

**MAC2166 – Introdução à Computação**  
**ESCOLA POLITÉCNICA**  
Primeira Prova – 07 de abril de 2015

Nome: \_\_\_\_\_

Assinatura: \_\_\_\_\_

Nº USP: \_\_\_\_\_ Turma: \_\_\_\_\_

Professor: \_\_\_\_\_

**Instruções:**

1. Não destaque as folhas deste caderno.
2. A prova contém 3 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
3. Qualquer questão pode ser resolvida em qualquer página. Se a questão não está na página correspondente ao enunciado basta indicar isto na página e escrever **QUESTÃO X** em letras **ENORMES** antes da solução.
4. A prova pode ser feita a lápis. Cuidado com a legibilidade e, principalmente, com a **TABULAÇÃO**.
5. Não é necessário apagar rascunhos no caderno de questões.
6. Não é permitido o uso de folhas avulsas para rascunho.
7. Não é permitido o uso de equipamentos eletrônicos.
8. Não é permitido a consulta a livros, apontamentos ou colegas.

**DURAÇÃO DA PROVA: 2 horas**

Questão	Valor	Nota
1	2,0	
2	4,0	
3	4,0	
Total	10,0	

### Questão 1 (vale 2,0 pontos)

Cada termo na sequência de Fibonacci é dado pela soma dos dois termos anteriores. Começando com 1 e 2, os primeiros 10 termos na sequência são:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Escreva um programa em linguagem C que lê um número inteiro  $n$  maior que zero ( $n > 0$ ) e que calcula e exibe a soma dos termos da sequência de Fibonacci que têm valores pares, considerando apenas os termos na sequência cujo valor não ultrapasse  $n$ .

### Solução

```
#include <stdio.h>

int main()
{
    int fib_i_minus_1 = 1,
        fib_i = 1,
        atual = 2,
        soma = 0,
        n;

    printf("Digite o valor de n: ");
    scanf("%d",&n);

    while (atual <= n)
    {
        if (atual % 2 == 0)
            soma = soma+atual;

        fib_i_minus_1 = fib_i;
        fib_i = atual;
        atual = fib_i_minus_1 + fib_i;
    }

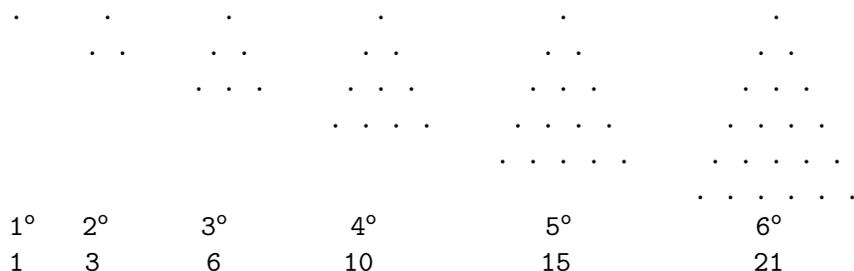
    printf("A soma vale: %d \n", soma);
    return 0;
}
```

**Questão 2** (vale 4,0 pontos)

A sequência de *números triangulares* é gerada por meio da adição de número naturais. Os 10 primeiros termos da sequência de números triangulares são:

1, 3, 6, 10, 15, 21, 28, 36, 45, 55, ...

Essa sequência é gerada a partir de um padrão de pontos que formam um triângulo, como mostrado abaixo. Adicionando uma nova linha de pontos e contando todos os pontos do triângulo, podemos encontrar o próximo número da sequência.



Note, por exemplo, que o 6° número triangular é calculado como  $1 + 2 + 3 + 4 + 5 + 6 = 21$ .

Os divisores positivos dos seis primeiros números triangulares são os mostrados a seguir:

- 1: 1
- 3: 1 3
- 6: 1 2 3 6
- 10: 1 2 5 10
- 15: 1 3 5 15
- 21: 1 3 7 21

Escreva um programa em linguagem C que lê um número inteiro  $n$  maior que zero ( $n > 0$ ) e exhibe como resposta a listagem dos divisores positivos dos  $n$  primeiros números triangulares, no mesmo formato mostrado no exemplo acima.

## Solução

```
#include <stdio.h>

int main()
{
    int n, candidato_divisor, i, triangular;

    printf("Digite o valor de n: ");
    scanf("%d", &n);

    triangular = 0;
    for (i = 1; i <= n; i++)
    {
        /* gera e exibe o proximo numero da sequencia de triangulares */
        triangular += i;
        printf("%d: ", triangular);

        /* encontra e exibe os divisores do numero triangular atual da sequencia */
        for (candidato_divisor = 1; candidato_divisor <= triangular; candidato_divisor++)
        {
            if (triangular % candidato_divisor == 0)
                /* encontrou um novo divisor */
                printf("%d ", candidato_divisor);
        }

        printf("\n");
    }
    return 0;
}
```

**Questão 3** (vale 4,0 pontos)

Sejam  $a$ ,  $b$  e  $c$  três inteiros, todos maiores que zero.

Suponha que  $a$  e  $b$  representam valores de moedas disponíveis e que  $c$  é o valor de uma cédula.

Escreva um programa em linguagem C que lê  $a$ ,  $b$  e  $c$ , e verifica se é possível trocar uma cédula de valor  $c$  por um punhado de moedas de valor  $a$  e  $b$ . Imprimir UMA (e apenas uma) forma de se fazer a troca (quantidade de moedas de valor  $a$  e de valor  $b$ ) caso a troca seja possível, ou uma mensagem de que a troca não é possível em caso contrário.

As mensagens emitidas pelo seu programa devem ser idênticas às mensagens mostradas nos exemplos abaixo (as partes sublinhadas são dados de entrada ou dependem dos dados de entrada).

```
Digite o valor de uma moeda: 5  
Digite o valor da outra moeda: 3  
Digite o valor da cédula: 11  
1 moeda(s) de 5 e 2 moeda(s) de 3
```

```
Digite o valor de uma moeda: 5  
Digite o valor da outra moeda: 3  
Digite o valor da cédula: 7  
Não é possível trocar a cédula
```

```
Digite o valor de uma moeda: 3  
Digite o valor da outra moeda: 5  
Digite o valor da cédula: 30  
0 moeda(s) de 3 e 6 moeda(s) de 5
```

```
Digite o valor de uma moeda: 4  
Digite o valor da outra moeda: 6  
Digite o valor da cédula: 121  
Não é possível trocar a cédula
```

```
Digite o valor de uma moeda: 4  
Digite o valor da outra moeda: 7  
Digite o valor da cédula: 17  
Não é possível trocar a cédula
```

## Solução 1 – sem repetições encaixadas

```
#include <stdio.h>
#define FALSE 0
#define TRUE 1

int main()
{
    int valor_moeda_1, valor_moeda_2, valor_cedula,
        quantidade_1, quantidade_2, achou = FALSE;

    printf("Digite o valor de uma moeda: ");
    scanf("%d", &valor_moeda_1);
    printf("Digite o valor da outra moeda: ");
    scanf("%d", &valor_moeda_2);
    printf("Digite o valor da cédula: ");
    scanf("%d", &valor_cedula);

    quantidade_1 = 0;
    while ((valor_moeda_1 * quantidade_1) <= valor_cedula
        && achou == FALSE)
    {
        quantidade_2 = (valor_cedula - valor_moeda_1 * quantidade_1) / valor_moeda_2;

        if ((valor_moeda_1 * quantidade_1 + valor_moeda_2 * quantidade_2) == valor_cedula)
            achou = TRUE;
        else
            quantidade_1++;
    }

    if (achou == TRUE)
        printf("%d moeda(s) de %d e %d moeda(s) de %d\n",
            quantidade_1, valor_moeda_1, quantidade_2, valor_moeda_2);
    else
        printf("Não é possível trocar a cédula\n");

    return 0;
}
```

## Solução 2 – com repetições encaixadas

```
#include <stdio.h>
#define FALSE 0
#define TRUE 1

int main()
{
    int valor_moeda_1, valor_moeda_2, valor_cedula, quantidade_1, quantidade_2, achou = FALSE;
    printf("Digite o valor de uma moeda: ");
    scanf("%d", &valor_moeda_1);
    printf("Digite o valor da outra moeda: ");
    scanf("%d", &valor_moeda_2);
    printf("Digite o valor da cédula: ");
    scanf("%d", &valor_cedula);

    quantidade_1 = 0;
    while ((valor_moeda_1*quantidade_1) <= valor_cedula && achou == FALSE)
    {
        quantidade_2 = 0;
        while ((valor_moeda_1 * quantidade_1 + valor_moeda_2 * quantidade_2) < valor_cedula)
            quantidade_2++;

        if ((valor_moeda_1 * quantidade_1 + valor_moeda_2 * quantidade_2) == valor_cedula)
            achou = TRUE;
        else
            quantidade_1++;
    }

    if (achou == TRUE)
        printf("%d moeda(s) de %d e %d moeda(s) de %d\n",
            quantidade_1, valor_moeda_1, quantidade_2, valor_moeda_2);
    else
        printf("Não é possível trocar a cédula\n");

    return 0;
}
```