Campeanado de programação em C
Disciplina: Algoritmos e Lógica de Programação II
Prof. Rafael S. Bressan

Regras do jogo.

- Tempo: 24h para entregar a solução dos problemas mencionados neste documentos.
- Equipe: As equipes podem ser de no máximo três alunos.
- Linguagem de programação: C
- Local de entrega: Class Room (Atividade 25/11/2020)
- Quem ganha: A equipe que entregar a maior quantidade de problemas solucionados no menor tempo.

1) Problema A - Sistema de controle de notas acadêmicas.
A secretaria acadêmica necessita de uma aplicação que ao receber um arquivo contendo os dados acadêmicos dos alunos o sistema retorna um arquivo com as informações de entrada e as situações de cada aluno (aprovados, reprovados ou exame).

Exemplo do arquivo de entrada (in.txt):
RA;NOME;1BIM;2BIM;

```
1  00348; Rafael Bressan; 10.0; 10.0
2  00150; Pedro M.; 5.8; 6.8
3  01888; Marcelo Farias da Silva; 4.6; 8.9
```

Calculo: Média aritmética.
MF = (1BIM + 2BIM)/2

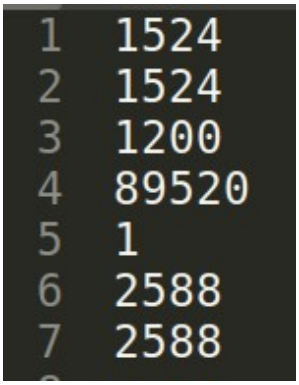
Regras: MF (Média Final)
0 … 1.9 → Reprovado
2.0 … 5.9 → Exame
6.0 … 10.0 → Aprovado

Exemplo do arquivo de saída (out.txt)

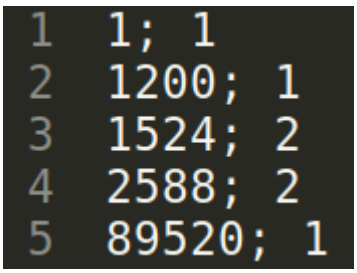
```
1  00348; Rafael Bressan; 10.0; 10.0; Aprovado
2  00150; Pedro M.; 5.8; 6.8; Aprovado
3  01888; Marcelo Farias da Silva; 4.6; 8.9; Aprovado
```

2) Problema B – Frequência numérica
Um cientista necessita identificar a frequência que os números se repetem em seus experimentos. Para tanto, ele salva arquivos contendo vários números inteiros, que podem variar entre 0 e 1000000. Desenvolva uma aplicação que leia o arquivo do cientista e devolva um arquivo apresentando a frequência que os números aparecem.

Exemplo do arquivo de entrada (in.txt):



Exemplo do arquivo de saída (out.txt)



3) Problema C – Portas (Autor: Leandro Luque (Fatec Mogi das Cruzes))

Durante as férias, João Pedro gosta de aproveitar o tempo brincando com os amigos do prédio onde mora. Uma de suas brincadeiras preferidas é o “Peão Abre-Fecha Porta”. A brincadeira começa com a escolha de um dos participantes, o peão. Em seguida, os outros participantes definem um número e o peão deve passar por todos os andares do prédio que sejam múltiplos desse número para inverter o estado da porta do apartamento localizado no andar – ou seja, se a porta estiver aberta, ele a fechará; se estiver fechada, ele a abrirá. As portas estão inicialmente todas fechadas e existe apenas um apartamento por andar. A brincadeira segue com os participantes definindo novos números e o peão abrindo/fechando as portas dos andares múltiplos desses números. Quando o grupo desejar, o peão estiver muito cansado, ou algum morador reclamar, cada participante da brincadeira, com exceção do peão, é questionado sobre o estado das portas do prédio em cada andar (na ordem do mais baixo para o mais alto). Aquele que acerta, ganha um doce e fica livre de ser peão durante todo o dia.

Como João Pedro adora doce e é um pouco preguiçoso, há tempos ele vem procurando por alguma forma de sempre ganhar a brincadeira. Para isso, ele pediu para o seu pai, um especialista em Informática, para desenvolver um programa que, dados os números que serão especificados pelos participantes, determina o estado final das portas dos andares.

Entrada

A entrada do programa é composta por diversos casos de teste. A primeira linha de cada caso de teste contém dois números inteiros A e N, separados por espaço, indicando o número de andares (variando de 1 a 100) e a quantidade de números que serão informados pelos participantes (variando de 1 a 200), respectivamente. Cada uma das N linhas seguintes contém um dos números especificados pelos participantes (variando de 1 a A). O último caso de teste é seguido por uma linha que contém dois zeros separados por um espaço em branco.

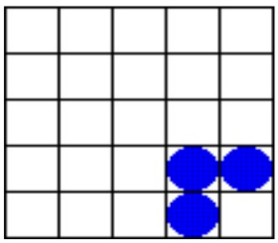
Saída

Para cada caso de teste, imprima uma linha contendo A caracteres, indicando o estado de cada uma das portas dos andares do prédio (o caractere mais à esquerda representa o andar mais baixo; o mais à direita representa o andar mais alto). Caso a porta do andar esteja aberta, imprima o caractere ‘O’. Caso a porta esteja fechada, imprima o caractere ‘C’.

Exemplo de entrada	Exemplo de saída
10 5 2 4 9 10 1 5 3 3 4 2 0 0	OCOOOCOOCO COOCC

4) Problema D – Jogo da Vida

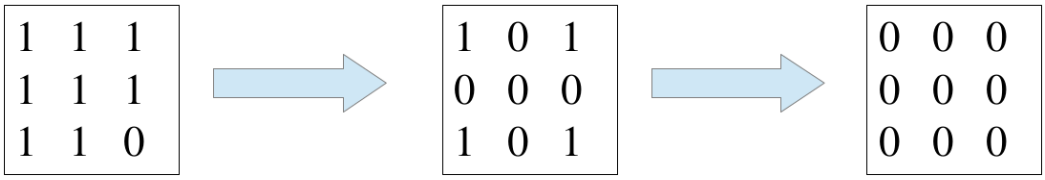
Simular o desenvolvimento de uma população representada por uma matriz com 0s e 1s. 1 representa uma célula viva, 0 sem nenhuma célula.



O cálculo da próxima geração segue algumas regras:
– Nascimento: uma célula com 3 vizinhos torna-se viva.
– Sobrevivência: uma célula com 2 ou 3 vizinhos permanece viva.
– Morte por solidão: uma célula morre se tiver menos que 2 vizinhos.
– Morte por superpopulação: uma célula morre se tiver mais do que 3 vizinhos.

Entradas (em um arquivo)
– Um inteiro n (número de linhas e colunas da matriz)
– Quantidade de gerações X (int)
– Matriz de inteiros (fazer a leitura linha por linha – 0 e 1)

Saída
A matriz após X gerações



5) Problema E – Análise numérica

O mesmo cientista do problema B precisa de uma ajuda para localizar um vetor que contem a menor distância entre N vetores extraídos dos experimentos que ele está realizando. A semelhança entre os vetores podem ser calculadas utilizando a distância euclidiana. O problema consistem em receber um arquivo contendo N vetores, calcula-se a distância entre todos os vetores e apresenta o vetor que contem a menor distância entre todos, como exemplo, se 4 vetores forem passados para programa, localiza-se qual deles tem a menor distância entre todos. Vale ressaltar que a entrada pode conter N vetores e cada vetor pode conter M elementos.

Exemplo de entrada:

```
5.1,3.5,1.4,0.2
4.9,3.0,1.4,0.2
4.7,3.2,1.3,0.2
4.6,3.1,1.5,0.2
5.0,3.6,1.4,0.2
5.4,3.9,1.7,0.4
4.6,3.4,1.4,0.3
5.0,3.4,1.5,0.2
4.4,2.9,1.4,0.2
4.9,3.1,1.5,0.1
5.4,3.7,1.5,0.2
4.8,3.4,1.6,0.2
4.8,3.0,1.4,0.1
4.3,3.0,1.1,0.1
5.8,4.0,1.2,0.2
5.7,4.4,1.5,0.4
5.4,3.9,1.3,0.4
5.1,3.5,1.4,0.3
5.7,3.8,1.7,0.3
5.1,3.8,1.5,0.3
5.4,3.4,1.7,0.2
5.1,3.7,1.5,0.4
4.6,3.6,1.0,0.2
5.1,3.3,1.7,0.5
4.8,3.4,1.9,0.2
5.0,3.0,1.6,0.2
5.0,3.4,1.6,0.4
5.2,3.5,1.5,0.2
5.2,3.4,1.4,0.2
4.7,3.2,1.6,0.2
4.8,3.1,1.6,0.2
```

A saída consiste apenas na apresentação de qual vetor representa a menor distância entre todos os outros.