



Transcrição de aula

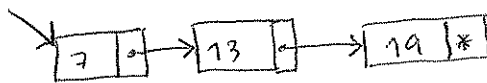
pág 42-44

Disciplina	Programação Imperativa - 1º ano - LEI	
Secretário	Número: A63 & 16	Nome: Marco Pereira
Data:	2011 - 04 - 12	Nº Página 3
Turno: T2		Nº Alunos 91

SUMÁRIO

Listas ligadas = listar, contar e inserir.

Listas Ligadas



```
typedef struct sLista
{
    int valor;
    struct sLista *seg;
} *Lista, NodeLista;

int main() ListaLista()
{
    Lista l1 = NULL;
    l1 = PushLista (PushLista (l1, 7), 5);
    ListaLista (l1);
}

void ListaLista (Lista l) /* Versão Iterativa */
{
    while (l) /* l != NULL */
    {
        printf ("%d, l -> valor);
        l = l -> seg;
    }
}
```




Transcrição (folha de continuação)

```

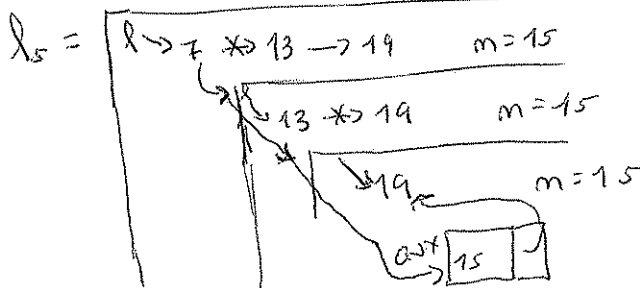
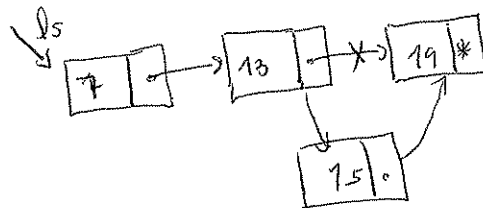
Lista sortedInsert (Lista l, int m) /* inserção ordenada */
{
  if (!l || (m < l->valor))
  {
    aux = (Lista) malloc(sizeof(NodoLista));
    aux->valor = m;
    aux->seg = l;
    return aux;
  }
  else
  {
    l->seg = sortedInsert (l->seg, m);
    return l;
  }
}

```

```

l5 = sortedInsert (l5, 15)

```



```

int Count (Lista l)
{
  if (!l)
    return 0;
  else
    return 1 + Count (l->seg);
}

```

```

{ ou
  return (!l) ? 0 : 1 + Count
    (l->seg)
}

```

()

()



Transcrição (folha de continuação)

```
Lista DeleteList (Lista l)
{
  if (l)
    DeleteList (l->seg)
  free (l)
  return null;
}

ls = DeleteList (ls);
```

DeleteList (ls);

