



Transcrição de aula

Disciplina	Programação Imperativa - 1º ano - LEI	
Secretário	Número: 62826	Nome: Nuno Silva
Data: 2011-04-26 T7	Nº Página	

(99 alunos)

SUMÁRIO

Implementação de stacks e queues em memória dinâmica.
 Caso de estudo = trabalhar com números gigantes.

```

LIFO - Stack < push - inserção na cabeça
                pop - remoção da cabeça
FIFO - Queue < enqueue - inserção na cauda
              (first in first out) dequeue - remoção na cauda

typedef struct slint
{
  int valor;
  struct slint * seg;
} *Lista, Nada;

Lista Pop (Lista l);
Lista EnQueue (Lista l, int n)

Lista EnQueue (Lista l, int n)
{
  Lista aux;
  if (!l)
  {
    aux = (Lista) malloc (sizeof (Nada));
    aux -> valor = n;
    aux -> seg = NULL;
  }
  }
  
```

(

)

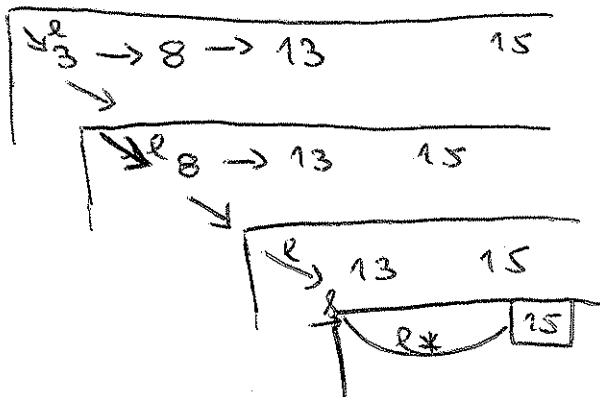


Transcrição (folha de continuação)

```
return aux;  
}  
else  
{  
l -> seg = EnQueue (l -> seg, n);  
return l;  
}  
}
```



```
l1 = EnQueue (l1, 15);
```



```
Lista EnQueueIt (Lista l, int n)
```

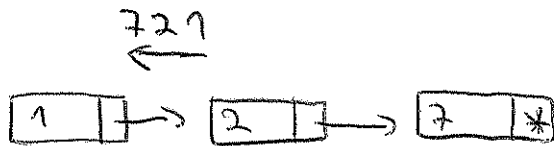
```
{ lista aux = l, aux2;  
while (aux)  
aux = aux -> seg;
```

```
typedef struct sLind  
{ char digito;  
struct sLind * seg;  
}* Lista, Nada;
```





Transcrição (folha de continuação)



```
void ListaNum (Lista l)
{
  if (l)
  {
    ListaNum (l->seg);
    printf ("%c", l->digito);
  }
}
```

```
Lista SomaListas (Lista a, Lista b)
{
  int soma, carry = 0;
  Lista res;
  if (!a)
    res = Duplica (b);
  else
    if (!b)
      res = Duplica (a);
    else
      {
        res = (Lista) malloc (sizeof (Node));
        soma = a->digito - '0' + b->digito - '0';
        res->digito = soma % 10 + '0';
        carry = soma / 10;
        res->seg = SomaListas (carry, a->seg, b->seg, carry);
      }
  return res;
}
```