Introdução à Programação - turma 1 (diurno)

Lista 03 - Régis S. Santos

2012

Exercício E- 14 Dizemos que um inteiro positivo n é *perfeito* se for igual à soma de seus divisores positivos diferentes de n. Exemplo: 6 é perfeito, pois $1 + 2 + 3 = 6$. Faça um programa em C++ que verifica se um dado número inteiro positivo é perfeito.

Solução:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, perfeito = 0;
    cout << "Digite um numero: " << endl;
    cin >> n;
    for (i = 1; i < n; i++)
    {
        if (n % i == 0)
            perfeito += i;
    }
    if (perfeito == n)
        cout << "'E perfeito." << endl;
    else
        cout << "Nao 'e perfeito." << endl;
    return 0;
}
```

□

Exercício E- 15 Um matemático italiano da idade média conseguiu modelar o ritmo do crescimento da população de coelhos através de uma sequência de números naturais que passou a ser conhecida como *sequência de Fibonacci*. O n-ésimo número da sequência de Fibonacci F_n é dado pela seguinte fórmula de recorrência:

$$\begin{cases} F_1 = 1 \\ F_2 = 2 \\ F_i = F_{i-1} + F_{i-2}, \quad \text{para } i \geqslant 3. \end{cases}$$

Faça um programa em C++ que dado n natural calcula F_n.

Solução:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, anterior, atual, temp;
    anterior = atual = 1;
    cout << "Digite o n-esimo termo da sequencia de
        Fibonacci: ";
    cin >> n;
    for (i = 3; i <= n+1; i++)
    {
        temp = atual;
        atual += anterior;
        anterior = temp;
    }
    cout << atual << endl;
    return 0;
}
```

□

Exercício E- 16 * Considere o conjunto $H = H_1 \cup H_2$ de pontos reais, onde

$$H_1 = \{(x, y) | x \leqslant 0, y \leqslant 0, y + x^2 + 2x - 3 \leqslant 0\} \text{ e}$$

$$H_2 = \{(x, y) | x \geqslant 0, y + x^2 - 2x - 3 \leqslant 0\}.$$

Faça um programa que lê um inteiro positivo n e uma sequência de n pontos reais (x, y) e verifica se cada ponto pertence ou não ao conjunto H. O programa deve também contar o número de pontos da sequência que pertencem a H.

Solução:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, pertence = 0;
    float x, y;
    cout << "Digite um numero: ";
    cin >> n;
    for (i = 0; i < n; i++)
    {
        cin >> x >> y;
        if ((x <= 0 && y <= 0 && y + x*x + 2*x - 3 <= 0) ||
            (x >= 0 && y + x*x - 2*x - 3 <= 0))
        {
            cout << "Pertence a H." << endl;
            pertence ++;
        }
        else
            cout << "Nao pertence a H." << endl;
    }
    cout << pertence << " pontos pertencem a H." <<endl;
    return 0;
}
```

□

Exercício E- 17 * Dada uma frase terminada por '.', faça um programa em C++ que determina quantas letras e quantas palavras aparecem no texto. Por exemplo, no texto "O voo GOL547 saiu com 10 passageiros." há 25 letras e 7 palavras.

Solução:

```
#include <iostream>
using namespace std;
int main()
{
    int palavras = 1, letras = 0;
    char c;
    cin >> noskipws >> c;
    while (c != '.')
    {
        if (c == ' ')
            palavras ++;
        if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')
            )
            letras ++;
        cin >> noskipws >> c;
    }
    cout << palavras << " palavras." << endl;
    cout << letras << " letras." << endl;
    return 0;
}
```

□

Exercício E- 18 Dada uma frase terminada por '.', faça um programa em C++ que determina quantos caracteres são divisíveis por 2 ou por 3. Lembre que cada caracter possui um número decimal a ele associado.

Solução:

```
#include <iostream>
using namespace std;
int main()
{
    int caracter = 0;
    char c;
    while (c != '.')
    {
        cin >> noskipws >> c;
        if (c % 2 == 0 || c % 3 == 0)
            caracter ++;
    }
    cout << caracter << endl;
    return 0;
}
```

□

Exercício E- 19 Dado um número real $0 < \varepsilon < 1$, faça um programa em C++ que calcula uma aproximação do número de Euler (uma variante deste nome é número de Néper) através da série infinita:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots + \frac{1}{k!} + \ldots$$

Inclua na aproximação todos os termos cujo valor absoluto é maior ou igual ao valor de ε.

Solução:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, j;
    float fat, soma = 1.0;
    cout << "Digite um numero: " << endl;
    cin >> n;
    for (i = 1; i <= n; i++)
    {
        fat = 1;
        for (j = 1; j <= i; j++)
        {
            fat *= j;
        }
        soma += 1.0 / fat;
    }
    cout << soma << endl;
    return 0;
}
```

□