

CIn - UFPE

Introdução à Programação

Lista de Exercícios 3

Monitoria 2011.2

É proibido o uso das funções das bibliotecas string.h, ctype.h e semelhantes.

- 1) Escreva um programa em C que recebe 10 números inteiros positivos (entre 0 e 100000), e vai colocando esses números em um vetor de 10 posições. Logo após, seu programa deve verificar se cada valor desse vetor é perfeito ou não. Um valor é dito perfeito quando ele é igual a soma dos seus divisores excetuando ele próprio.

(Ex: 6 é perfeito, $6 = 1 + 2 + 3$, que são seus divisores).

O seu programa deve imprimir na tela os índices dos inteiros perfeitos em ordem decrescente.

Entrada:

3 2 6 12 45 101 28 90 21 78

Saída

6 2 /*Índices em ordem decrescente dos dois números perfeitos (6 e 28) */

- 2) O dodecafonismo (do grego dodeka: 'doze' e fonos: 'som') é um sistema de organização de alturas musicas. No dodecafonismo as 12 notas musicais são tratadas como equivalentes, ou seja, sujeitas a uma relação ordenada e não hierárquica. As notas são organizadas em grupos de doze notas denominados séries as quais podem ser usadas de diferentes maneiras. Faça um programa que receba 12 elementos de um vetor, o índice do vetor representa uma nota musical e o valor que está guardado naquele índice representa a ordem em que ele será executado. O vetor recebido será considerado a Série inicial. A partir dessa série seu programa deve gerar a seguinte série:

- Série Retrograda: Copia os elementos da serie original para um novo vetor mas com a ordem inversa da série original.

OBS: A troca deve ser feita varrendo o vetor e depois trocando as 2 posições.

Ex:

Entrada:

8,2,5,0,3,1,10,4,9,11,6,7 // Aqui o usuário preencheu o vetor.

3	5	1	4	7	2	10	11	0	8	6	9
0	1	2	3	4	5	6	7	8	9	10	11

// Explicação. Como o 8 foi o primeiro numero inserido pelo usuário, o elemento de índice 8 do vetor representa a primeira nota a ser executada, logo, o conteúdo do vetor na posição de índice 8 deverá conter 0 (representa a primeira nota tocada). A explicação para os outros itens é análoga.

Saída: Série Retrograda

7,6,11,9,4,10,1,3,0,5,2,8 // Esta deve ser a saída depois de ser executada a troca. Essa nova série deve ser armazenada em um novo vetor, e depois da troca deve-se imprimir os elementos na nova ordem.

Na imagem a primeira nota a ser executada é a de índice 7, logo o

8	6	10	7	4	9	1	0	11	3	5	2
0	1	2	3	4	5	6	7	8	9	10	11

conteúdo do vetor nessa posição deve ser 0 (representa a primeira nota a ser executada).

Obs: A imagem com o vetor não deve ser impressa na tela. Ela é apenas para demonstrar como fica o preenchimento do vetor com as operações escolhidas. O que deve ser impresso na tela é a sequência da execução das notas como no exemplo acima.

3) Em lógica e matemática, uma **lógica proposicional** (ou cálculo sentencial) é um sistema formal no qual as fórmulas representam *proposições* que podem ser formadas pela combinação de proposições atômicas usando *conectivos lógicos* e um sistema de *regras de derivação*, que permite que certas fórmulas sejam estabelecidas como "*teoremas*" do sistema formal. Duas fórmulas (ou sentenças) são ditas equivalentes quando, para quaisquer valor das variáveis (combinações de 0 e 1), ambas geram o mesmo resultado, ou seja, possuem resultado da tabela verdade igual. (*Wikipedia*)

Você deve fazer um programa que receba do usuário duas sentenças e diga se elas são equivalentes ou não.

Observações: Cada expressão terá tamanho máximo de 5 caracteres, haverá apenas DUAS variáveis (x e y). Vale salientar que pode haver apenas UMA ocorrência DE CADA variável na expressão de entrada. As operações permitidas são E, OU, XOR, IMPLICA e NOT. Os respectivos caracteres são os seguintes:

E = '^'

OU = '|'

XOR = '?'

Implica = '>'

Not = '¬'

Quaisquer caracteres que não sejam estes ou as letras EM MAIUSCULO 'X' e 'Y' devem invalidar a expressão e uma mensagem de erro deve ser mostrada ao usuário. A expressão será simples e não conterá parênteses. As expressões NÃO devem conter espaço.

Entrada:

X^Y //expressão1

X|Y//expressão2

Saída

Não são equivalentes //resultado

Entrada:

$\neg X|Y$ //expressão1

X>Y //expressão2

Saída

São equivalentes //resultado

Entrada:

X > Y //expressão1

Z & (Y) //expressão2

Saída

Expressoes Invalidas! //resultado

4) Você deverá criar novas funções para uma biblioteca de string.

- Inserir – Insere um caractere em uma determinada posição da string;
- Remover – Remove o caractere determinado pela posição da string;
- Substituir – Substitui o caractere de uma determinada posição por um novo caractere;
- Alternar – Se o caractere determinado pela posição for minúsculo, alternar para maiúsculo, caso seja maiúsculo, alternar para minúsculo, caso não seja uma letra, não faz nada.
- Compara – O usuário insere outra string e o programa informa se ela é lexicograficamente posterior, superior ou igual imediatamente.
- Concatenar – O usuário insere uma string e ela deve ser concatenada SE ainda couber no espaço de 50 caracteres. Essa nova string poderá conter no máximo 20 caracteres.
- Finalizar – Encerra o programa e mostra a string com as modificações

Exemplos

Entrada:

caza
substituir 3 s
alternar 1
finalizar

Saída:

Casa

Entrada:

Ingreja
remover 2
finalizar

Saída:

Igreja

Entrada:

```
carapato  
inserir 3 r  
finalizar
```

Saída:

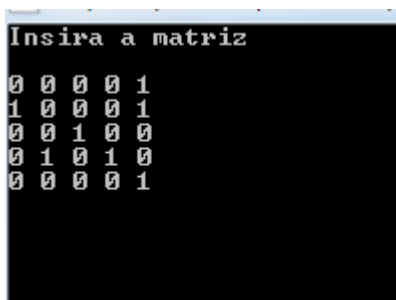
```
Carrapato
```

5) O azulejo é um importante material no ramo da construção civil por sua história e por sua característica de impermeabilidade unida à capacidade decorativa que é explorada principalmente em mosaicos. Cientistas desenvolveram uma tinta a base e bactérias inteligentes que podem mudar constantemente de lugar num azulejo de acordo com alguns critérios. Como a procura por essa tal tinta cresceu, você deve desenvolver um programa que simule essa mudança para a apreciação de clientes.

Uma matriz $n \times n$ ($0 < n \leq 50$) será preenchida com apenas '0' e '1' onde o 1 indica a presença da tinta num azulejo (representado por uma célula da matriz) e '0', a ausência. Após isso, essa matriz deve ser impressa sendo o '1' impresso com um 'X' e o '0' impresso com um espaço em branco. A cada interação, as mudanças deverão ser aplicadas e a nova matriz impressa. Uma nova interação deve começar quando o usuário digitar uma tecla. Se a tecla 'E' for pressionada, o programa encerra.

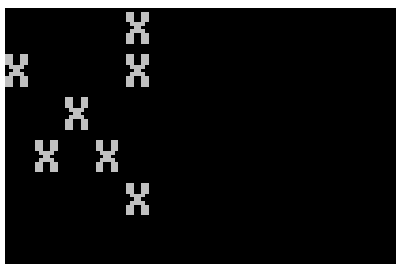
Os critérios de mudança são os seguintes:

- Se um azulejo com tinta tiver apenas um ou nenhum vizinho também com tinta, perde a tinta, pois a bactéria morre.
- Se um azulejo com tinta tiver 4 ou mais vizinhos com tinta, fica sem tinta pois a bactéria morre por falta de ar.
- Se um azulejo com tinta tem exatamente dois ou três vizinhos, ele permanece com tinta.
- Todo azulejo sem tinta que contém exatamente dois ou três vizinhos com tinta, passa a ser pintado.



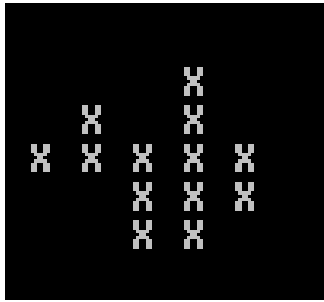
```
Insira a matriz  
0 0 0 0 1  
1 0 0 0 1  
0 0 1 0 0  
0 1 0 1 0  
0 0 0 0 1
```

Inserção da Matriz

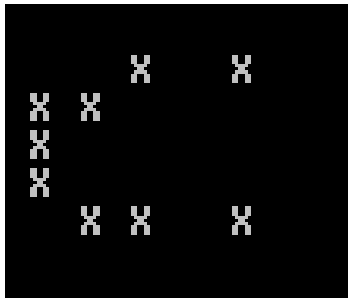


```
  X  
X  X  
  X  X  
X  X  
  X
```

Print da matriz após a inserção e o programa espera até o usuário digitar uma tecla



Primeiro Laço. Matriz sofre as mudanças e o programa espera que o usuário digite uma tecla. Aqui, após essa impressão ele digitou uma tecla diferente de 'E'



Segundo Laço. Matriz sofre as mudanças e o programa espera que o usuário digite uma tecla. Aqui, após essa impressão ele digitou 'E'. O programa foi encerrado.