valores delas SEM utilizar uma variavel auxiliar.

```
troca(a,b)
```

```
if(a>0 && b<0){
    a *=-1;
    b *=-1;
    a += b;
    b = a-b;
    a -= b ;
}
else if(a<0 && b<0){
    a *=-1;
    b *=-1 ;
    a += b;
    b = a-b;
    a -= b;
    a *=-1;
    b *=-1 ;
} else{
    a += b;
    b = a-b;
    a -= b;
}
```

5. O que acontece se chamarmos a funcao troca com os seguintes argumentos? troca(i, a[i])

O programa acusa que um ponteiro está tentando ser passado no lugar de uma referencia.

\*\*\*\*\* CONTADOR \*\*\*\*\*

6. Crie um algoritmo que le uma lista de numeros e conte o numero de negativos e nao-negativos no conjunto.

```
contaNegativosePositivos(A)
    int positivos = 0, negativos = 0;
    for(int i=0;i< A.size();i++)
        if(A[i] >= 0)
            positivos ++;
        else
            negativos ++;
```

7. Assuma o algoritmo do exercicio anterior, porem com o contador inicializado com o numero de alunos n. Ao termino, o valor do contador deve conter novamente o numero de alunos que foram aprovados. Se houveram mais aprovacoes do que reprovacoes, por que esta implementacao seria melhor que a original?

```
contaAprovações(A)
    int contador=n; //alunos
    for(int i=0;i< A.size();i++)
        if (A[i] < 0)
            contador --;
    return contador;
```

Em relação ao algoritmo anterior, não existem notas menores que zero. Então ele não funciona para o exercicio atual.

Entretando, aqui, ainda que todo o vetor de notas tenho que ser percorrido, somente um fluxo de decisão existe e se o numero de aprovacoes for mais, poucas vezes o contador será decrementado. Além disso e só um contador também, com uma simples conta de subtração, que no final terá as quantidade de aprovados.

\*\*\*\*\* SOMA \*\*\*\*\*

8. Crie um algoritmo para calcular a media de n numeros.

```
media(N)
    soma =0;
    tam = N.size();
    for(i=0;i<tam;i++)
        soma = soma+N[i];
    return media= soma/tam;
```

9. Crie um algoritmo que calcule a soma dos quadrados de n numeros,

```
somaDosQuadrados(N)
    soma =0;
    for(i=0;i<N.size();i++)
        soma = soma+(N[i]*N[i]);
    return soma;
```

10. Crie um algoritmo que calcula a media harmonica de n valores.

```
harmonica(N)
    soma =0;
    tam=N.size();
    for(i=0;i<tam;i++)
        soma=soma+(1/N[i]);
    return media = tam/soma;
```

11. Crie um algoritmo que calcula a soma dos primeiros n termos ( $n \geq 0$ ) das seguintes series:

(a)  $s = 1 + 2 + 3 + \dots$

(b)  $s = 1 + 3 + 5 + \dots$

(c)  $s = 2 + 4 + 6 + \dots$

(d)  $s = 1 + 1/2 + 1/3 + \dots$

a-

```
serie(n)
    soma=0;
    for (i=1; i<=n; i++)
        soma=soma+i;
```

b-

```
serie(n)
    soma=0;
    for (i=1; i<=n; i+=2)
        soma=soma+i;
```

c-

```
serie(n)
    soma=0;
    for (i=0; i<=tam; i+=2)
        soma=soma+i;
```

d-

```
serie(n)
    soma=0;
    for (i=1; i<=tam; i++)
        soma=soma+1/i;
```

12. Gere os primeiros n termos da sequencia: 1 2 4 8 16 32 ...  
sem utilizar multiplicacao.

```
imprimeSequencia(n)
    for (i=1; i<=n; i+=i)
        print i;
```

13. Desenvolva um algoritmo que imprime n valores da sequencia: 1 -1 1 -1 1 -1 ...

```
imprimeSequencia(n)
    sinal=1;
    while (n-->0)
        imprime sinal;
        sinal*=-1;
```

14. Crie um algoritmo que calcula a soma dos primeiros n termos ( $n \geq 1$ ) da serie:  $s = 1 - 3 + 5 - 7 + 9 \dots$

```
serie(n)
    sinal=1;
    soma=0;
    for (i=1; i<=n; i+=2)
        soma=soma+(i*sinal);
        sinal*=-1;
```

\*\*\*\*\* FATORIAL \*\*\*\*\*

15. Para um dado  $n$ , crie um algoritmo que calcula  $1/n!$

```
fatorial (n)
    soma =1;
    for(i=2;i<=n;i++)
        soma*=i;
    return 1/soma;
```

16. Para um dado  $x$  e um dado  $n$ , crie um algoritmo que calcula  $x^n/n!$

```
fatComMult(x,n)
    soma =1;
    for (int i=2;i<=n;i++)
        soma*=i;
    return (x*n)/soma ;
```

17. Crie um algoritmo que determina se um numero  $n$  e' ou nao um numero fatorial.  
(um numero e' fatorial se existe algum  $x$  tal que  $x! = n$ )

```
ehFatorial(float n)
    float soma=1;
    bool flag = false;
    for( int i=1;i<=n;i++){
        soma*=i;
        if(soma==n){
            flag = true;
            break;
        }
    }
    return flag;
```

18. Crie um algoritmo que, dado um inteiro  $n$ , encontre o maior numero fatorial presente como um fator em  $n$ .

Segundo meu entendimento do enunciado: quanto maior o numero, maior o fatorial, então o fatorial de  $n$  é o maior fatorial que tem  $n$  enquanto fator.  $N$  é maior que qualquer antecessor dele.

```
quemEhMaior(n)
    maior = -1;
    soma = 1;
    for(i=1;i<=n;i++)
        soma*=i;
    if(soma>maior)
        maior = soma;
```

19. Crie um algoritmo que simule multiplicacao utilizando adicao. A entrada devem ser dois inteiros que podem ser zero, positivo ou negativo.

```
simulaMult(m,n)
    soma = 0;
    if(m==0||n==0)
        return soma;
    if((m>0 && n<0) || (m<0 && n>0))
        return soma*=-1;
    else
        return soma;

for (int i =1 ;i<=abs(m);i++)
    soma +=abs(n);
```

20. Crie um algoritmo que avalia todos os coeficientes de x para um dado valor de n.

```
coeficiente(x,n)                                fatorial(n)
    if(n<x)                                     nFat=1;
        return 0;                             for(i=1;i<=n;i++)
    nFat=fatorial(n);                           nFat*=i;
    xFat=1;                                    return nFat;
    for(i=1;i<=x;i++)
        xFat*=i
        termo=nFat/xFat*fatorial(nFat-x));
    print termo;

//como eu poderia fazer um fatorial invertido
//guardando os valores em ordem inversa?
//pegar do maior fatorial e ir passando para o menor?
```

\*\*\*\*\* COMPLEMENTO \*\*\*\*\*

1. Crie um algoritmo para calcular a soma dos primeiros n termos da série:  $fs=0!+1!+2!+3!+\dots+n!$ , ( $n \geq 0$ )

```
serie(n)
    soma = 1;
    for(i=1;i<=n;i++)
        soma*=i;
```

2. A constante de crescimento exponencial é caracterizada pela expressão:  $e=1/0!+1/1!+1/2!+1/3!+\dots$

```
e()
    i=2;
    termo=3, fatorial=1, eSoma=2;
    //pq fat de 0 é 1 e fat de 1 é 1 logo termo eSoma = 2
    while (termo>0.0000001)
        fatorial *= i++;
        termo = 1/fatorial;
        eSoma += termo;
    return eSoma;
```

3. Crie um algoritmo para calcular a função  $\cos(x)$  como definida pela expansão da série infinita:  $\cos(x)=1-x^2/2!+x^4/4!-x^6/6!+\dots$

O erro aceitável é de  $10^{-6}$ .

```
cosseno(x)
    int i = 2, sinal = -1;
    float t = ((x*x)/i)*sinal;
    float cos=1;
    while(abs(t) > 0.0000001){
        cos += t;
        sinal *= -1;
        i += 2;
        t=(t*x*x)/((i-1)*i)*sinal;
    }
    return cos;
```