



UNIVERSIDADE FEDERAL DO PARÁ

ESTRUTURAS DE DADOS I (REVISÃO) - CAPÍTULO 3 – FUNÇÕES <L3>

- 1) Fazer um programa em C para calcular o fatorial de um número, construa uma função fat que tem o corpo: void fat (int n), a função principal main chama a função fat e a função fat imprime o fatorial.
- 2) Fazer um programa em C para calcular o fatorial de um número, nesse segundo caso a função fat que tem o seguinte corpo: int fat (int n), ou seja nesse caso a função fat deve retornar um int para a função principal main, a função main deve imprimir o fatorial.
- 3) Fazer um programa em C que calcula o número de combinações de n elementos tomados k a k, na qual a ordem dos elementos é relevante. Esse número é dado pela fórmula do arranjo: $a = n! / (n-k)!$
- 4) Mostrar a pilha de Execução para a função fatorial, considerar n=5. Considerar o código abaixo:

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

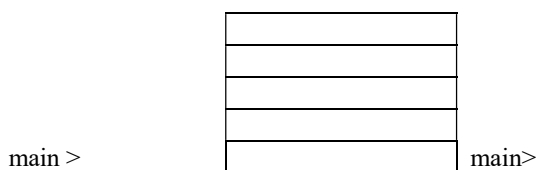
```
int main(void)
```

```
{
    int n=5;
    int r;
    r=fat(n);
    printf("O Fatorial de %d=%d\n",n,r);
    return 0;
}
```

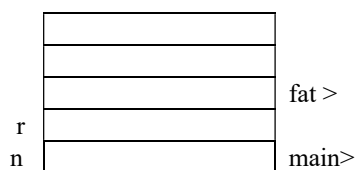
```
int fat(int n)
```

```
{
    int f=1;
    while(n!=0)
    {
        f*=n;
        n--;
    }
    return n;
}
```

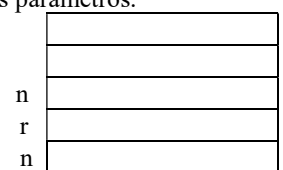
1-Início do Programa



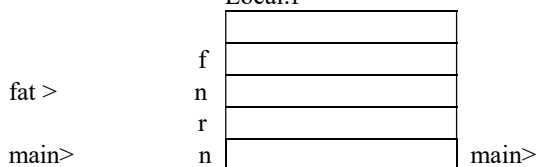
2-Declaração de n,r



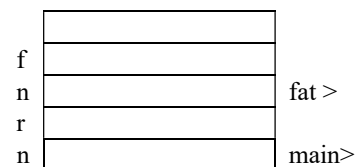
3-Chamada da Função:Copia dos parâmetros.



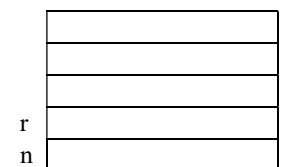
4-Declaração da variável Local:f



5-Final do laço



6-Retorno da função:desempilha



- 5) Construir uma função somaprod que calcula a soma e o produto de dois números, a função tem o seguinte estrutura: void somaprod (int a, int b, int *p, int *q), a chamada a função somaprod deve ser feita na função main, que deve conter duas variáveis locais p e q, os endereços de p e q são passados na chamada da função somaprod, e o valor da soma e do produto serão armazenados em p e em q respectivamente na função somaprod, visto que a função somaprod não retorna nenhum valor para a função main.

- 6) Implemente uma função que calcule a distância d entre dois pontos conforme a equação abaixo

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

, a função deve obedecer ao seguinte protótipo: float distancia(float x0, float y0, float x1, float y1);

- 7) Implemente uma função que efetua a troca de dois números, a função main passa dois parâmetros inteiros para a função troca que terá o seguinte corpo: void troca(int *px,int *py)

- 8) Implemente uma função que indique se um ponto(x,y) está localizado dentro de um retângulo. O retângulo é definido por seus vértices inferior esquerdo (xo,yo) e superior direito (x1,y1). A função deve ter como valor de retorno 1, se o ponto estiver dentro do retângulo, e 0 caso contrário, obedecendo o protótipo:

```
int dentro_ret(int xo,int yo,int x1,int y1,int x,int y)
```

- 9) Implemente uma função em C que calcula a média de 4 notas fornecidas pelo usuário.

- 10) Implemente uma função em C que calcule o número de Combinações de n elementos tomados k a k, na qual a ordem dos elementos é relevante.

Arranjo= $n!/(n-k)!$

PROF.MSC.LIDIO MAURO LIMA DE CAMPOS



UNIVERSIDADE FEDERAL DO PARÁ
ESTRUTURAS DE DADOS I (REVISÃO) - CAPÍTULO 3 – FUNÇÕES <L3>

11) Implemente o fatorial utilizando recursividade.

12) Implemente a função raízes, que calcula as raízes de uma equação do segundo grau do tipo $ax^2+bx+c=0$. Essa função deve obedecer o protótipo `void raizes(float a, float b, float c, float *x1, float *x2)` onde a,b,c representam os coeficientes da equação, e x1 e x2 são ponteiros para as variáveis onde devem ser guardadas as raízes da equação.

Observações

- (a) x1 deve guardar a raiz de menor valor e x2 a de maior valor.
- (b) Assuma que a equação tem sempre raízes distintas
- (c) Para o cálculo da raiz quadrada de um número, utilize a função `sqrt` definida na biblioteca padrão de funções do C. Essa função está definida no arquivo de cabeçalhos `math.h` e tem protótipo `double sqrt(double n)`.

13) Implemente um função `calc_circulo`, que calcula a área e a circunferência de um círculo de raio r. Essa função deve obedecer o protótipo

`void calc_circulo(float r, float *area, float *circunferencia)`

Fórmulas:

$$A = \pi * r^2, C = 2 * \pi * r, \pi = 3.14159265$$

14) Implemente uma função `calc_esfera`, que calcula o volume e a área da superfície de uma esfera de raio r. Essa função deve obedecer ao protótipo:

`void calc_esfera(float r, float *area, float *volume)`

Fórmulas:

$$V = 4/3 * \pi * r^3, A = 4 * \pi * r^2, \pi = 3.14159265$$

15) Implemente uma função `calc_cilindro`, que calcula o volume e a área da superfície de um cilindro de raio r e altura h. Essa função deve obedecer o protótipo.

`void calc_cilindro(float r, float h, float *area, float *volume)`

Fórmulas:

$$V = \pi * r^2 * h, A = 2 * \pi * r * h + 2 * \pi * r^2, \pi = 3.14159265$$

16) Implemente uma função `calc_cone`, que calcula o volume e a área da superfície de um cone de raio r e altura h. Essa função deve obedecer o protótipo

`void calc_cone(float r, float h, float *area, float *volume)`

Fórmulas:

$$V = 1/3 * \pi * r^2 * h, A = \pi * r * \sqrt{r^2 + h^2}, \pi = 3.14159265$$

17) Implemente a função raízes, que calcula as raízes de uma equação do segundo grau, do tipo $ax^2+bx+c=0$. Essa função deve obedecer o protótipo `int raizes(float a, float b, float c, float *x1, float *x2)`, onde a,b, e c representam coeficientes da equação, e x1 e x2 são ponteiros para as variáveis onde devem ser guardadas as raízes da equação. A função deve retornar o número de raízes que a equação possui.

Observações:

- (a) Se as raízes forem reais e distintas, x1 deve guardar a raiz de menor valor e x2 a de maior valor a função deve retornar 2.
- (b) Se as raízes forem reais e iguais $x1=x2$ a função deve retornar 1.

PROF.MSC.LIDIO MAURO LIMA DE CAMPOS



UNIVERSIDADE FEDERAL DO PARÁ
ESTRUTURAS DE DADOS I (REVISÃO) - CAPÍTULO 3 – FUNÇÕES <L3>

(c) Se não existirem raízes, $x_1=0, x_2=0$ a função deve retornar 0.

(d) Para a calculo da raiz quadrada de um numero, utilize a função sqrt definida na biblioteca padrão de funções do C. Essa função esta definida no arquivo de cabeçalhos math.h e tem protótipo double sqrt(double n).

18) Implemente a função calc_paralelepipedo, que calcula a área e o volume de um para de lados ab,c. Essa função deve obedecer o protótipo:

void calc_paralelepipedo(float a, float b, float c, float *area, float *volume)

Fórmulas:

$$A = 2(ab + ac + bc)$$

$$V = abc$$

19) Implemente a função calota, que calcula o volume e a área da superfície de uma calota esférica de raio r e altura h. Essa função deve obedecer o protótipo:

void calc_calota(float r, float h, float *area, float *volume)

Fórmulas:

$$V = \frac{1}{3} * \pi * h^2 (3*r-h), \quad A = 2 * \pi * r * h, \quad \pi = 3.14159265$$

20) Implemente a função calc_tronco, que calcula o volume e a área de uma superfície de um tronco de cone de raios a,b e altura h. Essa função deve obedecer o protótipo

void calc_tronco(float a, float b, float h, float *area, float *volume)

Fórmulas:

$$A = \pi (a+b) * \sqrt{h^2 + (b-a)^2},$$

$$V = \frac{1}{3} * \pi * h (a^2 + ab + b^2), \quad \pi = 3.14159265$$

OBS: Para a calculo da raiz quadrada de um numero, utilize a função sqrt definida na biblioteca padrão de funções do C. Essa função esta definida no arquivo de cabeçalhos math.h e tem protótipo double sqrt(double n).

Aplicações (primeira parte)

1) Implementar a Função Fatorial Usando Recursividade.

2) Codificar em C o seguinte problema data a função $V(t) = 1 - e^{-(t/T)}$, $t \geq 0$, simular a saída V(t) de acordo com a Tabela abaixo:

t	V(t)
0	
0.1	



UNIVERSIDADE FEDERAL DO PARÁ
ESTRUTURAS DE DADOS I (REVISÃO) - CAPÍTULO 3 – FUNÇÕES <L3>

0.2	
0.3	
0.4	
0.6	
0.7	
0.8	
0.9	
T	
2T	
3T	
4T	
5T	
Infinito	

Após obter os resultados encontrar os parâmetros:

$K=(V(\infty)-V(0))/A$ sendo $A=2$

$\Delta=632*(V(\infty)-V(0))+V(0)$

Plotar o Gráfico de $V(t)$