

ALGORITMOS I

8ª LISTA DE EXERCÍCIOS

1 Considere a função:

```
int f1 (int k, int l) {  
    if (k > l) return 1;  
    if (k == l) return 0;  
    return (-1);  
}
```

Calcule:

- a) $f1(2, 7)$
- b) $f1(3, 3) + f1(6, 4) + f1(2, 7)$
- c) $f1(2 * f1(1, 6), 5 * f1(8, 2) + 2 * f1(1, 1))$

2 Considere a função:

```
int f2 (int a, int b, int c) {  
    if (a > b && b > c)  
        return a;  
    if (a < b && b < c)  
        return a;  
    return (a+b+c)/2;  
}
```

Calcule:

- a) $f2(10, 8, -1)$
- b) $f2(1, 4, 5)$
- c) $f2(5, 2, -2)$
- d) $f2(4 * f2(10, 5, -1), 6 * f2(1, 3, 5) - 1, 2 - f2(100, 1, 3))$

- 3** Escreva uma função que retorne o cubo de seu argumento.
- 4** Faça uma função que toque o sinal sonoro do computador.
- 5** Escreva uma função que receba um caractere como argumento e, se for uma letra minúscula, retorne-a em maiúsculo, caso contrário retorne o próprio caractere.
- 6** Escreva uma função que retorne 1 se o argumento for um número ímpar e, 0 se for par.
- 7** Escreva uma função que retorne 1 se o argumento for um caractere de 0 a 9 e, 0 caso contrário.
- 8** Faça uma função para trocar o conteúdo de duas variáveis reais. Escreva o mo-

do de chamada desta função.

- 9** Escreva uma função que retorne o absoluto de um número real (Não utilize a função `fabs()` do compilador). Faça um programa, com reprocessamento, que utilize esta função.
- 10** Indique a saída impressa pelo programa abaixo:

```
#include <stdio.h>
int x, y;
/*-----*/
void A () {
    printf ("Passou pelo A.\n");
}
/*-----*/
void B (int x) {
    printf ("Passou em B com valor %d.\n",x);
}
/*-----*/
void C (int y) {
    printf ("Valor de y: %d\n",y);
    y = 1;
    B(y);
}
/*-----*/
int D (int r, int s) {
    if (r > s) return x;
    else return y;
}
/*-----*/
int main () {
    x = 20;
    y = 30;
    B(3);
    B(4);
    A();
    x = D(D(7,2),3);
    printf ("Valor de x: %d\n",x);
    C(x);
    C(x-3);
    C(y);
    A ();
}
```

- 11** Faça uma função que retorne x^y , sendo dados x e y reais como parâmetros de entrada. Escreva um programa que calcule a potência de 15 valores lidos.
- 12** Faça uma função que retorne o fatorial de um número, inteiro e positivo, sendo este número dado como parâmetro de entrada. Escreva o modo de chamada desta função.

- 13** Utilizando a função do exercício anterior, faça um programa, com reprocessamento, que calcule e imprima o valor do coeficiente binomial, dado pela expressão a seguir:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

- 14** Escreva uma função para imprimir o cabeçalho:

```
UNESP - DEPARTAMENTO DE COMPUTACAO
XX/XX/XX                                PÁG. XXX
```

A data e o número da página devem ser fornecidos como parâmetros.

- 15** Indique a saída impressa do programa a seguir:

```
#include <stdio.h>
int x, y;
/*-----*/
void A() {
    printf ("Passou pelo A.\n");
}
/*-----*/
void B (int x) {
    printf ("Passou pelo B com x = %d.\n",x);
}
/*-----*/
void C (int y) {
    printf ("Passou pelo C com valor de y igual a %d.\n",y);
    y = 1;
    B(y);
}
/*-----*/
int D (int r, int s) {
    r += s++;
    if (r > s+10)
        return r;
    return s;
}
/*-----*/
int main () {
    x = 15;
    y = 10;
    A();
    B(3);
    x = D(x,y);
    printf("Valor da funcao D: %d - x = %d e y = %d.\n",D(x-y,x+1),x,y);
    C(x);
    C(y+5);
}
```

- 16** Dado um inteiro positivo n , faça uma função que retorne o valor da seguinte soma:

$$S = \frac{1}{N} + \frac{2}{N-1} + \frac{3}{N-2} + \dots + \frac{N-1}{2} + \frac{N}{1}$$

Escreva um programa, com reprocessamento, que utilize essa função.

- 17** Escreva uma função que calcule o valor de e^x através da série:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

em que são dados por parâmetros o valor de x e o número de termos. Faça um programa que leia 10 valores de x e o número de termos e, calcule o valor da série usando a função acima.

- 18** Indique a saída impressa do programa abaixo:

```
#include <stdio.h>
#include <stdlib.h>
int a, b, c;
/*-----*/
void P1 (int a) {
    printf ("%d\n",a);
}
/*-----*/
void P2 (int b) {
    b = 2*b + 1;
    P1(b);
}
/*-----*/
void P3 (int b) {
    b = 2*b + 1;
    P1(b);
}
/*-----*/
int P4 (int a, int b, int c) {
    if (a > b && b > c) {
        P3(a);
        return (a+2);
    }
    if (a < b || c < b) {
        P2(b);
        return (2*b+c);
    }
    return 0;
}
/*-----*/
int main () {
```

```

a = 3;
b = 4;
c = -1;
printf ("%d\n", P4(a,b,c));
P1(a);
P2(a);
P3(a);
printf ("%d\n", P4(a,b-c,c));
P2(b);
P3(b);
printf ("%d\n", P4(a+2,b,c+b-1));
P1(a);
P1(b);
P1(c);
}

```

- 19** Dados dois números inteiros positivos, escreva uma função que retorne o máximo divisor comum entre eles usando o algoritmo de Euclides. Faça um programa que utilize esta função.

Exemplo:

	1	1	1	2
25	15	9	6	3
9	6	3	0	

= mdc(24,25)

- 20** Escreva uma função que receba dois números inteiros, positivos, e determine o produto dos mesmos, utilizando o seguinte método de multiplicação:
- dividir sucessivamente o primeiro número por 2 até que obtenha 1 como quociente;
 - paralelamente, dobrar, sucessivamente, o segundo número;
 - somar os números da segunda coluna que tenham como correspondente na primeira coluna um número ímpar. O total obtido é o produto procurado.

Exemplo: cálculo de $9 * 6$

9	6	->	6
4	12		
2	24		
1	48	->	<u>+48</u>
			54

A seguir escreva um programa que leia 10 pares de números e, calcule os respectivos produtos usando a função acima.

- 21** Um número primo é qualquer inteiro positivo que é divisível apenas por si próprio e por 1. Escreva uma função que receba um inteiro positivo e, retorne se este número é primo.
- 22** Dados N números inteiros positivos, calcule a soma dos que são primos. Utilize a

função do exercício anterior.

- 23** A famosa conjectura de Goldbach diz que todo inteiro par maior que 2 é a soma de dois números primos. Testes extensivos foram feitos sem contudo ser encontrado um contra-exemplo. Escreva um programa mostrando que a afirmação é verdadeira para todo número par entre 200 e 1500. O programa deve imprimir cada número e seus correspondentes primos. Utilize a função do exercício 21.

- 24** Sabe-se que um número da forma n^3 é igual a soma de n ímpares consecutivos.

Exemplo: $1^3 = 1$, $2^3 = 3+5$, $3^3 = 7+9+11$, $4^3 = 13+15+17+19$, ...

Dado M , determine os ímpares consecutivos cuja soma é igual a n^3 para n assumindo valor de 1 a M . Utilize o conceito de função.

- 25** Um matemático italiano da idade média conseguiu modelar o ritmo de crescimento da população de coelhos através de uma sequência de números naturais que passou a ser conhecida como sequência de Fibonacci. O n -ésimo número da sequência de Fibonacci F é dado pela seguinte fórmula de recorrência:

$$\begin{cases} F_1 = 1 \\ F_2 = 1 \\ F_n = F_{n-1} + F_{n-2}, \text{ para } n \geq 3 \end{cases}$$

Escreva uma função que dado n calcule F_n . Faça um programa, com reprocessamento, que utilize esta função.

- 26** Dado um número inteiro positivo, escreva uma função que determine a sua decomposição em fatores primos. Faça um programa que utilize esta função.

- 27** Qualquer número natural de quatro algarismos pode ser dividido em duas dezenas formadas pelos seus dois primeiros e dois últimos dígitos.

Exemplo: 1278: 12 e 78.

Escreva um programa que imprima todos os milhares (4 algarismos) cuja raiz quadrada seja a soma das dezenas formadas pela divisão acima. Utilize o conceito de função.

Exemplo: A raiz de 9801 = 99 = 98 + 01.

Portanto, 9801 é um dos números a ser impresso.

- 28** Dado N e dois números naturais i e j diferentes de 0, escreva uma função que imprima em ordem crescente os N primeiros naturais que são múltiplos de i ou de j ou de ambos. Faça um programa, com reprocessamento, que utilize esta função.

Exemplo: Para $N = 6$, $i = 2$ e $j = 3$ a saída deverá ser: 0,2,3,4,6,8.

- 29** Um número natural é triangular se ele é produto de três números naturais consecutivos. Faça uma função que dado N natural, verifique se N é triangular. Escreva um programa que verifique se M números são triangulares.

Exemplo: 120 é triangular, pois $4 \cdot 5 \cdot 6 = 120$.

- 30** Um número N inteiro positivo é perfeito se for igual a soma de seus divisores positivos diferentes de N. Escreva uma função que verifique se um dado número inteiro positivo é perfeito. Escreva um programa, com reprocessamento, que utilize esta função.

Exemplo: 6 é perfeito, pois $1 + 2 + 3 = 6$.

- 31** Um número i é congruente módulo m a j se $i \text{ Mod } m = j \text{ Mod } m$. Dados N, j e m naturais não nulos, imprima os N primeiros naturais congruentes a j módulo m. Utilize o conceito de função.

- 32** Dados três números naturais, escreva uma função que verifique se eles formam os lados de um triângulo retângulo. Faça um programa, com reprocessamento, que utilize esta função.

- 33** Para evitar erros de digitação de seqüências de números de importância fundamental, como a matrícula de um aluno, o C.P.F. do Imposto de Renda, o número da conta bancária, geralmente se adiciona ao número um dígito verificador. Por exemplo, o número de matrícula 811057 é usado como 8110573, onde 3 é o dígito verificador calculado da seguinte maneira:

- a) cada algarismo do número é multiplicado por um peso começando por 2 e crescendo de 1 da direita para a esquerda;

Exemplo: $8 \cdot 7, 1 \cdot 6, 1 \cdot 5, 0 \cdot 4, 5 \cdot 3, 7 \cdot 2$

- b) somam-se as parcelas obtidas;

Exemplo: $56 + 6 + 5 + 0 + 15 + 14 = 96$

- c) obtém-se o resto da divisão desta soma por 11;

Exemplo: 96 dividido por 11 dá resto 8.

- d) subtrai-se de 11 o resto obtido;

Exemplo: $11 - 8 = 3$

- e) se o valor encontrado for 10 ou 11, o dígito verificador será 0, nos outros casos o dígito verificador é o próprio valor encontrado.

Escreva uma função que retorne o dígito verificador de um número inteiro. Faça

um programa, com reprocessamento, que utilize esta função.

- 34** Escreva uma função que, dada uma data, retorne o dia da semana. Faça um programa, com reprocessamento, que utilize esta função.

$$n = \text{Int}(365,25) * g(a,m) + \text{Int}(30,6 * f(m)) + d - 621049$$

$$ds = ((n/7) - \text{Int}(n/7)) * 7 + v + 1$$

$$g(a,m) = \begin{cases} a - 1, & \text{se } m \leq 2 \\ a, & \text{se } m > 2 \end{cases}$$

$$f(m) = \begin{cases} m + 13, & \text{se } m \leq 2 \\ m + 1, & \text{se } m > 2 \end{cases}$$

$$v = \begin{cases} 2, & \text{se } n < 36523 \\ 1, & \text{se } 36523 \leq n < 73048 \\ 0, & \text{se } n \geq 73048 \end{cases}$$

O dia da semana será dado por ds da seguinte forma: 1-domingo, 2-segunda, ...

- 35** Escreva uma função que calcule a distância entre dois pontos. Escreva também, o modo de chamada desta função.
- 36** Dado um número inteiro positivo N , escreva uma função que imprima todos os inteiros entre 1 e N que são comprimentos da hipotenusa de um triângulo retângulo com catetos inteiros. Faça um programa, com reprocessamento, que utilize esta função.

- 37** Escreva um programa que solicita dois valores (n, k) e calcule:

- a) O número de permutações P_n

$$P_n = n!$$

- b) O número de arranjos $A_{n,k}$

$$A_{n,k} = \frac{n!}{(n-k)!}$$

- b) O número de combinações $C_{n,k}$

$$C_{n,k} = \frac{n!}{k!(n-k)!}$$

O programa deve utilizar o conceito de função.

- 38** Escreva uma função que receba dois horários por parâmetro e, retorne o tempo, em segundo, entre esses dois horários. Faça um programa, com reprocessamento, que utilize essa função.
- 39** Faça uma função que determine todos os pares de números amigáveis entre um intervalo de valores, fornecidos por parâmetro. *Um par de números é amigável quando cada um deles é igual à soma dos divisores do outro.* Escreva um programa, com reprocessamento, que mostre todos os pares de números amigáveis no intervalo de 1 e 100000.

- 40** Faça um programa que sorteie um número aleatório entre 0 e 500 e pergunte ao usuário qual é o "número mágico". O programa deverá indicar se a tentativa efetuada pelo usuário é maior ou menor que o número mágico e contar o número de tentativas. Quando o usuário conseguir acertar o número o programa deverá classificar o usuário como:
- De 1 a 3 tentativas: muito sortudo
 - De 4 a 6 tentativas: sortudo
 - De 7 a 10 tentativas: normal
 - > 10 tentativas: tente novamente