

Implemente uma lista circular com tamanho fixo (40 elementos) em que são inseridas e "removidas" palavras. Os elementos armazenados devem ter a seguinte estrutura:

```
typedef struct no {  
    char pal[20];  
    int urg;  
    struct no *prox; } Elem;
```

A entrada para seu programa é um arquivo, chamado pacotes.dat, contendo K linhas contendo:

intA string intB

Em que intA pode ter os valores 0 (inserir palavra) ou 1 ("remover" palavra), string é um mnemônico (a palavra) entre:

ACK, NOACK, SEND, RECV, DEL, PRTY, NULL

E intB é um valor entre 0 e 10, a ser armazenado no campo urg. Esse valor será usado quando a palavra "removida" for PRTY, fazendo com que a próxima palavra a ser "removida" seja aquela que está no máximo a urg posições de PRTY (não pode passar da última palavra inserida).

A execução do programa termina ao chegar na linha contendo a string NULL.

A saída de seu programa deve ser um arquivo, chamado lidos.dat, contendo todas as palavras "removidas" durante a execução, uma por linha do arquivo.

ARQUIVO DE ENTRADA: pacotes.dat

ARQUIVO DE SAÍDA: lidos.dat

ARQUIVO COM CÓDIGO FONTE: buffercirc.c