

Teste Laboratorial (Turmas P1, P2 e P11)

Duração: 50 min

13 de dezembro de 2018

Sem consulta

A empresa **Chocolates&Doces** está a preparar a época natalícia que se avizinha, na qual costuma ter bastantes encomendas de **“Pai Natais”** de chocolate. Este ano está a programar vender um modelo especial de **“Pai Natais”**.

Pretende-se desenvolver uma aplicação que auxilie a **Chocolates&Doces** na contabilização de embalagens necessárias para empacotar um conjunto de encomendas de **“Pai Natais”** de chocolate. Assuma que as encomendas são feitas em número de **“Pai Natais”** e que cada caixa permite embalar 10 **“Pai Natais”**.

- a) Implemente uma função que receba como argumento a quantidade (número) de **“Pai Natais”** de uma encomenda. A função deve retornar o número de caixas necessário para empacotar todas os **“Pai Natais”** dessa encomenda.
- b) Desenvolva um programa que vá pedindo ao utilizador o número de **“Pai Natais”** de cada encomendas, garantindo que são valores superiores a 0 (zero), e vá apresentando o número de caixas necessário para cada encomenda. Para este efeito deve tirar partido da função definida na alínea anterior. Quando o utilizador indicar um número de **“Pai Natais”** nulo (zero) o programa deve terminar a sua execução imprimindo um resumo final constituído pelo número total de caixas para todas as encomendas, maior número de caixas necessário para uma encomenda e número médio de caixas por encomenda.

Nota: Caso não tenha conseguido resolver a alínea **a)**, assuma que já existe uma função feita e use-a para resolver a alínea **b)**.

Exemplo de execução do programa:

Numero de **Pai Natais** da encomenda 1: **45**

Numero de caixas necessário: 5

Numero de **Pai Natais** da encomenda 2: **70**

Numero de caixas necessário: 7

Numero de **Pai Natais** da encomenda 3: **-1**

Numero de **Pai Natais** da encomenda 3: **27**

Numero de caixas necessário: 3

Numero de **Pai Natais** da encomenda 4: **0**

Resumo das encomendas:

Numero total de caixas: 15

A maior encomenda necessita de 7 caixas

Media de caixas por encomenda: 5.00

Na resolução de cada uma das alíneas deve apresentar:

- **Análise do problema** (dados de entrada, resultados pretendidos, conhecimento requerido e estratégia) **(25%)**;
- **Algoritmo** (pseudocódigo ou fluxograma) **para a solução que propõe (50%)**;
- **Programa completo em linguagem C (25%)**.

Função

Especificação ou análise da função:

Dados de Entrada:

num (inteiro) - Número de “Pai Natais” de chocolate de uma encomenda

Resultados Pretendidos:

nc (inteiro) - Número de caixas necessárias para empacotar os “Pai Natais” da encomenda

Conhecimento requerido:

POR_CAIXA = 10, cada caixa permite embalar 10 “Pai Natais”

Como calcular o número de caixas?

Inicialmente o número de caixas da encomenda é igual à divisão inteira de *num* por POR_CAIXA

(*nc* ← *num*/POR_CAIXA)

Ver se o resto da divisão inteira de *num* por POR_CAIXA é diferente de zero;

se sim é necessária + 1 caixa para os “Pai Natais” que sobraram (em número inferior a 10)

Estratégia:

A função tem como parâmetro *num*

Calcula e devolve o valor de *nc*

Pseudocódigo:

INÍCIOFUNÇÃO nCaixas

RECEBE (num)

nc ← *num*/POR_CAIXA

sobram ← *num*%POR_CAIXA

// % é o resto da divisão inteira

SE (sobram ≠ 0) ENTÃO

nc ← *nc* + 1

FIM SE

DEVOLVE (*nc*)

FIM FUNÇÃO nCaixas

Codificação em C:

```
int nCaixas(int num)
{
    int sobram;
    int nc;
    nc = num/POR_CAIXA;
    sobram = num%POR_CAIXA;
    if (sobram)
        nc++;
    return nc;
}
```

Programa principal

Especificação ou análise do problema:

Dados de Entrada:

n_{pn} (inteiro) - Número de “Pai Natais” de chocolate de cada uma das encomendas

Resultados Pretendidos:

caixas (inteiro) – Número de caixas necessárias para empacotar cada uma das encomendas

tCaixas (inteiro) – Total de caixas necessárias para empacotar todas as encomendas

maxCaixa (inteiro) – maior número de caixas necessário para uma encomenda

media (real) – número médio de caixas por encomenda

Conhecimento requerido:

Como calcular a média?

Soma de todos os valores a dividir pelo número de valores

Como calcular o máximo?

Inicialmente *maxCaixa* é zero.

Quando aparecer um número de caixas (*caixas*) maior que *maxCaixa* substituir o valor de *maxCaixa*

Estratégia:

Para cada um dos valores de *n_{pn}* positivos:

Calcular *caixas* (número de caixas necessárias para empacotar a encomenda), usando a função *nCaixas*

Somar *caixas* a *tCaixas*

Calcular o máximo de *caixas* até ao momento e guardar em *maxCaixa*

Calcular *media*

Mostar Resumo das Encomendas (*tCaixas*, *maxCaixa*, *media*)

Pseudocódigo:

INÍCIO programa

nencomenda <- 0

tCaixas <- 0

maxCaixa <- 0

FAZER

OBTEM (npn)

ENQUANTO (npn<0)

ENQUANTO (npn>0) FAZER

nencomenda <- nencomenda + 1

caixas ← CHAMA nCaixas(npn)

MOSTRA(caixas)

tCaixas <- tCaixas + caixas

SE (maxCaixa < caixas) ENTAO

maxCaixa <- caixas

FIM SE

FAZER

OBTEM (npn)

ENQUANTO (npn<0)

FIMENQUANTO

MOSTRA(tCaixas) MOSTRA(maxCaixas)

SE (tCaixa ≠ 0) ENTAO

media <- tCaixas/nencomenda

MOSTRA(media)

FIM SE

FIM programa

Codificação em C:

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

#define POR_CAIXA 10

int nCaixas(int num)
{
    int nc, sobram;
    nc = num/POR_CAIXA;
    sobram= num%POR_CAIXA;
    if (sobram)
        nc++;
    return nc;
}

int main()
{
    int npn, caixas, tCaixas=0, maxCaixa=0, nencomenda=0;
    double media;

    do{
        printf("Numero de Pai Natais da encomenda %d: ", nencomenda+1);
        scanf("%d", &npn);
    } while(npn<0);
```

```

while (npn)
{
    nencomenda++;
    caixas = nCaixas(npn);
    printf("Numero de caixas necessário: %d\n", caixas);
    tCaixas+=caixas;
    if (maxCaixa<caixas)
        maxCaixa=caixas;

    do{
        printf("Numero de Pai Natais da encomenda %d: ", nencomenda+1);
        scanf("%d", &npn);
    } while(npn<0);
}

printf("Resumo das encomendas:\n");
printf("Numero total de caixas: %d\n", tCaixas);
printf("A maior encomenda necessita de %d caixas\n", maxCaixa);

if (tCaixas)
{
    media = (double)tCaixas/nencomenda;
    printf("Media de caixas por encomenda: %.2lf\n", media);
}

return 0;
}

```