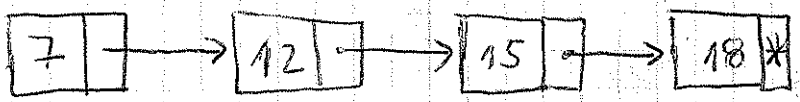


Ricardo Silva

20/4/2010



```

typedef struct sLInt
{
  int valor;
  struct sLInt *seg;
} *LInt, Noto;
  
```

remover 15  
free(l)

Em Haskell.

```

remover: [Int] -> Int -> [Int]
remover [] = []
remover (x:xs) m
  | x == m = xs
  | otherwise = x:remover xs m
  
```

Em C

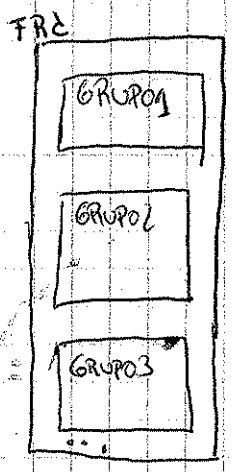
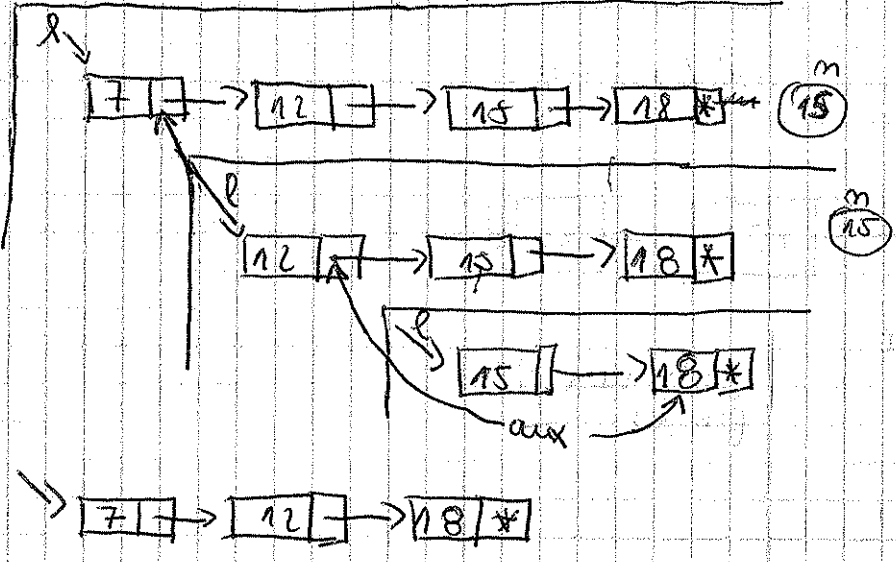
```

LInt remover (LInt l, int m)
{
  LInt aux;
  if (!l || l->valor > m)
    return l;

  else
    if (l->valor == m)
      aux = l->seg;
      free(l);
      return aux;

  else
    l->seg = remover(l->seg, m);
    return l;
}
  
```

$l_1 = \text{remover}(l_1, 15)$



Grupo  
 - nome  
 - LAlunos [1..3]  
 - OBS  
 - Nota

Aluno  
 - nome  
 - numero

```
typedef struct sAluno
{
    char nome[60];
    char numero[10];
} Aluno;
```

```
typedef struct sLAluno
{
    Aluno a;
    struct sAluno * seg;
} *LAluno, **LAluno;
```

```
typedef struct sGrupo
{
    char nome[20];
    LAluno lalunos;
    char obs[250];
    float nota;
} Grupo;
```

```
typedef struct sFrD
{
    Grupo g;
    struct sFrD * seg;
} *FrD, **FrD;
```

```
void listar (FrD f)
{
    printf ("Resultados ... \n");
    if (f)
    {
        listarGrupo (f -> g);
        listar (f -> seg);
    }
}
```

```
void listaGrupo (Grupo g)
{
    printf ("Grupo: %s\n", g.nome);
    listaAlunos (g.alunos);
    printf ("Obs: %s\n", g.obs);
    printf ("Nota: %.2f\n", g.nota);
}
```

```
void listaAlunos (LAluno l)
{
    if (l)
    {
        listaAluno (l->a);
        listaAlunos (l->seg);
    }
}
```

```
void listaAluno (Aluno a)
{
    printf ("%s %s\n", a.nome, a.numero);
}
```

