

Instituto Federal de Educação, Ciência e Tecnologia do Maranhão - IFMA

Departamento de Eletroeletrônica – DEE

Curso de Engenharia Elétrica Industrial

Introdução à Programação

Prof. Francisco Santos

Aluno: _____

Lista de Exercícios

1. Implemente, sem o uso de funções, um programa que resolva o seguinte problema:

$$x^n = x.x.....x \quad (n \text{ vezes})$$

```
1 // Programa que calcula potência
2
3 #include <stdio.h> //inclusão da biblioteca
4
5 main()
6 {
7     int base, expoente, potencia, i; // declaração da variáveis do programa
8
9     printf("Entre com a base\n"); //solicitação da base da potência
10    scanf("%d",&base);
11
12    printf("Entre com o expoente\n"); //solicitação do expoente da potência
13    scanf("%d",&expoente);
14
15
16    //cálculo da potência
17    potencia =1;
18    for (i = 1; i <= expoente; i++)
19        potencia = potencia*base;
20
21    //impressão da potencia
22    printf("O resultado: %d\n",potencia);
23
24 }
```

2. O quadrado de um número positivo pode ser encontrado pelo seguinte método:

O quadrado de um número N positivo é igual à soma dos n primeiros números ímpares.

Esse algoritmo pode ser traduzido da seguinte forma:

$$n^2 = \sum_{i=0}^{n-1} (2i + 1)$$

Faça um programa que solicite do usuário um numero N e forneça o seu quadrado composto à partir da sequência de números ímpares.

```

1 // Programa que calcula o quadrado de um número
2
3 #include <stdio.h> //inclusão da biblioteca
4
5 main()
6 {
7     int N, i,quadrado; // declaração da variáveis do programa
8
9     printf("Entre com o numero\n"); //solicitação do número
10    scanf("%d",&N);
11
12    //cálculo do quadrado do número
13    quadrado=0;
14    for (i = 0; i <= N-1; i++)
15        quadrado = quadrado +(2*i+1); //adiciona somente os numeros ímpares
16
17
18    //impressão do resultado
19    printf("O Quadrado desse numero: %d\n",quadrado);
20
21 }

```

3. Implemente, sem uso de funções e biblioteca, um programa que faça a função Exp, definida pela seguinte fórmula:

$$Exp(x, n, t) = \frac{x}{1+t} + \frac{x}{(1+t)^2} + \frac{x}{(1+t)^3} + \dots + \frac{x}{(1+t)^n}$$

sendo (x, n, t) fornecidos pelo usuário.

```

1 // Programa o Exp
2 #include <stdio.h> //inclusão da biblioteca
3
4 main ()
5 {
6     // declaração da variáveis do programa
7     float x, t, den, resultado;
8     int n, i, j;
9
10    //solicitação dos dados
11    printf ("Entre com o valor de x\n");
12    scanf ("%f", &x);
13
14    printf ("Entre com o valor de n\n");
15    scanf ("%d", &n);
16
17    printf ("Entre com o valor de t\n");
18    scanf ("%f", &t);
19
20    resultado = 0;
21    for (i = 1; i <= n; i++)
22    {
23        den = 1; // Cálculo da potência do denominador
24        for (j = 1; j <= i; j++)
25            den = den * (1 + t);
26
27        resultado = resultado + x / den; //somatório de cada fração
28    }
29
30    printf("O resultado: %f", resultado);
31
32 }

```

4. O fatorial de um numero N é definido pela seguinte equação:

$$n! = n.(n-1).(n-2).(n-3)...2.1$$

Faça um programa que solicite do usuário um numero e calcule o fatorial.

```

1 // Programa que calcula o fatorial de um número N
2 #include <stdio.h> //inclusão da biblioteca
3
4 main ()
5 {
6     // declaração da variáveis do programa
7     int i, fatorial,N;
8
9     //solicitação dos dados
10    printf ("Entre com o valor de N\n");
11    scanf ("%d", &N);
12
13    fatorial = 1;
14    for (i = N; i >=1; i--)
15        fatorial = fatorial*i;
16
17
18    printf("O fatorial: %d", fatorial);
19
20 }

```

5. A velocidade de um sistema MUV é definida como:

$$v(t) = v_0 + at$$

Faça um programa que solicite do usuário a velocidade inicial v_0 e o tempo de estudo t e forneça como saída o conjunto de dados $(v(t), t)$ desde o instante inicial até o tempo t .

```

1 // Programa que calcula v(t) =v0+at
2 #include <stdio.h> //inclusão da biblioteca
3
4 main ()
5 {
6     // declaração da variáveis do programa
7     float v_t, v_0, a, t;
8
9     a = 2; //aceleração de 2 m/s^2
10
11    //solicitação dos dados
12    printf ("Entre com o valor da velocidade inicial\n");
13    scanf ("%f", &v_0);
14
15    printf ("Entre com o valor do tempo de estudo\n");
16    scanf ("%f", &t);
17
18
19
20    //impressão dos valores (v(t),t)
21    printf("valores (v(t),t):\n");
22    for(int i=0;i<=t;i++)
23    {
24        v_t = v_0 + a*i;
25        printf("(%.1f,%.1f)\n", v_t,i);
26    }
27 }

```

6. O valor do campo magnético \vec{B} em um condutor percorrido por uma corrente elétrica i depende da distancia r do condutor. Sua equação é definida por:

$$\vec{B} = \frac{\mu_0 i}{2\pi r}$$

onde $\mu_0 = 4\pi 10^{-7}$ é a permeabilidade magnética no vácuo. Faça um programa que solicite do usuário o valor da corrente elétrica e a distância, e forneça o valor do campo magnético quando a distancia tende a um valor r .

```
1 // Programa que calcula B=μ0i/2πr
2 #include <stdio.h> //inclusão da biblioteca
3 #include<math.h>
4 #define Pi 3.14 //definição de uma constante
5 main ()
6 {
7 // declaração da variáveis do programa
8 float u_0, B;
9 int d,r,c_i;
10
11 u_0 = 4*Pi*pow(10,-7); //valor da permeabilidade magnética
12
13 //solicitação dos dados
14 printf ("Entre com o valor da corrente (A)\n");
15 scanf ("%d", &c_i);
16
17 printf ("Entre com o valor da distância d(cm)\n");
18 scanf ("%d", &d);
19
20
21
22
23 //impressão dos valores (B,r)
24 printf("valores (B.10^-5,r):\n");
25 for(r=1;r<=d;r++)
26 {
27
28 B = u_0*c_i/(2*Pi*r*pow(10,-2)); // r =r*pow(10,-2) convertendo em cm e calculando B
29 B = B*pow(10,5); //colocado B n potencia de 10^-5
30 printf("(%.3f,%d)\n", B,r);
31 }
32 }
```

7. Considere uma tensão alternada definida pela seguinte função:

$$v(t) = V_p \sin(\omega t + \theta)$$

onde:

- V_p é o valor máximo que a tensão pode ter
- ω é a velocidade angular(rad/s) de acordo com uma frequência f ($\omega = 2\pi f$)
- θ é o ângulo ou fase inicial

Faça um programa que tem as seguintes entradas saídas:

entradas:

frequência (em Hz)

fase inicial (em graus)

valor máximo da tensão

tempo de análise (em segundos)

Saída

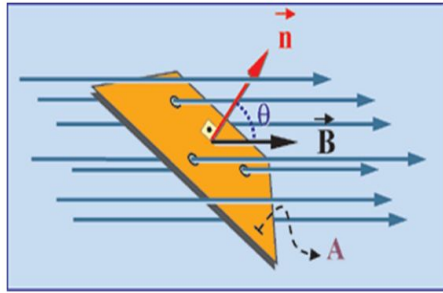
valores da tensão em cada instante de tempo:

$(v(t_0), t_0), (v(t_1), t_1), (v(t_2), t_2), (v(t_3), t_3), \dots (v(t), t)$

considere $t_0 = 0$ e a variação do tempo em intervalos de 100ms.

```
1 // Programa que calcula  $v(t) = V_p \sin(\omega t + \theta)$  para um intervalo de tempo t
2 #include <stdio.h> //inclusão da biblioteca
3 #include <math.h>
4 #define Pi 3.24 //declaração de constante
5 main ()
6 {
7     float Vp,Vt,w,t,teta,f,tempo; // declaração das variáveis do programa
8
9     //solicitação dos dados
10    printf ("Entre com o valor da frequencia (Hz)\n");
11    scanf ("%f", &f);
12
13    printf ("Entre com o valor da fase inicial(graus)\n");
14    scanf ("%f", &teta);
15
16    printf ("Entre com o valor maximo da tensao (V)\n");
17    scanf ("%f", &Vp);
18
19    printf ("Entre com o valor do tempo de analise (ms)\n");
20    scanf ("%f", &tempo);
21
22    w = 2*Pi*f; // cálculo da velocidade angular
23    teta = teta*Pi/180; //convertendo de graus para radiano
24    tempo = tempo*pow(10,-3); //convertendo o valor em ms
25
26    printf("valores (V(t),t):\n"); //impressão dos valores (V(t),t)
27    for(t=0; t<=tempo; t=t+0.1)
28    {
29        Vt = Vp*sin(w*t+teta); //calculo de  $v(t) = V_p \sin(\omega t + \theta)$ 
30        printf("%.2f,%0.1f\n", Vt,t);
31    }
32 }
```

8. Sabe-se que o fluxo magnético é calculado pela seguinte expressão:



$$\phi = \vec{B} \cdot A \cdot \cos\theta$$

onde:

- ϕ é o valor do fluxo (Wb)
- \vec{B} é o valor do campo magnético (T)
- A é a área da superfície do material
- θ é o ângulo de incidência em relação ao vetor normal \vec{n} do campo magnético sobre a superfície de área A

Faça um programa que calcule o fluxo magnético de acordo com a escolha feita pelo usuário. Assim, deve ser fornecido um menu com as opções dos tipos de superfícies (quadrado, retângulo, trapézio, triângulo), o valor do campo magnético e o ângulo de incidência.

```

1 // Programa que calcula  $\phi = B \cdot A \cdot \cos\theta$ 
2 #include <stdio.h> //inclusão da biblioteca
3 #include <math.h>
4 main ()
5 {
6     float Bc, fluxo, A, teta, l, a, b, B, h; // declaração das variáveis do programa
7     int tipo;
8
9     //solicitação dos dados
10    printf ("Entre com o tipo de superficie \n");
11    printf ("1: Quadrado\n 2: Retangulo\n 3: Trapezio\n 4: Triangulo\n");
12    scanf ("%d", &tipo);
13
14    if(tipo==1){
15        printf ("Entre com o lado do quadrado(cm)\n");
16        scanf ("%f", &l);
17        A = l*l;
18    }
19
20
21
22    if(tipo==2){
23        printf ("Entre com os lados do retangulo(cm)\n");
24        scanf ("%f", &a);
25        scanf ("%f", &b);
26        A = a*b;
27    }
28
29
30
31    if(tipo==3){
32        printf ("Entre com a base menor, base maior e altura do trapezo(cm)\n");
33        scanf ("%f", &b);
34        scanf ("%f", &B);
35        scanf ("%f", &h);
36
37        A = (b+B)*h/2;
38    }
39
40
41    if(tipo==4){
42        printf ("Entre com a base e altura do triangulo(cm)\n");
43        scanf ("%f", &b);
44        scanf ("%f", &h);
45
46        A = b*h/2;
47    }
48
49    printf ("Entre com o valor do campo B(10^-3)\n");
50    scanf ("%f", &Bc);
51
52    printf ("Entre com o valor do angulo (graus)\n");
53    scanf ("%f", &teta);
54
55    A = A*pow(10, -4); //convertendo a área em m^2
56    teta = teta*Pi/180; //convertendo de graus para radiano
57    Bc = Bc*pow(10, -3); // convertendo o campo em potencia de 10^-3
58
59    fluxo = Bc*A*cos(teta);
60    fluxo = fluxo*pow(10, 6); //convetendo em 10^-6
61    printf ("O valor do fluxo(10^-6): %.1f\n", fluxo);
62 }

```

9. Último chefe do Mário 🍄 🍄

O cálculo de algumas características nos circuitos elétricos são essenciais no sistema elétrico.

Existem dispositivos que, através de sensores, capturam os dados das redes elétricas e realizam a computação necessária para a apresentação dos valores dessas características. Dentre essas características está a tensão eficaz, também conhecida por V_{rms} , obtida pela fórmula a seguir:

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T V(t)^2 dt}$$

onde T é o período da função.

Essa fórmula pode ser reescrita na forma de somatório para a integral definida, de forma que, para esse caso particular:

$$\int_0^T x^2 dx \approx \sum_{i=1}^n \left(\frac{f(x_i)^2 + f(x_{i-1})^2}{2} \right) \cdot (x_i - x_{i-1})$$

Onde n é o número de pontos (x,y), sendo considerados no cálculo da função.

Dado a seguinte expressão:

$$v(t) = 311 \sin(377t)$$

onde $\theta = 377t$ rad

com t variando de 0 a 16,67 ms em intervalos de n amostras. Escreva um programa que determine o valor n para um valor rms de 220V com erro de aproximação de 10^{-3} .

```

1 // Programa que calcula Vrms=v1TfT0V(t)2dt
2 //inclusão da biblioteca
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include<math.h>
6
7 main ()
8 {
9 // declaração das variáveis do programa
10 float Vrms,Vdef,T,t_prox,t_ant,erro,Dif,passo,soma,f_ant,f_prox;
11 int n,i;
12
13 //ajustes iniciais
14 T= 16,67*pow(10,-3);
15 erro = pow(10,-3); // erro definido pelo usuário
16 Vdef = 220; // valor inicial da tensão
17
18 //inicio do laço
19 n=1;
20 do{
21 passo = T/n; //tamanho do passo
22 t_ant =0; //valor início xo na função f(x)
23 soma =0; //variável para armazenar o somatorio dos valores de f(x)
24
25 for(i=1; i<=n; i++)
26 {
27 t_prox = passo*i;
28 f_ant = 311*sin(377*t_ant);
29 f_prox = 311*sin(377*t_prox);
30 soma = soma + ((pow(f_prox,2) + pow(f_ant,2))/2)*(t_prox - t_ant);
31
32 t_ant = t_prox;
33 }
34
35 Vrms = sqrt(soma/T); // cálculo do valor Vrms
36 Dif = abs(Vdef - Vrms); ///cálculo do erro
37
38 if(Dif>erro)
39 n++;
40
41 }while(Dif>erro); //condição de parada
42
43
44 //impressao do valor final dos dados
45 printf("O valor de Vrms: %.2f\n",Vrms);
46 printf("O valor de n foi: %d",n);
47
48 }

```