

Cada questão abaixo vale 2,0 pontos e você tem um tempo de 80 minutos de prova.

Assunto da prova: comandos de atribuição, input, print, condicional (if) e funções.

1. Para entrar na Montanha Russa do Banco de Gringotes, você precisa ter **entre** 16 e 80 anos de idade e tem que ter pelo menos 1,60m de altura. Faça um **programa** que pede para o usuário informar sua idade e altura. Depois disso, o programa deve **informar** ao usuário se ele pode ou não entrar na Montanha Russa do Banco de Gringotes.
2. Você está na equipe de desenvolvimento de um remake do jogo "pedra, papel e tesoura". Pediram para você implementar a função **winner** (vide abaixo) que verifica qual jogador venceu uma partida. Nas regras do "pedra, papel e tesoura", dois jogadores escolhem um desses três símbolos. Pelas regras, pedra ("stone") vence a tesoura ("scissor"), tesoura vence o papel ("paper") e papel vence a pedra. O jogo também pode dar empate, quando ambos escolhem o mesmo símbolo. Implemente apenas a função **winner**, a seguir, que recebe a jogada de cada um dos dois jogadores, que é uma das cadeias de caracteres (string): "stone", "scissor" ou "paper". A função **winner** deve retornar como resultado uma das três cadeias (strings) a seguir: "player1" quando o jogador 1 vencer, "player2" quando o jogador 2 vencer, ou "draw" quando ocorrer um empate.
`def winner(play1, play2):`
3. Um aluno precisa ter média setenta (70) nas três primeiras avaliações para ser aprovado em uma disciplina. Quando isso não ocorre, ele precisa fazer uma quarta prova, que é a reposição. A reposição substitui a menor das três primeiras avaliações (caso ela seja maior que a menor nota dessas três). Faça a função **notaReposicao** que recebe três notas em uma disciplina (inteiros variando de 0 a 100) e retorna a nota necessária para o aluno passar na reposição. **Importante:** se o aluno já se encontra aprovado com essas três primeiras avaliações, retorne zero (0) como resultado.
`def notaReposicao (nota1, nota2, nota3):`
4. No War digital, quando o exército faz um ataque, ele lança três dados. O território atacado se defende com três dados. O maior resultado do ataque deve ser comparado com o maior resultado da defesa. Da mesma forma, o valor do meio e o menor valor devem ser comparados. O empate nos dados favorece a defesa. Assim sendo, um ataque com {4, 6, 5} e uma defesa com {5, 4, 4} vão resultar em dois acertos do ataque (o maior ataque 6 é maior que a maior defesa 5; e o meio de ataque 5 é maior que o meio da defesa 4, porém o menor ataque 4 é igual a menor defesa 4 não resultando em um terceiro acerto). Crie uma função ataque que recebe o resultado dos três dados de ataque e três dados de defesa e retorna a quantidade de acertos do ataque.
`def ataque(atq1, atq2, atq3, def1, def2, def3):`
5. Há quatro tipos de animais na terra de Wakanda: elefante, humano, papagaio e cachorro. Um robô escaneia formas de vida e transforma cada resultado disso em um número inteiro. Um exemplo disso é que, ao ver uma bananeira, ele identifica o inteiro 457. Os inteiros que representam os quatro tipos de animais são, respectivamente, 123, 234, 345 e 456. A função **animaisScanner** informa os inteiros que representam as últimas cinco formas de vida escaneadas. Implemente essa função, retornando a quantidade de animais escaneados entre as cinco últimas formas de vida.
`def animaisScanner(vida1, vida2, vida3, vida4, vida5) :`