

Prof. Msc. Elias Batista Ferreira  
Prof. Dr. Gustavo Teodoro Laureano  
Profa. Dra. Luciana Berretta  
Prof. Dr. Thierson Rosa Couto

## **Sumário**

<b>1</b>	<b>Considerações para Assembly</b>	<b>2</b>
<b>2</b>	<b>Arrecadação de Jogos (+)</b>	<b>3</b>
<b>3</b>	<b>Quadrado de pares (+)</b>	<b>4</b>
<b>4</b>	<b>Maior segmento igual de uma sequência (++)</b>	<b>5</b>
<b>5</b>	<b>Número perfeito (+++)</b>	<b>6</b>

# 1 Considerações para Assembly

(i) Só considerar números inteiros e multiplicação/divisão de inteiros

(ii) Transformar a entrada do programa em um valor por linha

(iii) Usar o template baixo para os programas:

```
#-----  
.data  
  
# Variáveis aqui...  
  
.text  
  
.globl main  
main:  
    #-- Inicializacao: não mexa -----  
    pushq %rbp  
    movq  %rsp, %rbp  
    subq  $128, %rsp  
    movq  %rax, 8(%rsp) ; movq %rbx, 16(%rsp) ; movq %rcx, 24(%rsp)  
    movq  %rdx, 32(%rsp) ; movq %rsi, 40(%rsp) ; movq %rdi, 48(%rsp)  
    movq  %r8, 56(%rsp) ; movq %r9, 64(%rsp) ; movq %r10, 72(%rsp)  
    movq  %r11, 80(%rsp) ; movq %r12, 88(%rsp) ; movq %r13, 96(%rsp)  
    movq  %r14, 104(%rsp) ; movq %r15, 112(%rsp)  
    #-----  
  
# Código aqui...  
  
#-- Finalizacao: não mexa -----  
    movq  8(%rsp), %rax ; movq 16(%rsp), %rbx ; movq 24(%rsp), %rcx  
    movq  32(%rsp), %rdx ; movq 40(%rsp), %rsi ; movq 48(%rsp), %rdi  
    movq  56(%rsp), %r8 ; movq 64(%rsp), %r9 ; movq 72(%rsp), %r10  
    movq  80(%rsp), %r11 ; movq 88(%rsp), %r12 ; movq 96(%rsp), %r13  
    movq 104(%rsp), %r14 ; movq 112(%rsp), %r15  
  
    movl $0, %eax  
    leave  
    ret  
#-----
```

## 2 Arrecadação de Jogos (+)



(+)

Escrever um algoritmo que lê o público total de futebol, as percentagens de pessoas nas seguintes categorias: *popular*, *geral*, *arquibancada* e *cadeiras* e forneça a renda total do jogo. Sabe-se que o valor dos ingressos por categoria são dados pela tabela abaixo:

Categoria	Valor ingresso
Popular	R\$ 1,00
Geral	R\$ 5,00
Arquibancada	R\$10,00
Cadeiras	R\$ 20,00

### Entrada

A entrada contém uma linha inicial com um valor inteiro informando o número de casos de testes que ocorrem nas linhas seguintes. Cada caso de teste seguinte é formado por uma linha contendo os seguintes valores, separados entre si por um espaço:

- O número de pessoas que compraram ingresso para o jogo correspondente ao caso de teste.
- A percentagem de pessoas que compraram ingresso na categoria *Popular*.
- A percentagem de pessoas que compraram ingresso na categoria *Geral*.
- A percentagem de pessoas que compraram ingresso na categoria *Arquibancada*.
- A percentagem de pessoas que compraram ingresso na categoria *Cadeiras*.

### Saída

O programa deve gerar uma linha para cada caso de teste na entrada, contendo a frase: A RENDA DO JOGO N.  $x$  E =  $y$ , onde  $x$  corresponde a ordem do caso de teste na entrada e  $y$  é um valor real com duas casas decimais que corresponde ao valor da renda total do jogo  $x$ .

### Exemplo

Entrada
3
55000 20.2 50.4 30.2 10.2
49732 15.2 53.4 20.24 11.16
67890 30.0 42.20 23.8 4.0
Saída
A RENDA DO JOGO N. 1 E = 428010.00
A RENDA DO JOGO N. 2 E = 352003.09
A RENDA DO JOGO N. 3 E = 379505.09

### 3 Quadrado de pares (+)



(+)

1

Escreva um programa para ler um valor inteiro  $N$  e que gere o quadrado de cada um dos valores pares, de 1 até  $N$ , inclusive  $N$ , se for o caso.

#### Entrada

A entrada conterá uma linha com um valor inteiro  $N$ ,  $5 < N < 2000$ .

#### Saída

A saída deve conter, uma linha para cada quadrado computado. Em cada linha deve constar uma expressão do tipo  $x^2 = y$ , onde  $x$  é um número par e  $y$  é o seu valor elevado ao quadrado. Imediatamente após o valor de  $y$  deve aparecer o caractere de quebra de linha: ‘\n’.

#### Exemplo

Entrada
6
Saída
$2^2 = 4$
$4^2 = 16$
$6^2 = 36$

---

<sup>1</sup>Fonte: Site do URI - <https://www.urionlinejudge.com.br/judge/pt/problems/view/1073>.

## 4 Maior segmento igual de uma sequência (++)



(++) (POLI 87) Dados  $n$  e uma sequência de  $n$  números inteiros, determinar quantos segmentos de números iguais consecutivos compõem essa sequência.

### Entrada

O programa deve ler um número inteiro maior que zero  $n$  e uma sequência de  $n$  números inteiros em qualquer ordem.

### Saída

O programa deve apresentar a mensagem "O comprimento do segmento de numeros iguais e:  $k$ ", onde  $k$  é o tamanho do maior segmento crescente encontrado.

### Exemplo

<b>Entrada</b>
2
1 1
<b>Saída</b>
O comprimento do segmento de numeros iguais e: 2
<b>Entrada</b>
7
0 0 2 5 5 5 6
<b>Saída</b>
O comprimento do segmento de numeros iguais e: 3

## 5 Número perfeito (+++)



(+++)

Dado um número  $n$  inteiro e positivo, dizemos que  $n$  é perfeito se  $n$  for igual à soma de seus divisores positivos diferentes de  $n$ . Construa um programa que leia um número inteiro  $n$ , apresente a soma dos divisores de  $n$  e verifique se o número informado é perfeito ou não.

### Entrada

O programa deve ler um número inteiro  $n$ .

### Saída

O programa deve apresentar uma linha contendo o texto: " $n = d_1 + d_2 + d_3 + \dots + d_k = x$  (MENSAGEM)", onde  $n$  é o número lido,  $d_i$  são os divisores de  $n$  em ordem crescente,  $x$  é a soma dos divisores e MENSAGEM é a mensagem "NUMERO PERFEITO" ou "NUMERO NAO E PERFEITO".

### Observações

Suponha que o usuário sempre fornecerá um número maior que 1.

### Exemplo

Entrada
6
Saída
6 = 1 + 2 + 3 = 6 (NUMERO PERFEITO)

Entrada
12
Saída
12 = 1 + 2 + 3 + 4 + 6 = 16 (NUMERO NAO E PERFEITO)