

1ª LISTAInstruções Importantes

- Esta lista é para ser feita INDIVIDUALMENTE.
 - Escolha LIVREMENTE quais questões quer resolver e indique-as logo no início de sua folha-resposta. O somatório de pontos não pode exceder 10. Caso exceda, só serão consideradas, em ordem de resolução, as questões cujo somatório de pontos seja menor ou igual a 10.
 - O trabalho deverá ser entregue OBRIGATORIAMENTE em pdf. Cada questão deverá ter como resposta o devido código fonte e um print da tela de execução demonstrando que seu programa executou corretamente (use o Code::Blocks ou o moveIC).
 - O arquivo pdf deverá ser enviado para m.felix@suite.iff.edu.br
-

1. (1 Pt) Escreva em C uma função “int perfeito(int n)” que retorne 1 caso x seja um número perfeito e 0, caso contrário. Em seguida, escreva um programa que liste os números perfeitos, em ordem crescente, até um valor limite dado no início pelo usuário.

Definição: um número é dito “perfeito” se é igual à soma de seus divisores próprios. Os divisores próprios de um número são todos os seus divisores, exceto ele mesmo. Exemplo: 28 é perfeito porque $28 = 1 + 2 + 4 + 7 + 14$ (Note que 1, 2, 4, 7 e 14 são os divisores próprios de 28).

2. (2 Pts) Escreva um programa em C para fatorar um número natural dado pelo usuário.

A saída deverá ser dada no seguinte formato, supondo que a entrada tenha sido 87120:

$$87120 = 2^4 \times 3^2 \times 5 \times 11^2$$

(Obs.: usamos o símbolo “^” para representar a operação de potenciação.)

3. (3 Pts) A Conjectura de Goldbach afirma que, para todo número natural N, par e maior que 3, sempre existem dois números primos P e Q tais que $N = P + Q$. Sabendo disso, escreva um programa em C para, dado um tal N, verificar que a conjectura é verdadeira para valores até esse N.

4. (2 Pts) Dados dois polinômios P(n) e Q(n), escreva um programa C para calcular a soma e o produto de P e Q. Note que um polinômio na forma canônica $P(n) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + \dots + a_0 \cdot x^0$, pode ser representado por um vetor com n+1 posições, cada uma para armazenar um coeficiente a_n . Seu programa deverá usar 4 vetores, para representar P, Q, P+Q e P×Q.

5. (3 Pts) Pesquise o problema conhecido por “Torre de Hanoi”. Implemente em C um programa recursivo para resolver uma torre de Hanói com n discos, onde n é dado pelo usuário.

6. (1 Pt) Escreva um programa iterativo para calcular o n-ésimo termo de uma sequência de Fibonacci onde os dois primeiros termos da sequência são dados pelo usuário. Observe a solução recursiva apresentada num dos nossos vídeos e a definição da sequência de Fibonacci.

7. (1 Pt) Escreva em C um programa para gerar e carregar um vetor de 10000 posições inteiras como valores inteiros no intervalo de 0 a 10000. Seu programa deverá encontrar o maior e o menor elemento armazenado no vetor, bem como o número de valores pares e de valores ímpares, informando esses resultados ao fim da execução.

8. (3 Pts) Implemente em C um programa para ordenar os elementos gerados e armazenados no vetor da questão 7. O algoritmo utilizado deverá ser mais eficiente do que o BubbleSort.

9. (1 Pt) Escreva em C um programa para retornar o maior número presente na sequência de Collatz iniciada com um valor $n_0 > 4$, dado pelo usuário.

Definição: a sequência de Collatz $\langle n_0, n_1, n_2, \dots, 4, 2, 1 \rangle$ é definida da seguinte forma: sendo n_0 o valor inicial dado pelo usuário, $n_0 > 4$, teremos que $n_{i+1} = n_i/2$, caso n_i seja par e $n_{i+1} = 1 + 3*n_i$, caso n_i seja ímpar. Note que essa sequência sempre converge para 1, qualquer que seja o valor de n_0 dado pelo usuário.