

2ª lista de Lógica de Programação

Estruturas de repetição

1. Use a estrutura de repetição **enquanto condição faz** para mostrar uma contagem de números pares na tela. Essa contagem deve iniciar em 0 e terminar com um número previamente informado pelo teclado.
2. Resolva o exercício 1 novamente, porém desta vez usando uma estrutura de repetição **para variável de início ate fim passo incremento**.
3. Modifique o algoritmo da questão 2 para que o usuário possa escolher, usando o teclado, se quer mostrar números pares ou ímpares.
4. Em um jogo de adivinhação, o jogador deve descobrir um número entre 1 e 10 usando até três tentativas. A cada tentativa o jogador é informado se o número foi descoberto, ou se ele é maior ou menor do que a tentativa. Se após três tentativas o jogador não descobrir o número, então ele perde o jogo. Faça um algoritmo que implemente esse jogo.
5. Faça um algoritmo para calcular a média de 4 notas de 0 a 10, porém desprezando a menor nota.
6. Faça um algoritmo que leia 5 números do teclado, e faça uma média dos três números intermediários (quer dizer, despreze o maior e o menor deles).
7. Faça um algoritmo que mostre na tela todos os números entre 1 e 100 que sejam múltiplos de 3 ou 5.
8. Faça um algoritmo que leia o nome e idade de 5 pessoas, e mostre o nome da pessoa mais velha.
9. Faça um algoritmo que descubra quando a função $f(x) = 2 \cdot x^3 + n^2 - n$ se torna maior ou igual a 100.
10. Geradores de números pseudo-aleatórios (PRNG) são de grande utilidade e importância para muitos algoritmos. Por exemplo, qualquer algoritmo que precise sortear um número pode usar um gerador desses (como o exercício sobre o jogo de 21, introduzido na questão 15 desta lista). Um gerador muito usado devido à sua simplicidade é o Gerador Linear Congruente (LCG), dado pela expressão:

$$X_{n+1} = (aX_n + c) \bmod N$$

... sendo X_{n+1} o próximo número da sequência, X_n o número anterior, e a , c e N parâmetros do gerador.

O LCG gera uma sequência de números com um determinado período, após o que a sequência se repete. O período máximo é limitado pelo parâmetro N , porém dependendo dos parâmetros a e c esse período pode ser ainda menor. Veja os exemplos abaixo de algumas sequências:

$a=3, c=1, N=7$: 0, 1, 4, 6, 5, 2

$a=3, c=1, N=11$: 0, 1, 4, 2, 7

$a=7, c=3, N=13$: 0, 3, 11, 2, 4, 5, 12, 9, 1, 10, 8, 7

Escreva um algoritmo que mostre todos os números de uma sequência gerada por um LCG. Repare que os dados de entrada são os parâmetros do gerador: a , c e N .

11. DESAFIO: a respeito da questão anterior, foi dito que o período da sequência é limitado por N , porém depende também dos parâmetros a e c . Assim, faça um algoritmo que, dado N , descubra valores para a e c que maximizem o período. Algumas observações:

. o período máximo é N

. $0 < a < N$

. $0 < c < N$

12. Modifique o algoritmo da calculadora (ver a 1ª lista de exercícios de lógica de programação) para que fique resolvendo contas até que o usuário informe que não deseja mais continuar. Quer dizer, a calculadora lê os dois números e a operação, calcula e mostra o resultado, e em seguida pergunta ao usuário se ele deseja fazer uma nova conta. Caso o usuário deseje continuar, a calculadora volta a pedir os dois números e a operação. Caso contrário, ela termina a execução.
13. Refaça a calculadora para que ela opere como uma calculadora real: o usuário digita um número, depois a operação, outro número, outra operação, outro número, e assim continua até que ele digite enfim o sinal "=", quando então a calculadora mostra o resultado. Exemplos:

$$1 + 3 - 2 / 2 = 1$$

$$3 * 4 - 2 / 5 + 4 = 6$$

14. Faça um algoritmo que mostre um triângulo composto por caracteres "#". O tamanho da base deve ser informado pelo teclado, e deve ser no mínimo 5 e no máximo 15. Ex:

Com base = 5:

```
#
###
#####
```

Com base 9:

```
#
###
#####
#####
#####
```

Obs: repare que o tamanho da base precisa ser ímpar para que o triângulo fique simétrico.

15. DESAFIO: Faça um algoritmo para o jogo de 21 (Blackjack). Nesse jogo de cartas, o objetivo é conseguir um resultado mais perto de 21 sem ultrapassar ("quebrar"). A banca (representada pelo algoritmo) joga contra o jogador, e vence quem chegar mais perto de 21, ou conseguir exatamente 21 na soma das cartas. Caso ocorra empate a banca vence. Quem ultrapassar 21 na soma das cartas automaticamente perde a jogada. O jogador inicia o jogo com um crédito de R\$ 100, e cada jogada implica uma aposta de R\$ 10. Se as cartas do jogador forem um Ás e um 10, a banca paga 1,5 vezes a aposta. Qualquer outra combinação de cartas vencedora a banca paga 1 vez a aposta.

Valores das cartas:

. Cartas de Rei, Dama, Valeta e 10 valem cada uma 10.

. Cartas de Ás valem 1 ou 11, como o jogador desejar

. Todas as outras cartas são contadas de acordo com seus valores de face

16. DESAFIO: faça um algoritmo que mostre o calendário de um determinado mês. Os dados de entrada são o mês cujos dados devem ser mostrados, e o dia da semana em que esse mês inicia. Assuma que esse mês está em um ano que não é bissexto. O resultado deve ser como mostrado abaixo:

Ex: janeiro, começando na 5a feira:

Dom	Seg	Ter	Qua	Qui	Sex	Sab
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Ex: setembro, iniciando na 3a feira:

Dom	Seg	Ter	Qua	Qui	Sex	Sab
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

17. MEGA-DESAFIO: Aproveitando o exercício da questão 16, faça um algoritmo que mostre o calendário de um determinado mês de um ano, sendo ambos informados pelo teclado. Ex: janeiro de 2009, março de 2000, dezembro de 1977. Seu algoritmo deve conseguir mostrar calendários de um mês de qualquer ano depois de 1970 (inclusive). *Dicas:*

i) use 1/1/1970 como data de referência, identificando em que dia da semana iniciou aquele ano.

ii) descubra em que dia da semana inicia o ano em que fica o mês cujo calendário deve ser mostrado. Para isto você deve calcular quantos dias se passaram desde a data de referência até o primeiro dia do ano em questão (não esqueça de considerar os anos bissextos).