Introdução à Programação - turma 1 (diurno)

Lista 02 - Régis S. Santos

2012

Exercício E- 8 * Escolha 10 problemas vistos em sala de aula e faça um programa em C++ para cada um deles. Você pode copiar o programa feito no quadro.

Solução:

A seguir os exercícios solicitados com enunciado e resposta, observe o nome de cada arquivo logo no início de cada enunciado.

(1) **E08prob01.cpp**

Dada uma sequência de números inteiros diferentes de zero, terminada por um zero, calcular a sua soma. Por exemplo, para a sequência 12 7 4 -6 8 0 o seu programa deve escrever o número 35.

```
#include <iostream>
using namespace std;
int main()
{
    int numero, soma;
    soma = 0;
    cin >> numero;
    cout << numero << endl;
    while (numero != 0)
    {
        soma = soma + numero;
        cin >> numero;
        cout << numero << endl;
    }
    cout << "soma = " << soma << endl;
    return 0;
}
```

(2) **E08prob02.cpp**

Dados os números inteiros n e k com $k \geqslant 0$, determinar n^k. Por exemplo, dados os números 3 e 4 o programa deve escrever o número 81.

```
#include <iostream>
using namespace std;
int main()
{
    int n, k, resultado, potencia;
    cout << "Digite um numero:";
    cin >> n;
```

```
    cout << "Digite outro numero:";
    cin >> k;
    resultado = 1;
    potencia = 0;
    while (potencia < k)
    {
        resultado = resultado * n;
        potencia = potencia + 1;
    }
    cout << resultado << endl;
    return 0;
}
```

(3) **E08prob03.cpp**

Dado um número inteiro $n \geqslant 0$, calcular $n!$.

$$n! = \begin{cases} 1 & \text{, se } n = 0 \\ n & \text{, se } n = 1 \\ n(n-1)! & \text{, se } n > 1 \end{cases}$$

```
#include <iostream>
using namespace std;
int main()
{
    int n, resultado, i;
    cout << "Digite um numero:";
    cin >> n;
    resultado = 1;
    i = n;
    while (i >= 1)
    {
        resultado = resultado*i;
        i = i - 1;
    }
    cout << resultado << endl;
    return 0;
}
```

(4) **E08prob05.cpp**

Dados um número inteiro $n \geqslant 0$, e uma sequência com n inteiros, determinar a soma dos inteiros positivos e a soma dos inteiros negativos da sequência. Por exemplo, para a sequência 6 -2 7 0 -5 8 4 o programa deve escrever o número 19 e -7.

```
#include <iostream>;
using namespace std;
int main()
{
    int n, i, numero, pos, neg;
    i = 0;
    pos = 0;
    neg = 0;
    cout << "Digite o comprimento da sequencia: ";
    cin >> n;
```

```
    while (i < n)
    {
        cout << "Digite o proximo numero: ";
        cin >> numero;
        if (numero > 0)
        {
            pos = pos + numero;
        }
        else
        {
            neg = neg + numero;
        }
        i = i + 1;
    }
    cout << "A soma dos inteiros positivos eh: " << pos <<
       endl;
    cout << "A soma dos inteiros negativos eh: " << neg <<
       endl;
    return 0;
}
```

(5) **E08prob06.cpp**

Dados um número inteiro $n \geqslant 0$, e uma sequência com n inteiros, determinar quantos números da sequência são pares e quantos são ímpares. Por exemplo, para a sequência 6 -2 7 0 -5 8 4 o programa deve escrever o número 4 para o número de pares 2 para o número de ímpares.

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, numero, par, impar;
    i = 0;
    par = 0;
    impar = 0;
    cout << "Digite o comprimento da sequencia: ";
    cin >> n;
    while (i < n)
    {
        cout << "Digite o proximo numero: ";
        cin >> numero;
        if (numero % 2 == 0)
        {
            par = par + 1;
        }
        else
        {
            impar = impar + 1;
        }
        i = i + 1;
    }
    cout << "A sequencia possui " << par << " inteiros pares
        e " << impar << " inteiros impares." << endl;
    return 0;
}
```

(6) **E08probextra01.cpp**

Dados um número inteiro $n \geqslant 0$, e um digito d ($0 <= d <= 9$) determinar quantas vezes d ocorre em n. Por exemplo, para $n = 63453$ e $d = 3$ o programa deve imprimir 2.

```
#include <iostream>
using namespace std;
int main()
{
    int n, d, resto;
    resto = 0;
    cout << "Digite um numero: ";
    cin >> n;
    cout << "Digite um digito: ";
    cin >> d;
    while (n > 0)
    {
        if (n % 10 == d)
        {
            resto = resto + 1;
        }
        n = n / 10;
    }
    cout << "O numero " << d << " aparece "<< resto << "
       vezes."<< endl;
    return 0;
}
```

(7) **E08prob08.cpp**

Dados um número inteiro $n > 0$ e as notas de n alunos, determinar quantos alunos ficaram de recuperação. Um aluno está de recuperação se sua nota estiver entre 30 e 50 (nota máxima 100).

```
#include <iostream>
using namespace std;
int main(){
    int n, i, nota, rec;
    cout << "Digite o numero de alunos: ";
    cin >> n;
    rec = 0;
    for (i = 0; i < n; i = i + 1)
    {
        cout << "Digite as notas: ";
        cin >> nota;
        if (nota >= 30)
        {
            if (nota <= 50)
            {
                rec = rec + 1;
            }
        }
    }
    cout << rec << " alunos estao de recuperacao." << endl;
    return 0;
}
```

(8) **E08prob11.cpp**

Dado um número inteiro $n > 0$, verificar se n é primo.

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, primo = 1;
    cout << "Digite um numero: ";
    cin >> n;
    if (n == 1)
    {
        primo = 0;
    }
    for (i = 2; i < n; i++)
    {
        if (n % i == 0)
        {
            primo = 0;
        }
    }
    if (primo == 1)
    {
        cout << n << " 'e primo." << endl;
    }
    else
    {
        cout << n << " nao 'e primo." << endl;
    }
    return 0;
}
```

(9) **E08prob14.cpp**

Dados um inteiro $n > 0$, e uma sequência com n inteiros, verificar se a sequência é uma progressão aritmética.

```
#include <iostream>
#define TRUE 1
#define FALSE 0

using namespace std;
int main()
{
    int n, i, anterior, atual, razao = 0, res = TRUE;
    cout << "Digite o comprimento da sequencia: ";
    cin >> n;
    cin >> anterior;
    if (n > 1)
    {
        cin >> atual;
        razao = atual - anterior;
    }
    for (i = 2; i < n; i++)
    {
        anterior = atual;
```

```
        cin >> atual;
        if (atual - anterior != razao)
            res = FALSE;
    }
    if (res) /*ou res = TRUE*/
        cout << "'E uma P.A." << endl;
    else
        cout << "Nao 'e uma P.A." << endl;
    return 0;
}
```

(10) **E08prob15.cpp**

Sabe-se que cada número da forma n^3 é igual a soma de n ímpares consecutivos. Exemplos:

$$1^3 = 1$$
$$2^3 = 3 + 5$$
$$3^3 = 7 + 9 + 11$$
$$4^3 = 13 + 15 + 17 + 19$$

Dado um inteiro $m > 0$, determinar os ímpares consecutivos cuja soma é igual a n^3, para n assumindo valores de 1 a n.

```
#include <iostream>
using namespace std;
int main()
{
    int m, n, i, candidato, termo;
    cout << "Digite um numero: ";
    cin >> m;
    for (n = 1; n <= m; n++)
    {
        candidato = n*n - n + 1;
        cout << n << "^3 = ";
        termo = candidato;
        for (i = 0; i < n - 1; i++)
        {
            cout << termo << " + ";
            termo += 2;
        }
        cout << termo << endl;
    }
    return 0;
}
```

□

Exercício E- 9 Dizemos que um número natural n com pelo menos dois algarismos é *palíndrome* se o primeiro algarismo de n é igual ao seu último algarismo, o segundo algarismo de n é igual ao penúltimo algarismo, e assim por diante. Exemplos: 567765 e 32423 são palíndromes e 567675 não é palíndrome.

Dado um inteiro $n, n \geqslant 10$, verificar se n é palíndrome.

Solução:

```
#include<iostream>
using namespace std;
int main()
{
    int n, resto, soma = 0, temp;
    cout << "Digite um numero: ";
    cin >> n;
    temp  = n;
    while(n != 0)
    {
         resto = n % 10;
         n = n / 10;
         soma = soma * 10 + resto;
    }
    if(temp == soma)
         cout << "'e palindrome" << endl;
    else
         cout << "nao 'e palindrome" << endl;
    return 0;
}
```

□

Exercício E- 10 Dado um inteiro positivo n, determine o valor dado pela seguinte equação:

$$F_n = \begin{cases} \sum_{k=1}^{n} \frac{1}{k^2} & \text{, se } n \text{ é ímpar} \\ \sum_{k=1}^{n/2} \frac{1}{2^k} & \text{, se } n \text{ é par} \end{cases}$$

Solução:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, j, pot = 1;
    float soma = 0.0;
    cout << "Digite um numero: ";
    cin >> n;
    /* verifica se n 'e par */
    if (n % 2 == 0)
    {
```

```
12          /* \sum 1/k^2 */
13          for (i = 1; i <= n; i++)
14          {
15              pot = i * i;
16              soma += 1.0 / pot;
17          }
18      }
19      else
20      {
21          /* \sum 1/2^k */
22          for (i = 1; i <= n / 2; i++)
23          {
24              pot = 1;
25              for (j = 1; j <= i; j++)
26                  pot *= 2;
27              soma += 1.0 / pot;
28          }
29      }
30      cout << soma << endl;
31      return 0;
32  }
```

□

Exercício E- 11 Dizemos que um número i é congruente a j módulo m se $i\%m = j\%m$. Exemplo: 35 é congruente a 39 módulo 4, pois $35\%4 = 3 = 39\%4$.

Dados inteiros positivos n, j, e m, imprimir os n primeiros naturais congruentes a j módulo m.

Solução:

```
1   #include <iostream>
2   using namespace std;
3   int main()
4   {
5       int m, n, i, j;
6       cout << "Digite tres numeros: ";
7       cin >> n >> j >> m;
8       for (i = 0; i <= n; i++)
9           if (i % m == j % m)
10              cout << i << " ";
11      cout << endl;
12      return 0;
13  }
```

□

Exercício E- 12 Dado um inteiro n, calcular o valor da seguinte soma:

$$\frac{1}{n} + \frac{2}{n-1} + \frac{3}{n-2} + \ldots + \frac{n}{1}$$

Solução:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, k;
    float sn = 0.0;
    cout << "Digite um numero: ";
    cin >> n;
    k = n;
    for (i = 1; i <= n; i++)
    {
        sn = sn + (float)i / k;
        k--;
    }
    cout << sn << endl;
    return 0;
}
```

□

Exercício E- 13 * Faça um programa que calcula a soma

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \ldots + \frac{1}{9999} - \frac{1}{10000}$$

pelas seguintes maneiras:

a) adição dos termos da direita para a esquerda;

b) adição dos termos da esquerda para a direita;

c) adição separada dos termos positivos e dos termos negativos da esquerda para a direita;

d) adição separada dos termos positivos e dos termos negativos da direita para a esquerda.

Compare e discuta os resultados obtidos no computador.

Solução:
E13p1.cpp

```
#include <iostream>
using namespace std;
int main()
{
    /* d = denominador */
    int i, d;
    float soma = 0.0;
```

```
    for (i = 10000; i > 0; i--)
    {
        /* verifica se i 'e par. */
        if (i % 2 == 0)
            d = -i;
        else
            d = i;
        soma += 1.0 / d;
    }
    cout << soma << endl;
    return 0;
}
```

E13p2.cpp

```
#include <iostream>
using namespace std;
int main()
{
    /* d = denominador */
    int i, d;
    float soma = 0.0;
    for (i = 1; i <= 10000; i++)
    {
        /* verifica se i 'e par. */
        if (i % 2 == 0)
            d = -i;
        else
            d = i;
        soma += 1.0 / d;
    }
    cout << soma << endl;
    return 0;
}
```

E13p3.cpp

```
#include <iostream>
using namespace std;
int main()
{
    int i;
    float somaPos = 0.0, somaNeg = 0.0;
    for (i = 1; i <= 10000; i++)
    {
        /* verifica se i 'e par. */
        if (i % 2 == 0)
            somaNeg += 1.0 / -i;
        else
            somaPos += 1.0 / i;
    }
    cout << "soma positivos: " << somaPos << endl;
    cout << "soma negativos: " << somaNeg << endl;
    cout << "soma total: " << somaPos + somaNeg << endl;
    return 0;
}
```

E13p4.cpp

```cpp
#include <iostream>
using namespace std;
int main()
{
    int i;
    float somaPos = 0.0, somaNeg = 0.0;
    for (i = 10000; i > 0; i--)
    {
        /* verifica se i 'e par. */
        if (i % 2 == 0)
            somaNeg += 1.0 / -i;
        else
            somaPos += 1.0 / i;
    }
    cout << "soma positivos: " << somaPos << endl;
    cout << "soma negativos: " << somaNeg << endl;
    cout << "soma total: " << somaPos + somaNeg << endl;
    return 0;
}
```

Observe os resultados obtidos:
E13p1: 0.693097
E13p2: 0.693091

E13p3:
soma positivos: 5.24036
soma negativos: -4.54726
soma total: 0.693102

E13p4:
soma positivos: 5.24035
soma negativos: -4.54725
soma total: 0.693098
Os resultados sofrem alguma alteração depois de algumas casas decimais.

□