Professor Pietro Martins de Oliveira

ALGORITMOS

do início ao fim

PACOTE DE EXERCÍCIOS 2: SUB-ROTINAS EM C

1) (Adaptado de ASCENCIO e CAMPOS, 2008) O custo de um carro novo ao consumidor final é o preço de fábrica somado ao percentual de lucro do distribuidor, acrescido dos impostos aplicados ao preço de fábrica. Faça um programa que receba o preço de fábrica de um veículo, o percentual de lucro do distribuidor e o percentual de impostos. Em cada item, crie uma função distinta para calcular e retornar:

a) o valor correspondente ao lucro do distribuidor;
b) o valor correspondente aos impostos;
c) o preço final do veículo.

Após criar cada uma das funções, desenvolva um algoritmo que declare e invoque cada uma das funções, mostrando o lucro do distribuidor, os impostos e o valor final do veículo.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

```
Exemplo de execução - Exercício 1 - Caso de teste
Insira o preço de fábrica:
20000
Insira a porcentagem de lucro do distribuidor:
30
Insira a porcentagem de impostos:
30
Lucro do distribuidor: R$ 6000
Impostos: R$ 6000
Valor final: R$ 32000
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

```
SUB-ROTINAS - Exercício 1 - Solução
01 #include <stdio.h>
02 float calcLucroDist(float p_fab, float por_dist){
03     return p_fab * por_dist / 100.0;
04 }
05 float calcImpostos(float p_fab, float por_imp){
06     return p_fab * por_imp / 100.0;
07 }
08 float calcPrecoFinal(float p_fab, float v_dist, float v_im
   p){
09     return p_fab + v_dist + v_imp;
10 }
11 int main(){
12     float preco_f, porce_dist, porc_i, lucr_dist, imp, vlr
   _f;
13     printf("Insira o preço da fábrica:\n");
```

```
14      scanf("%f", &preco_f);
15      printf("Insira a porcentagem de lucro do distribuidor:
    \n");
16      scanf("%f", &porce_dist);
17      printf("Insira a porcentagem de impostos:\n");
18      scanf("%f", &porc_i);
19      lucr_dist = calcLucroDist(preco_f, porce_dist);
20      imp = calcImpostos(preco_f, porc_i);
21      vlr_f = calcPrecoFinal(preco_f, lucr_dist, imp);
22      printf("Lucro do distribuidor: R$ %.2f.\n", lucr_dist)
    ;
23      printf("Impostos: R$ %.2f.\n", imp);
24      printf("Valor final: R$ %.2f.\n", vlr_f);
25  }
```

2) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o número de horas trabalhadas por um gestor e o valor do salário mínimo vigente. Crie uma função que calcule o salário a receber do gestor, seguindo as regras abaixo:

I - a hora trabalhada vale a metade do salário mínimo;
II - o salário bruto equivale ao número de horas trabalhadas multiplicado pelo valor da hora trabalhada;
III - o imposto equivale a 3% do salário bruto;
IV - o salário a receber equivale ao salário bruto menos o imposto.
Crie um algoritmo que invoque a respectiva função e mostre o salário a receber.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

```
Exemplo de execução - Exercício 2 - Caso de teste
Insira número de horas trabalhadas:
180
Insira valor do salário mínimo:
1000
Salário a receber:  87300
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

```
SUB-ROTINAS - Exercício 2 - Solução
```

```c
#include <stdio.h>
float calcSalRec(float n_hr, float s_min){
    float hr_t, s_brt, imp, s_rec;
    hr_t = s_min/2.0;
    s_brt = n_hr * hr_t;
    imp = s_brt * 0.03;
    s_rec = s_brt - imp;
    return s_rec;
}
int main(){
    float horas_trab, sal_min, sal_rec;
    printf("Insira o número de horas trabalhadas:\n");
    scanf("%f", &horas_trab);
    printf("Insira o valor do salário mínimo:\n");
    scanf("%f", &sal_min);
    sal_rec = calcSalRec(horas_trab, sal_min);
    printf("Salário a receber: %f.\n", sal_rec);
}
```

3) (Adaptado de ASCENCIO e CAMPOS, 2008) Pedro comprou um saco de ração para seus gatos, com o peso em quilos. Faça uma função que receba o peso do saco de ração, em quilos, o número de gatos e a quantidade de ração fornecida para cada gato por dia, em gramas. A função deve retornar o total de quilos de ração restante no saco, após um dia de consumo. Assim sendo, considerando que Pedro possui dois gatos, crie um algoritmo que invoque a função recém criada para calcular e mostrar quanto restará de ração no saco após cinco dias.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

```
Exemplo de execução – Exercício 3 - Caso de teste
Qual o peso do saco (quilos)?
20
Qual o número de gatos?
2
Qual o peso da porção diária (gramas)?
250
Após 5 dias, sobrarão:  17.5 Kg de ração
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

```
SUB-ROTINAS – Exercício 3 – Solução
```

```c
#include <stdio.h>
float calcRacaoSaco(float s, float n, float q){
    float r = s - (n*q/1000.0);
    return r;
}
int main(){
    float saco, n_gatos, qtde_gr, sobra;
    printf("Qual o peso do saco (quilos)?\n");
    scanf("%f", &saco);
    printf("Qual o número de gatos?\n");
    scanf("%f", &n_gatos);
    printf("Qual o peso da porção diária (gramas)?\n");
    scanf("%f", &qtde_gr);
    sobra = calcRacaoSaco(saco, n_gatos, qtde_gr);
    sobra = calcRacaoSaco(sobra, n_gatos, qtde_gr);
    sobra = calcRacaoSaco(sobra, n_gatos, qtde_gr);
    sobra = calcRacaoSaco(sobra, n_gatos, qtde_gr);
    sobra = calcRacaoSaco(sobra, n_gatos, qtde_gr);
    printf("Após 5 dias, sobrarão: %.3f Kg de ração.\n", s
obra);
}
```

4) (Adaptado de ASCENCIO e CAMPOS, 2008) Cada degrau de uma escada tem X cm de altura. Faça uma função que receba essa altura, em centímetros, e a altura que o usuário deseja alcançar ao subir a escada, em metros. A função deve retornar o número de degraus necessários para se atingir a altura desejada (desprezando a altura do próprio usuário). Em seguida, crie um algoritmo para que o usuário possa informar os dados de entrada da função e, ao final, calcule e mostre o número de degraus.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

```
Exemplo de execução - 4 - Caso de teste
Insira a altura de cada degrau (cm):
25
Insira a altura da escada (m):
2
Número de degraus:  8
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

```
SUB-ROTINAS - Exercício 4 - Solução
01 #include <stdio.h>
```

```c
int calcDegraus(int a_deg, float a_esc){
    return (a_esc*100) / a_deg;
}
int main(){
    float escada;
    int degrau, n_degraus;
    printf("Insira a altura de cada degrau:\n");
    scanf("%d", &degrau);
    printf("Insira a altura da escada (m):\n");
    scanf("%f", &escada);
    n_degraus = calcDegraus(degrau, escada);
    printf("Número de degraus: %d.\n", n_degraus);
}
```

5) (Adaptado de ASCENCIO e CAMPOS, 2008) Sabe-se que o quilowatt de energia custa um milésimo do salário mínimo. Faça um procedimento que receba o valor do salário mínimo e quantidade de quilowatts consumida por uma residência. O procedimento deve calcular e retornar através de passagem de parâmetros por referência:

a) o valor, em reais, de cada quilowatt;
b) o valor, em reais, a ser pago por essa residência;
c) o valor, em reais, a ser pago com desconto de 15%.

Sabendo disso, desenvolva um algoritmo que peça para o usuário inserir o valor do salário mínimo e a quantidade de quilowatts consumida. Invoque o respectivo procedimento e mostre, na tela, as informações dos itens a), b) e c).

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

```
Exemplo de execução – Exercício 5 - Caso de teste
Insira o salário mínimo:
1000
Insira a quantidade de KW gastos:
200
Valor de 1 KW (em R$):  1
Valor a ser pago (em R$):  200
Valor com desconto de 15% (em R$):  170
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

```
SUB-ROTINAS – Exercício 5 – Solução
```

```c
#include <stdio.h>
void calcKW(float v_sal, float q_kw, float *v_kw, float *v_rs, float *v_dsc){
    *v_kw = v_sal / 1000.0;
    *v_rs = *v_kw * q_kw;
    *v_dsc = *v_rs - (*v_rs)*0.15;
}
int main(){
    float val_sal, qtde_kw, val_kw, val_reais, val_desc;
    printf("Insira o salário mínimo:\n");
    scanf("%f", &val_sal);
    printf("Insira a quantidade de KW gastos:\n");
    scanf("%f", &qtde_kw);
    calcKW(val_sal, qtde_kw, &val_kw, &val_reais, &val_desc);
    printf("Valor de 1 KW (em R$): %.2f.\n", val_kw);
    printf("Valor a ser pago: R$ %.2f.\n", val_reais);
    printf("Valor com desconto de 15%%: R$ %.2f.\n", val_desc);
}
```

6) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um procedimento que receba um número real, calcule e retorne:

a) a parte inteira desse número;
b) a parte fracionária desse número.

Crie um algoritmo que peça para o usuário inserir o número real e, em seguida, calcule e mostre o que se pede nos itens a) e b).

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

```
Exemplo de execução - Exercício 6 - Caso de teste
Insira um número real:
3,14
Parte inteira:  3
Parte facionária:  0.14
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

```
SUB-ROTINAS - Exercício 6 - Solução
01 #include <stdio.h>
```

```c
void numReal(float n, int *i, float *fr){
    *i = (int)n;
    *fr = n - (float)*i;
}
int main(){
    float num, parte_frac;
    int parte_int;
    printf("Insira um número real:\n");
    scanf("%f", &num);
    numReal(num, &parte_int, &parte_frac);
    printf("Parte inteira: %d.\n", parte_int);
    printf("Parte fracionária: %f.\n", parte_frac);
}
```

7) (Adaptado de ASCENCIO e CAMPOS, 2008) Crie um procedimento que receba a quantidade de dinheiro em reais que uma pessoa que vai viajar possui. Essa pessoa vai passar por vários países e precisa converter seu dinheiro em dólares, euro e libra esterlina. Sabe-se que a cotação do dólar é de R$ 4,00, do euro é R$ 4,25 e do iene é R$ 0,10. O procedimento deverá fazer a leitura dos dados do usuário e exibir o resultado das conversões diretamente, sem passagem de parâmetros. Desenvolva um algoritmo que invoque o procedimento para realizar os cálculos.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

```
Exemplo de execução - Exercício 7 - Caso de teste
Insira o valor em reais:
100
Em dólares:  25
Em euros:  23.5294117647059
Em ienes:  1000
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

```c
SUB-ROTINAS - Exercício 7 - Solução
#include <stdio.h>
void cotacoes(){
    float reais;
    printf("Insira o valor em reais:\n");
    scanf("%f", &reais);
    printf("Em dólares: %.2f.\n", reais/4.00);
    printf("Em euros: %.2f.\n", reais/4.25);
    printf("Em ienes: %.2f.\n", reais/0.1);
```

```
09 }
10 int main(){
11     cotacoes();
12 }
```