Documentação dos Códigos
<047>

MUITO FÁCIL:

1.

No início do Main, é declarada a variável “r” em float para que não ocorram erros ao calcular o valor em graus, já que terá, provavelmente, números após a vírgula. O programa recebe o valor de “r” e chama a função radianos_graus, que calcula por meio da fórmula e transforma o valor em graus, retornando o valor e printando na tela com 1 casa decimal de arredondamento.

2.

O programa recebe o valor de uma variável anteriormente declarada em inteiro, para determinar horas, e já chama a função para converter o valor para os outros parâmetros de tempo pedidos. A função “converter” cria uma lista vazia de 5 elementos para ser preenchida pelos valores convertidos a partir da hora. Um laço de repetição é criado apenas para separar quais valores serão colocados na lista, de forma organizada, um por um, sendo convertidos a partir das fórmulas. No fim do laço, a lista já é gradativamente retornada à tela do usuário.

FÁCIL:

1.

São inicialmente declaradas as variáveis que vão receber, respectivamente, número de repetições e a palavra a ser repetida. O programa recebe as duas variáveis e chama a função “repeti”. Esta, cria um laço de repetição em “i”, durando até a quantidade de “n” (vezes a ser repetido), printando uma lista de “n” vezes a palavra digitada., com um if-else apenas para organizar os componentes gráficos da lista.

2.

O programa toma o valor de um número e chama a função “conta_uns”, que declara a variável “q” (variável de suporte que conta a quantidade de “1’s”), a variável t (para medir o tamanho da string gerada pela função “itoa”, que transforma um número em binário, aplicando a sequência em forma de string, em um char declarado), “i” (suporte para o laço de repetição) e “bin” (variável char declarada como string de 30 caracteres máximos, posteriormente recebendo a sequência binária do “itoa”). A função gera um laço de repetição simples que recebe cada caractere da string “bin”, individualmente, e conta em “q”, que será retornado como quantidade de “1’s” na sequência binária.

Médio:

1.

No main, são declaradas as variáveis que receberão as coordenadas dos dois pontos, com um aviso na tela de execução de como formatar a entrada. Todas as variáveis foram declaradas em float. A seguir, é chamada a função “distancia” dentro do printf, para retornar o valor com duas casas decimais. Esta função, retornando em double, já que utiliza “pow” e “sqrt” (por isto todas as variáveis estão em float, para evitar problemas na equação), aplica a fórmula de distância entre dois pontos, usando as coordenadas de cada um, para mostrar o comprimento do segmento de reta entre eles.

2.

O início do programa apenas declara uma variável do tipo inteiro, para receber seu valor e chamar a função “crossbots”. A função recebe o valor digitado e aplica o operador “%”, que apresenta o resto de uma divisão. Com isso, em um laço de repetição criado apenas para organizar os possíveis resultados e declarações de string necessárias para cada situação, são aplicados “if’s”, para descobrir, respectivamente, se o número é múltiplo de 15 (se for, já declara a string com “Crossbots”, retorna a mesma e fecha o laço, para que os outros “if’s” não sejam computados, já que se for múltiplo de 15, é de 5 e de 3 ao mesmo tempo), múltiplo de 3 ou múltiplo de 5, retornando o texto correto. Caso não seja múltiplo nem de 3, nem de 5, o próprio número é retornado.