

Exercícios sobre Funções

Instruções: Para todas as questões abaixo, escreva um programa no Processing para resolvê-las.

1. Faça um programa capaz de “desenhar” uma linha. Para efeito de exibição, a linha será desenhada por uma sequência de pontos desenhados na tela. Seguem alguns detalhes sobre a linha a ser desenhada:
 - o desenho será feito por meio de chamada a uma função ou procedimento.
 - a função receberá os seguintes parâmetros:
 - a coordenada x do vértice no ponto mais alto e mais à esquerda da linha;
 - a coordenada y do vértice no ponto mais alto e mais à esquerda da linha;
 - a orientação da linha: o programa passará à função o caractere ‘h’, se for para desenhar linha horizontal, e o caractere ‘v’, se for para desenhar uma linha vertical.
 - o comprimento dessa linha. Entenderemos como comprimento a quantidade de pontos que a linha conterá.
 - **Exemplo:** a chamada à função `linha(200, 200, 'h', 20);` desenhará, a partir da posição 200, 200 na tela, uma sequência horizontal de 20 pontos.
2. Faça um programa que desenhe 4 retângulos de cores diferentes. Para o desenho dos retângulos, construa uma função (não use a função `rect()` do Processing). Defina os parâmetros que ela vai receber e utilize a função em seu programa.
Dica: Você pode usar a sua função de desenhar linhas para desenhar as bordas do seu retângulo.
3. Faça um programa que contenha um procedimento que receba um valor numérico representando um pagamento em Reais a ser feito. O procedimento deverá listar a forma de pagamento em espécie que utilize a menor quantidade de cédulas possível. Assuma que os valores manipulados por seu programa não conterão centavos.
4. Faça um programa que contenha e use um procedimento que exiba os n primeiros termos de uma série de Fibonacci.
OBS: uma série de Fibonacci é uma sequência de números tal que o número seguinte é a soma dos dois números anteriores a ele na série.
Exemplo: para n = 10, os 10 primeiros termos são 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.

Assim, para ter graça, seu programa deverá pedir que o valor de `n` seja superior a 2.

5. Faça um programa que contenha e use uma função `divide`, que recebe dois parâmetros:
- um dividendo
 - um divisor
- e a função retornará se a divisão é exata ou não.

6. Faça um programa que contenha e use uma função `distancia`. Ela receberá as coordenadas de dois pontos no espaço 2D e retornará o valor da distância Euclidiana entre esses dois pontos.

Dica: para calcular a distância Euclidiana entre os pontos `p1` e `p2` no espaço 2D, utilizamos a fórmula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

OBS: será necessário pesquisar as funções `sqrt()`, `sqr()` e `pow()` do Processing.

7. Observe o programa abaixo. Ele contém um laço `for`. Reescreva o programa usando um laço `while`, de modo que ele produza o mesmo resultado.

```
void setup(){
  int i;
  for(i = 0; i < 10; i++)
    println(dobro(i));
}

float dobro(int n){
  return n * 2;
}
```

8. Observe o programa abaixo. Ele contém um laço `while`. Reescreva o programa usando um laço `for`, de modo que ele produza o mesmo resultado.

```
void setup(){
  int i;
  i = 10;
  while(i >= 0){
    if(par(i))
      println(i);
    i--;
  }
}

boolean par(int n){
  if(n % 2 == 0)
    return true;
  else
    return false;
}
```

9. Faça um programa que contenha e use uma função que receba um valor entre 100 e 999 e retorne o “espelho” desse número. Por exemplo, se a função receber o número 135, ele retornará como resultado o número 531.

10. Crie uma função que calcule a soma de dois números. Crie outra função que calcule a subtração de dois números. Crie uma outra função que calcule a multiplicação de dois números. Por fim, crie uma última função que calcule a divisão de dois números. Crie um programa que realiza operações matemáticas utilizando as quatro funções, exibindo os resultados. A escolha da operação a ser efetuada, bem como os dois números a serem fornecidos, podem ser obtidos por geração de números aleatórios.

trilha sonora sugerida:

https://www.youtube.com/watch?v=vE4VIA_9OrI&index=0&list=RDvE4VIA_9OrI

Bom trabalho!!