Aula 5

Resolução do trabalho autónomo e exercícios sobre vetores

Iniciativa Conjunta:

Com o apoio de:

Trabalho autónomo da aula 2

1. Obter o quadrado de um número.
2. Obter a diferença entre dois números.
3. Obter a média entre dois números.
4. Obter a área de um retângulo, dando os lados.
5. Dada uma temperatura em Celsius, obter a temperatura em Fahrenheit.
6. Dada uma temperatura em Celsius, obter a temperatura em kelvin.
7. Dado um comprimento em centímetros, obter o valor em polegadas.
8. Dado um preço, obter o IVA a 23% correspondente ao mesmo.
9. Dado um número com casas decimais (double), devolve o inteiro aproximado.
10. Obter o perímetro de um círculo, dado o seu diâmetro.
11. Obter a área de um prisma retangular, dando a sua largura, altura e comprimento.

Trabalho autónomo da aula 3

1. Criar uma função que dado o ano de nascimento devolve a idade.
2. Criar uma função que divide dois números. Caso a divisão seja por zero, a função deve devolver 0.
3. Criar uma função que devolve verdadeiro se o número dado for par.
4. Criar uma função que devolve verdadeiro se o número dado é positivo e falso caso contrário.
5. Criar uma função que devolva o conteúdo escrito na consola. Para este exercício deve utilizar o Scanner.
6. Criar uma função que devolve o número de rodas que o veículo possui, dado esse mesmo veículo. Assuma apenas os seguintes veículos: automóvel, moto, bicicleta, triciclo, camião, monociclo.
7. Criar uma função que indica a nota qualitativa dada uma nota quantitativa.
8. Criar uma função que receba um número inteiro e devolva uma string com o nome do mês correspondente.
9. 9. Criar uma função que peça ao utilizador o dia, o mês de nascimento e devolva uma
10. string com o signo correspondente.
11. Criar uma função que peça ao utilizador três números e os coloque por ordem crescente.

Trabalho autónomo da aula 4

1. Criar uma função que mostre a soma de todos os números no intervalo de 1 até 100 utilizando o While.
2. Criar uma função que mostre a soma de todos os números no intervalo de 1 até 100 utilizando o For.
3. Criar uma função que devolva o número de divisores de um número inteiro n. Esta função deverá iterar sobre os números naturais até n, contando os números que são divisores de n.
4. Criar uma função que devolva o somatório dos divisores próprios de um número inteiro n (o conjunto dos divisores exclui o próprio número). Esta função deverá iterar sobre os número naturais até n (exclusive), acumulando os números que são divisores de n.
5. Criar uma função que que recebe como argumento um número natural e devolve verdadeiro caso seja primo, ou falso caso contrário.
6. Criar uma função que permite saber se existe algum número primo num dado intervalo (aberto).

Exercícios sobre vetores

1. Construir um vetor de números naturais até um dado número *n*. Exemplo: naturals(5)->{1,2,3,4,5}
2. Construir um vetor de dígitos aleatórios (números de 0 a 9), dado o comprimento. Exemplo: randomDigits(5)->{8,2,9,1,2}
3. Construir um vetor capaz de armazenar 50 números inteiros. Em seguida faça o seu preenchimento automático com os números 101 a 150, ou seja na posição número 0 fica 101, na posição número 1 fica 102 e por aí adiante.
4. Copiar (replicar) um vetor de inteiros, tendo o novo vetor o mesmo tamanho do argumento. Exemplo: copy({1,2,3},6)->{1,2,3,0,0,0} copy({1,2,3,4,5,6},3)->{1,2,3}
5. Verificar se existe um determinado número num vetor. Exemplo: exists(5,{1,3,4,5})->true exists(1,{2,3})->false
6. Contar o número de ocorrências de um determinado caractere. Exemplo: count(a,{a,b,c,a})->2
7. Construir um sub-vetor de outro vetor, dados os índices do primeiro elemento e último a incluir. Exemplo subarry(2,4,{a,d,r,a,c,r,w})}->{r,a,c}

Exercícios sobre vetores

1. Obter a primeira metade um vetor *v*, incluindo um parâmetro booleano para permitir se o elemento do meio é para incluir (caso o comprimento do vetor seja ímpar). Se o comprimento for par, este parâmetro não terá efeito. Exemplo firstHalf({b,a,s,w,q}, true)->{b,a,s}
2. Construir um vetor juntando outros dois vetores (parte esquerda e parte direita). Exemplo: merge({1,2},{9,10})
3. Construir um vetor invertido com base noutro. Ou seja, o novo vetor será composto pelos elementos do vetor dado pela ordem inversa. Exemplo: invert({t,q,a)}->{a,q,t}
4. Construir um vetor com base noutro, de modo o dobro do tamanha e cada elemento duplicado. Exemplo: duplicateEveryElement({a,s,d})->{a,a,s,s,d,d}
5. Construir um vetor com base noutro, sendo a primeira metade uma cópia e a segunda metade os mesmos elementos para ordem inversa. Exemplo: duplicateInverted({3,2,1})->(3,2,1,1,2,3)
6. Construir um vetor com base noutro, representado um cópia sem o elemento do meio (caso o tamanho seja ímpar) Exemplo: copyWithoutMiddleElement({1,2,3,4,5})->{1,2,4,5}
7. Construir um vetor com n números da sequência Fibonacci. Exemplo: fibonacciSequence(7)->{0,1,1,2,3,5,8}