

Aula Prática 7

Prazo final: 29/04/2018

26 de abril de 2018

Instruções gerais

Use um `scanf` para cada entrada, ou seja, se você tem que ler nove dígitos do teclado, use a função `scanf` nove vezes, uma para cada valor a ser lido.

Não utilize a função `system("pause")` nos arquivos a serem submetidos.

Não utilize mensagens para a entrada dos dados (ex.: "Entre com o numero 1: "), utilize a função `printf` apenas para imprimir as mensagens descritas nas atividades.

Atenção: Para esta lista de exercícios é necessário utilizar funções conforme pedem as atividades, caso o contrário, a nota será 0, independente da nota do Moodle.

1 Área, perímetro e volume (apv.c)

Implemente funções para calcular o perímetro de um círculo, área de um círculo e o volume de uma esfera. O programa deve utilizar a seguinte função `main()`:

```
int main(){
    float r, v;
    printf("Raio: ");
    scanf("%f", &r);

    v = calcular_perimetro(r);
    printf("Perimetro: %.2f\n", v);

    v = calcular_area(r);
```

```

    printf("Area: %.2f\n", v);

    v = calcular_volume(r);
    printf("Volume: %.2f\n", v);

    return 0;
}

```

Considere $\pi = 3,1415$ e as fórmulas:

Perímetro: $p = 2 \times \pi \times r$

Área: $a = \pi \times r^2$

Volume: $v = 4 \times \pi \times \frac{r^3}{3}$

- **Entrada:** Um número real.
- **Saída:** Definida pela função **main** acima

2 Velocidade e Aceleração (velacc.c)

Faça um programa que receba um valor de velocidade (v_0), um valor de aceleração (a) e um valor de tempo (t) e retorne qual a velocidade final e a distância percorrida por um veículo após o intervalo de tempo t , com velocidade inicial igual a v_0 e aceleração igual a a .

Neste exercício você deverá implementar uma função para calcular a velocidade final e outra para calcular a distância percorrida.

Considere as fórmulas:

Velocidade final: $v = v_0 + at$

Distância percorrida: $s = v_0t + \frac{at^2}{2}$

- **Entrada:** Três números reais na seguinte ordem: Velocidade, aceleração e tempo.
- **Saída:** Somente as seguintes mensagens de texto, utilizando a função **printf**:
 "Velocidade final: %.2f\n"
 "Distancia percorrida: %.2f\n"

3 Conceito (conceito.c)

Faça uma função que receba quatro notas de atividades e obtenha o conceito do aluno. Se qualquer uma das notas estiver fora do intervalo entre 0 e 100 a função deve retornar 0, caso o contrário, deve calcular a média aritmética das quatro notas, obter o conceito conforme a tabela abaixo e retornar 1:

O protótipo da função deve ser:

```
int obter_conceito(int n1, int n2, int n3, int n4, char *conceito);
```

- **Entrada:** 4 valores inteiros.
- **Saída:** Somente **uma** das seguintes mensagens: "Nota invalida!\n" "Conceito: %c\n"

Pontuação	Conceito
90 a 100	A
80 a 89	B
70 a 79	C
50 a 69	D
40 a 49	E
0 a 39	F

4 Combinações (comb.c)

O número de combinações possíveis de n objetos tomados de p a p pode ser calculado pela fórmula:

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

Faça um programa que receba os valores de n e p e calcule o valor de $C_{n,p}$. O programa deve implementar as funções definidas pelos protótipos abaixo:

```
int fatorial(int v); int combinacao(int n, int p, int *cnp);
```

A função `combinacao` deve retornar 0 se n ou p forem negativos, ou se p for maior do que n . Caso o contrário, a função deve armazenar o valor de $C_{n,p}$ na variável `cnp` e retornar o valor 1.

- **Entrada:** Dois números inteiros na seguinte ordem: n depois p .
- **Saída:** Somente **uma** das seguintes mensagens:
"Parametro invalido!\n"
"Combinacao: %d\n"

5 Conta bancária (conta.c)

O programa abaixo simula operações básicas de saque e depósito em uma conta bancária, entretanto possui erros na utilização de ponteiros. Identifique estes erros e realize as correções necessárias.

Atenção, os cabeçalhos das funções não podem ser alterados.

```
#include<stdio.h>

void depositar(float d, float *saldo){
    saldo += d;
}

int sacar(float d, float *saldo){
    if (saldo > d){
```

```

        *saldo -= d;
        return 1;
    } else {
        return 0;
    }
}

int main(){
    float *saldo, aux;
    printf("Saldo inicial: ");
    scanf("%f", saldo);

    printf("Depositar: ");
    scanf("%f", &aux);
    depositar(aux, saldo);

    printf("Sacar: ");
    scanf("%f", &aux);
    sacar(aux, &saldo);

    printf("Saldo %.2f\n", saldo);
    return 0;
}

```

- **Entrada:** Já definida pela função **main()** do código acima.
- **Saída:** Já definida pela função **main()** do código acima.