

Lista de exercícios:

Grupo I - programa seqüenciais simples

1. Fazer um programa em C que pergunta um valor em metros e imprime o correspondente em decímetros, centímetros e milímetros.
2. Fazer um programa em C que imprime uma tabela com a tabuada de 1 a 9
3. Fazer um programa que solicita um número decimal e imprime o correspondente em hexa e octal.
4. Fazer um programa em "C" que pergunte um valor em graus Fahrenheit e imprime no vídeo o correspondente em graus Celsius usando as fórmulas que seguem.
 - a) Usar uma variável double para ler o valor em Fahrenheit e a fórmula $C = (f - 32.0) * (5.0 / 9.0)$.
 - b) Usar uma variável int para ler o valor em Fahrenheit e a fórmula $C = (f - 32) * (5 / 9)$.
5. Fazer um programa em "C" que solicite 2 números e informe:
 - a) A soma dos números;
 - b) O produto do primeiro número pelo quadrado do segundo;
 - c) O quadrado do primeiro número;
 - d) A raiz quadrada da soma dos quadrados;
 - e) O seno da diferença do primeiro número pelo segundo;
 - f) O módulo do primeiro número.

Grupo II - explorando os comandos de controle de fluxo

1. Faça um programa em "C" que lê dois valores e imprime:
 - se o primeiro valor for menor que o segundo, a lista de valores do primeiro até o segundo;
 - se o primeiro valor for menor que o segundo a lista de valores do segundo até o primeiro em ordem decrescente;
 - se ambos forem iguais a mensagem "valores iguais".
2. Fazer um programa que imprime a tabela ASCII (código decimal, código hexa, caracter) para os códigos de 0 a 127.
3. Fazer um programa em "C" que lê o preço de um produto e inflaciona esse preço em 10% se ele for menor que 100 e em 20% se ele for maior ou igual a 100.

OBS: não use o comando "if" ou o operador de condição "?".

4. Fazer um programa que lê um valor, um operador (+,-,*,/) e outro valor e imprime o resultado da expressão:

<valor 1> <operador> <valor 2>

5. Escrever um programa em "C" que solicita as notas das duas provas feitas por cada um dos alunos de uma turma (as notas tem de estar no intervalo [0 10]) e imprime para cada um a média das notas. O programa deve parar imediatamente após ter sido digitado o valor 50 para a nota da primeira prova.

Grupo III - Explorando comandos do pré-processador

1. Escrever a seqüência de comandos do pré-processador que define o tipo de dados BOOL (com as constantes TRUE e FALSE) caso isso ainda não tenha sido feito.
2. Fazer um programa em "C" que le dois números e utiliza uma função chamada "soma" e outra chamada "subtracao" para imprimir a soma e a diferença entre os dois números. Ambas funções devem receber dois inteiros como parâmetro e retornar um inteiro como resultado. O programa deve ter duas versões para os protótipos das funções selecionáveis através da definição das constantes: "KR" (Kernighan e Ritchie) e "ANSI" (padrão ANSI).

Grupo IV - Explorando o uso de funções

1. Fazer uma função que calcula a enésima potência de uma variável real x:

$$f(x, n) = x^n$$

2. Fazer uma função que calcula o fatorial de um número. Implementar uma versão recursiva e uma versão com laço.

3. Dado que podemos calcular e^x por:

$$e^x = 1 + x + x^2/2! + x^3/3! + \dots$$

Fazer um trecho de programa em "C" que lê um valor para x e calcula o valor de e^x . O valor deve ser calculado enquanto o termo calculado for maior que $10E-6$.

4. Exercício: fazer um programa em "C" que solicita o total gasto pelo cliente de uma loja, imprime as opções de pagamento, solicita a opção desejada e imprime o valor total das prestações (se houverem).

1) Opção: a vista com 10% de desconto

2) Opção: em duas vezes (preço da etiqueta)

3) Opção: de 3 até 10 vezes com 3% de juros ao mês (somente para compras acima de R\$ 100,00).

OBS: fazer uma função que imprime as opções solicita a opção desejada e retorna a opção escolhida. No programa principal, testar a opção escolhida e ativar a função correspondente (uma função para cada opção).

Grupo V - Ponteiros

1. Procure determinar quais valores são impressos ao final deste programa. Confira sua resposta testando o programa no micro. Execute-o passo a passo conferindo o valor das variáveis em cada momento.

a) void main()

```
{  
int a,b,*c;  
  
a = 3;  
b = 4;  
c = &a;  
b++;  
*c = a+2;  
printf("%d %d",a,b);  
}
```

b) void main()

```
{  
int a,b,*c;  
  
a = 4;  
b = 3;  
c = &a;  
*c = *c + 1;  
c = &b;  
b = b+4;  
printf("%d %d %d",a,b,*c);  
}
```

c) void main()

```
{  
int a,b,*c,*d,*f;  
  
a = 4;  
b = 3;  
c = &a;  
d = &b;  
*c /= 2;  
f = c;  
c = d;  
d = f;  
printf("%d %d",*c,*d);  
}
```

d)

int calcula(int);

void main()

```
{
```

```

int a,b,c;
char d;
a=1;b=2;c=3;d='A';
a+=b*c;
d=(a>7)?d-1:d+1;
b = calcula(b);
c = calcula(calcula(a));
a = c++;
printf("%d - %d - %d - %c\n",a,b,c,d);
}

```

```

int calcula(int x)
{
    int i;

    if ((x=x*2)>5) return(x+3);
    for(i=0;i<10;i++)
    {
        if (i<5) continue;
        if (x>8) break;
        x+=2;
    }
    return(x);
}

```

Grupo VI - Vetores

1. Fazer um programa em "C" que lê 10 valores e imprime o maior e o menor valores lidos.
2. Fazer um programa que lê um conjunto de 10 valores e os imprime ordenados.
3. Fazer uma rotina que recebe como parâmetro um array de 5 posições contendo as notas de um aluno ao longo do ano e devolve a média do aluno.
4. Fazer uma rotina que recebe um array do tipo double e o número de valores que devem ser solicitados ao usuário e devolve o array preenchido com os valores digitados.
5. Fazer um programa em "C" que lê um conjunto de 10 valores inteiros e verifica se algum dos valores é igual a média dos mesmos.
6. Fazer um programa que lê valores para uma matriz do tipo "float" de 5 linhas por 3 colunas e imprime a diferença entre a média dos elementos das colunas pares e a média dos elementos das linhas ímpares.

Grupo VII - Strings

1. Fazer um programa em "C" que lê um string qualquer de no máximo 80 caracteres e imprime:
 - Quantos caracteres tem o string;

- Quantos caracteres são de pontuação;
- Quantos caracteres são números;
- Quantos caracteres são minúsculas.

2. Fazer um programa em "C" que lê um string contendo palavras separadas por um espaço em branco cada e as imprime uma abaixo das outras.
3. Fazer um programa em "C" que lê um string do teclado e se utiliza de uma rotina recursiva para imprimir o string de maneira normal e de trás para diante.
4. Fazer um programa em "C" que pergunta o nome, o endereço, o telefone e a idade de uma pessoa e monta um string com a seguinte frase:

"Seu nome é ..., você tem ... anos, mora na rua ... e seu telefone é"

5. Fazer uma rotina que aguarda um string do teclado e retorna o valor 1 se o string digitado foi "SIM" e 0 se o string digitado foi "NAO". A rotina só deve retornar alguma coisa se o string digitado for "SIM" ou "NAO".
6. Fazer uma rotina que recebe um string como parâmetro e imprime quantas palavras (separadas por espaços em branco) o mesmo contém.
7. Implemente um rotina que faça a mesma coisa que a função "strcpy".
8. Fazer um programa em "C" que solicita um número inteiro e soletra o mesmo na tela.

Ex:

124: um, dois, quatro

9. Fazer um programa em "C" que leia nomes de pessoas compostos por um pré-nome e um sobrenome separados por um espaço em branco e imprima:
 - A lista de nomes em ordem alfabética do pré-nome;
 - A lista de nomes em ordem alfabética de sobrenomes;
 - O número médio de letras por nome.

Obs:

- a quantidade de nomes lidos deve ser menor ou igual a 20;
 - uma sequência de nomes é encerrada pela palavra "FIM" ou quando for lido o 20 nome.
10. Escrever uma função que recebe um string e um caracter como parâmetro e remove todas as ocorrências do caracter do string.
 11. Escreva uma função em "C" que receba um string um caracter e o índice de uma posição do string como parâmetro e insira o caracter na posição "empurrando" todos os demais para o lado.
 12. Fazer uma rotina em "C" que recebe um string como parâmetro e devolve o endereço do primeiro caracter branco encontrado.

13. Fazer uma rotina que recebe como parâmetro um string contendo um número e um inteiro indicando a base na qual o número está expresso, retornando o seu valor em decimal

Ex: `conv2dec("345", 8) => 229`

Grupo VIII - Passagem de parâmetros e estruturas

1. Fazer uma função que retorna a soma, a diferença e o produto entre dois números.
2. Fazer uma função em "C" que retorna a razão entre dois números. A função deve retornar pelo comando `return` o valor 1 se a operação foi possível e o valor 0 se a operação não foi possível (divisão por zero, por exemplo). O resultado da divisão deve retornar por um parâmetro por referência.
3. Fazer uma rotina em "C" que recebe um vetor de números inteiros como parâmetro onde todos os valores exceto o último são positivos e devolve:

- a média dos valores do vetor;
- o menor valor do vetor (sem considerar o último)
- o maior valor do vetor

4. Fazer uma função para ler e retornar o valor das 3 notas de um aluno.
5. Construir um programa em "C" que implementa uma agenda eletrônica. O programa deve ter um menu com as seguintes opções:
 - Entrar um novo nome na agenda.
 - Imprimir na tela os dados de uma das pessoas cadastradas (conforme solicitação).
 - Imprimir a lista de nomes cadastrados que comecem pela letra indicada.
 - Fim

Cada entrada da agenda deve ter os seguintes campos:

```
char nome[30];  
char endereco[100];  
char fone[10];  
long int CEP;
```

Obs: a agenda deve ter capacidade para 100 entradas.

6. Fazer um programa em "C" que lê uma lista de 20 produtos e e preços e armazena-os em um array do tipo da estrutura abaixo. O programa deve, em seguida, ordenar o vetor em ordem alfabética de nome de produto e inflacionar os produtos cujo valor for menor que 100 em 5%. Por fim a lista de produtos/preços deve ser impressa.

OBS: usar uma rotina que recebe uma estrutura do tipo `PROD` com parâmetro e atualiza o preço, uma que lê os dados para a estrutura do tipo `PROD` e outra capaz de imprimir a estrutura.

```
typedef struct
{
    char nome[80];
    float preco;
} PROD;
```

7. Construir um programa em "C" que implementa uma agenda eletrônica. O programa deve ter um menu com as seguintes opções:

- 1- Entrar um nome na agenda
- 2- Imprimir na tela os dados de uma das pessoas cadastradas (consulta por nome)
- 3- Imprimir na impressora a lista dos nomes que começam pela letra indicada