

## Aula Prática 01 – Revisão BCC201 e Alocação Dinâmica de Memória

**Data de entrega: 26/08/2018 até 22:00.**

**Atenção:** *O que vale é o horário do RunCodes, e não do seu, ou do meu, relógio.*

### Procedimento para a entrega:

1. Para cada questão, implemente sua solução.
  2. Para testar suas soluções, implemente um único método *main()*, que poderá conter, por exemplo, um *Menu* de interações e possibilidades do usuário especificar os dados de entrada.
  3. Especifique o *Makefile* com as instruções necessárias para compilação e execução do seu código, sendo que o *Makefile* deve conter também o redirecionamento da entrada e da saída (e.g., `./prog.exe < input.txt > output.txt`.)
  4. Compacte em um único diretório o seu código fonte juntamente com o *Makefile*, o arquivo de entrada e o arquivo de saída usados para testes (e.g., *input.txt*, *output.txt*).
  5. Faça a entrega do arquivo compactado, obrigatoriamente em formato *.zip*, no *RunCodes*, na tarefa correspondente.
- Não utilize caracteres acentuados ou especiais para nomes de pastas, arquivos e na especificação de comentários no código.
  - Implemente em conformidade com boas práticas para reuso e modularização do código.

**- Bom trabalho!**

## Questão 01

Implemente um programa para calcular o  $\cos(X)$ . O valor de  $X$  deverá ser informado em graus. O valor do cosseno de  $X$  (em radianos<sup>1</sup>) será calculado pela soma dos  $N$  primeiros termos da série a seguir:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \frac{x^{12}}{12!} - \dots$$

### Entrada

A entrada começa com apenas um inteiro  $z$  na primeira linha, representando o número de cálculos a serem realizados. A seguir, são apresentadas as informações dos  $z$  cálculos a serem feitos, um em cada linha subsequente. Cada linha de informação de cálculo possui dois inteiros, sendo que: o primeiro é o valor em graus de  $X$ ; o segundo é a quantidade de termos  $N$  a serem utilizados no cálculo de  $\cos(X)$ .

<sup>1</sup>Utilize para a conversão a relação: 1 grau = 0.0174532925 radianos

## Saída

A saída deve respeitar o seguinte formato:

```
<valor do primeiro X em radianos> <valor do primeiro N> <valor do primeiro cos(X)>
<valor do segundo X em radianos> <valor do segundo N> <valor do segundo cos(X)>
:
<valor do último X em radianos> <valor do último N> <valor do último \cos(X)>
```

## Exemplo de entrada e Saída

Entrada	Saida
5	0.17 15 0.98481
10 15	0.87 10 0.64279
50 10	3.40 20 -0.96593
195 20	3.67 30 -0.86603
210 30	6.11 18 0.98481
350 18	

## Questão 02

Os pais de uma criança  $A$  investem inicialmente na poupança de seu filho o montante inicial  $X_A$ . Todo mês eles depositam  $D_A$  na conta. Essa poupança ainda tem um rendimento de  $R_A\%$  ao mês. Já os pais de uma criança  $B$  iniciam o investimento de seu filho com  $X_B$ , não realizam depósitos mensais, porém o investimento tem um rendimento de  $R_B\%$  ao mês (sendo obrigatoriamente  $R_A < R_B$ ).

Implemente um programa que calcule e imprima o saldo de ambos no prazo final de  $M$  meses. O programa deverá informar em quantos meses, dentro do prazo definido, a criança  $B$  terá mais dinheiro em conta do que a criança  $A$ , ou informar que isto não ocorrerá dentro deste prazo.

## Entrada

A entrada começa com um inteiro  $n$  na primeira linha, representando o número de simulações de investimentos a serem realizados. A seguir, são apresentadas as informações das  $n$  simulações, uma em cada linha. As informações de cada simulação são definidas pela sequência de valores:  $X_A D_A R_A X_B R_B M$ .

## Saída

A saída será formada pelo resultado de cada simulação, sendo uma linha para cada uma delas. São três saídas possíveis para cada simulação:

- <saldo da criança A> <saldo da criança B> <quantidade de meses em que B supera A>
- <saldo da criança A> <saldo da criança B> B nao supera A
- Rendimento de B invalido

## Exemplo de entrada e Saída

Entrada	Saida
3	1015.27 1061.06 8
500 50 1 450 10 9	Rendimento de B invalido
300 20 5 400 4 6	149.46 56.27 B nao supera A
100 10 2 50 3 4	

### Questão 03

Implemente uma função que receba uma matriz e imprima o maior valor, o menor valor e a média dos valores. O método `main` de seu programa deverá interpretar a entrada e realizar a chamada da função quantas vezes forem necessárias. A matriz deve ser alocada dinamicamente, a liberação de memória é obrigatória.

#### Entrada

A entrada começa com um inteiro  $n$  na primeira linha, representando o número matrizes a serem computadas. A seguir, são apresentadas as informações das  $n$  matrizes. As informações de cada matriz são definidas pela sequência de linhas: na primeira linha <quantidade L de linhas> <quantidade C de colunas>, e nas  $L$  linhas posteriores os  $C$  valores correspondentes às colunas de cada linha.

Antes das informações de cada matriz haverá uma linha vazia de separação.

#### Saída

A saída deve respeitar o seguinte formato:

```
<maior valor de M1> <menor valor de M1> <média dos valores de M1>
<maior valor de M2> <menor valor de M2> <média dos valores de M2>
:
<maior valor de Mn> <menor valor de Mn> <média dos valores de Mn>
```

#### Exemplo de entrada e Saída

Entrada	Saída
2	3 1 1.833 3 1 1.667
3 2	
1 2	
3 1	
2 2	
2 3	
1 2 3	
1 1 2	

### Questão 04

Deseja-se publicar o número de acertos de cada aluno em uma prova em forma de testes. A prova consta de  $N$  questões, cada uma com cinco alternativas identificadas por 1, 2, 3, 4 e 5. Para isso são dados:

- O gabarito.
- O número de alunos da turma.
- As respostas de cada aluno.

Seu programa deverá interpretar a entrada e computar os acertos de todos os alunos, fornecendo como saída o número de chamada do aluno e a quantidade de acertos que ele obteve.

Você deverá armazenar os dados em uma única matriz, que conterà o gabarito e os dados dos alunos, e esta deverá ter sua memória alocada e liberada dinamicamente.

## Entrada

A primeira linha fornece o número  $N$  de questões e a quantidade  $A$  de alunos. A segunda linha fornece as  $N$  respostas que compõem o gabarito. Finalmente, as  $A$  linhas posteriores fornecem as respostas de cada aluno, ao final de cada linha é fornecido o número de chamada do aluno que forneceu tais respostas.

## Saída

A saída deve respeitar o seguinte formato:

```
<numero de chamada do aluno1> <quantidade de respostas corretas do aluno1>  
<numero de chamada do aluno2> <quantidade de respostas corretas do aluno2>  
:  
<numero de chamada do alunoA> <quantidade de respostas corretas do alunoA>
```

## Exemplo de entrada e Saída

Entrada	Saida
5 8	1 3
2 1 2 3 5	2 3
1 3 2 3 5 1	3 5
1 1 4 3 5 2	4 3
2 1 2 3 5 3	5 3
2 4 2 1 5 4	6 2
3 1 5 3 5 5	7 4
2 3 2 1 4 6	8 3
2 2 2 3 5 7	
1 1 3 3 5 8	