

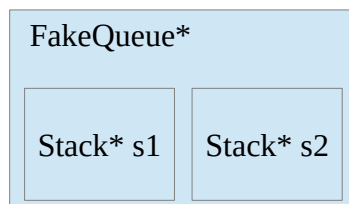
Estrutura de Dados I – Lista de Exercícios Fila (Implementação Estática)

prof. Andres Jessé Porfirio

- 1) Dada uma Fila f e uma Pilha p , crie uma função que recebe e inverte f utilizando p (p deve ser criada e destruída dentro da função, não sendo visível ao restante do programa). Crie também uma função que recebe uma Fila f e imprime todos os seus elementos. Nota: p e f são estruturas que trabalham apenas com caracteres. Utilize a seguinte especificação:

```
void invert_queue(Queue* q);  
void print_queue(Queue* q);
```

- 2) Utilizando duas Pilhas, simule o comportamento FIFO de uma Fila (implemente add, remove, isEmpty, etc). Dica: crie uma nova estrutura chamada “FakeQueue”, dentro dela adicione duas Pilhas. Crie também funções específicas para esta nova estrutura ex: “fake_queue_create”, “fake_queue_add”, etc.



FakeQueue deve funcionar como uma caixa preta, s1 e s2 não deve ser visíveis ao programa principal, as funções “add”, “remove”, etc. devem “enganar” o usuário, fazendo-o acreditar que manipula um Fila, sendo que na verdade são duas Pilhas “mascaradas”. FakeQueue deve ter comportamento FIFO!

- 3) Faça o teste de mesa do seguinte fragmento de código e apresente sua saída (printf's):

```
Fila* f;  
adicionar(f,5);  
adicionar(f,6);  
adicionar(f,7);  
adicionar(f,8);  
int x = primeiro(f);  
int y;  
y = remover(f);  
adicionar(f,x);  
adicionar(f,++y);  
remover(f);  
y = remover(f);  
while( esta_vazia(f)==0 )  
    x = remover(f);  
Printf("%d\n",x);
```

- 4) Faça o teste de mesa do seguinte fragmento de código e apresente sua saída (printf's):

```
Fila* f;  
int x = 2;  
adicionar(f,4);  
adicionar(f,x);  
adicionar(f,primeiro(f));  
adicionar(f,3+primeiro(f));  
adicionar(f,0);  
remover(f);  
while(esta_vazia(f)==0)  
{  
    x = remover(f);  
    printf("%d\n",x);  
}
```