



# LICENCIATURA EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

## EEC1104 – PROGRAMAÇÃO

### Mini-teste 2 – Conjunto de Problemas

#### 1. O Ginásio

O exercício físico num ginásio consiste em levantar uma barra com um certo peso. A barra é levantada um certo número de vezes, a que se chama de “repetições”. O exercício é composto por várias séries, isto é, ao fim de  $N$  repetições o atleta descansa e depois retoma novamente outras  $N$  repetições. Este procedimento é repetido um certo número de vezes, a que se chama o número de “séries”. Alguns atletas, pela idade avançada ou condição física, não podem ultrapassar um certo limite para o peso total levantado. O atleta escolhe o peso e número de repetições que pretende, mas o exercício termina antes que esse valor limite seja ultrapassado. Desenvolva um programa para calcular o peso total levantado pelo atleta ao longo das séries e repetições, sabendo que esse total não pode ultrapassar o valor limite.

Dados de entrada (a pedir ao utilizador):

- peso – o peso da barra a ser levantada (p.ex. 25,5 Kg)
- rep – o número de repetições (p.ex. 12)
- ser – o número de séries (p.ex. 3)
- max – o valor máximo de peso total a não ultrapassar (p.ex. 70,5 Kg)

Dados de saída (a apresentar no ecrã):

- o peso total levantado (para os valores acima especificados dá 51,0 Kg)
- a) Implemente uma função que devolve o peso total levantado ao fim de um número de repetições, sem que esse peso ultrapasse o máximo.
- b) Implemente uma função que devolve o peso total levantado ao fim de um número de séries, sem que esse peso ultrapasse o máximo. Utilize a função anterior.
- c) Escreva o programa que calcula o peso total levantado. Utilize as funções anteriores.

#### 2. O Banco

Os clientes do banco podem depositar quantias que ficam a render juros. Por exemplo, uma quantia de 100€ a uma taxa de juro de 2% dá 102€ ao fim de 1 ano, 104.04€ ao fim de 2 anos, 106.12€ ao fim de 3 anos, etc. Suponha que para incentivar depósitos mais longos, a taxa de juro cresce ao fim de um certo número de anos. Por exemplo, se é de 2% nos dois primeiros anos, poderá ser de 3% nos dois anos seguintes, depois 4%, etc. Pretende-se desenvolver um programa que calcule a quantia obtida ao fim de um certo número de anos.

Dados de entrada (a pedir ao utilizador):

- quantia – valor inicial depositado pelo cliente (p.ex. 250,00 €)
- anos – número de anos que quantia fica a render juros (p.ex. 12)
- taxajuro – o valor inicial da taxa de juro (p.ex. 2,5%)
- pertaxa – período ao fim do qual a taxa de juro sobe (p.ex. 2 anos)
- aumtaxa – o valor a somar à taxa de juro em cada novo período (p.ex. 0,5%)

Dados de saída (a apresentar no ecrã):

- a quantia obtida no final (para os valores acima especificados dá 388,71 €)
- a) Implemente uma função que devolve a quantia final ao fim de um certo número de anos com taxa de juro fixa.
- b) Implemente uma função que devolve a quantia final ao fim de um certo número de anos com taxa de juro crescente. Utilize a função anterior.
- c) Escreva o programa que calcula a quantia final obtida. Utilize as funções anteriores.

### 3. O Casino

Um dos jogos disponíveis num casino é um jogo de apostas. O jogador não aposta dinheiro; em vez disso, aposta fichas. Cada jogo tem um “ganho”. Por exemplo, se o jogador apostou 20 fichas e o jogo tem um ganho de 5, se ganhar a aposta recebe 100 fichas, senão não recebe nada. Em ambos os casos, o valor apostado não é devolvido. O problema é que os jogos deste casino estão viciados. Tipicamente, um jogo tem um ganho de  $n$  ao fim de  $k$  apostas. Por exemplo, um jogo tem ganho 2 de 3 em 3 jogadas, mas fora disso tem ganho zero. Isto quer dizer que as jogadas 3, 6, 9,... dão ganho 2 e todas as outras tem ganho zero. Em geral, os jogadores do casino apostam mais quando estão a ganhar e menos quando acabaram de perder uma aposta. Assuma que aquilo que o jogador aposta a mais quando ganha é igual ao que aposta a menos quando perde. O jogo termina ao fim de um número de jogadas ou caso o valor da aposta chegue a zero. Pretende-se desenvolver um programa para determinar o saldo final do jogador (em número de fichas).

Dados de entrada (a pedir ao utilizador):

- numfichas – número de fichas que o jogador dispõe inicialmente (p.ex. 100)
- aposta – valor da primeira aposta (em fichas, p.ex. 10)
- num-apostas – número de apostas a realizar (p.ex. 20)
- delta – valor a somar ou a subtrair à aposta, conforme o jogador ganhe ou perca a aposta anterior (p.ex. 3)
- $k$  – número de jogadas até o jogo dar ganho (p.ex. 3)
- $n$  – ganho do jogo de  $k$  em  $k$  jogadas (p.ex. 2)

Dados de saída (a apresentar no ecrã):

- o número de fichas no final do jogo (para os valores acima especificados dá 72)

a) Implemente uma função que dá o ganho do jogo para uma certa jogada, indicada pelo número da jogada.

b) Implemente uma função que devolve o número de fichas do jogador ao fim de uma jogada, indicada pelo seu número. Utilize a função anterior.

c) Escreva o programa que calcula o número de fichas com que fica o jogador no final. Utilize as funções anteriores.

### 4. As Garrafas

Uma empresa que produz refrigerantes vende-os em garrafas com uma certa capacidade. Esta capacidade é medida em centilitros e é um número sempre inteiro. As garrafas são colocadas em grades. Por exemplo, é comum haver grades de 6, 12 ou 24 garrafas. Encher uma grade consiste em encher as garrafas dessa grade. A questão é que algumas das garrafas têm um defeito de fabrico que depende do número da garrafa. De  $k$  em  $k$  garrafas aparece uma que tem menos capacidade que o habitual. A primeira garrafa defeituosa é a que tem o número  $k$ , a segunda  $2k$ , a terceira  $3k$ , etc. Além disso, o defeito agrava-se com o tempo: a primeira garrafa defeituosa tem menos 1 cl que o habitual, a segunda garrafa defeituosa tem menos 2 cl, a terceira menos 3 cl, e assim por diante. Pretende-se desenvolver um programa para determinar quantos centilitros são necessários para encher uma grade de garrafas, sabendo que algumas das garrafas têm este defeito.

Dados de entrada (a pedir ao utilizador):

- capac – capacidade de cada garrafa (p.ex. 33 cl)
- grade – número de garrafas que cabem numa grade (p.ex. 24)
- $k$  – intervalo entre garrafas defeituosas (p.ex. 5)

Dados de saída (a apresentar no ecrã):

- número de centilitros necessários para encher a grade (para os valores acima especificados dá 782)

a) Implemente uma função que devolve a capacidade de uma certa garrafa, indicada pelo número da garrafa.

b) Implemente uma função que dá o número de centilitros necessários para encher uma garrafa, indicada pelo seu número. Utilize a função anterior.

c) Escreva o programa que calcula o número de centilitros necessários para encher uma grade. Utilize as funções anteriores.

## 5. O Telemóvel

Para calcular o custo de uma chamada de telemóvel é necessário conhecer a sua duração, o preço fixo e o preço por minuto. O custo da chamada é deduzido no saldo do cliente. Durante a chamada o saldo do cliente pode esgotar-se, resultando na interrupção da chamada. Admita que o saldo do cliente é actualizado minuto a minuto, isto é, o cliente é cobrado o minuto seguinte mesmo que não o chegue a utilizar completamente. Os clientes dispõem de um tarifário que lhes dá bónus (em euros) ao fim de um certo período (em minutos). Este bónus é somado ao seu saldo actual de  $X$  em  $X$  minutos, durante a chamada. Pretende-se um programa para determinar o saldo final do cliente após uma chamada com uma certa duração.

Dados de entrada (a pedir ao utilizador):

- sal – saldo inicial do cliente (p.ex. 25,00 €)
- dur – duração da chamada em minutos (p.ex. 6 minutos)
- pf – preço fixo da chamada (p.ex. 0,30 €)
- ppm – preço por minuto (p.ex. 0,10 €)
- per – período em minutos ao fim do qual o cliente recebe um bónus no seu saldo (p.ex. 5 minutos)
- bonus – bónus a somar ao saldo do cliente (p.ex. 0,05 €)

Dados de saída (a apresentar no ecrã):

- o saldo final do cliente (para os valores acima especificados dá 24,15 €)

a) Implemente uma função que devolve o saldo final do cliente ao fim de uma chamada, admitindo que não existem bónus.

b) Implemente uma função que devolve o saldo final do cliente ao fim de uma chamada, admitindo que existe bónus. Utilize a função anterior.

c) Escreva o programa que calcula o saldo final do cliente no final da chamada. Utilize as funções anteriores.

## 6. A Bolsa

Numa bolsa de valores é possível comprar e vender acções, onde existe um preço por acção. Cada participante na bolsa tem um conjunto de acções. Chama-se “carteira” ao número total de acções que um participante tem. Cada participante tem também uma certa quantia em dinheiro, que pode usar para comprar acções, ou que pode aumentar pela venda das suas acções. No entanto, só é possível comprar e vender acções em lotes (por exemplo, em lotes de 5 acções). Uma característica da bolsa é que a venda de um lote de acções afecta o próprio preço por acção (faz descer esse preço). Assim, o próximo lote já será vendido a um preço inferior ao do primeiro lote. Assuma que o preço por acção sofre sempre o mesmo decréscimo quando é vendido um lote (por exemplo, o preço desce sempre 1€ após a venda de um lote). Considere também que se o preço por acção descer abaixo de um certo valor mínimo, o participante deixa de vender. Pretende-se desenvolver um programa para determinar com quanto dinheiro ficaria um participante se vendesse as suas acções, isto é, se vendesse acções até não ter mais lotes para vender, ou até o preço por acção ter atingido o valor mínimo aceitável.

Dados de entrada (a pedir ao utilizador):

- dinheiro – dinheiro do participante (p.ex. 350,00 €)
- carteira – número de acções do participante (p.ex. 38)
- lote – número de acções num lote (p.ex. 5)
- ppa – preço por acção (p.ex. 8,50 €)
- min – valor mínimo de ppa que o participante aceita para vender as suas acções (p.ex. 6,00 €)
- dec – valor a subtrair ao ppa sempre que é vendido um lote (p.ex. 1,00 €)

Dados de saída (a apresentar no ecrã):

- dinheiro do participante após as vendas (para os valores acima especificados dá 462,50 €)

a) Implemente uma função que devolve o dinheiro com que fica o participante após a venda de um lote de acções.

b) Implemente uma função que devolve o dinheiro com que fica o participante após a venda das suas acções, até não ter mais lotes para vender ou até o preço por acção descer abaixo do mínimo. Utilize a função anterior.

c) Escreva o programa que calcula o dinheiro com que o participante fica após as vendas. Utilize as funções anteriores.