

Resolução dos exercícios

```
int folhas (Arv* a)
{
    if (arv_vazia(a->esq) && arv_vazia(a->dir)) return 1;
    else if (!arv_vazia(a->esq) && arv_vazia(a->dir)) return folhas(a->esq);
    else if (arv_vazia(a->esq) && !arv_vazia(a->dir)) return folhas(a->dir);
    return folhas(a->esq) + folhas(a->dir);
}
```

```
int igual (Arv* a, Arv* b)
{
    return
        arv_vazia(a) && arv_vazia(b)
        ||
        (!arv_vazia(a) && !arv_vazia(b) &&
         a->info == b->info &&
         igual(a->esq, b->esq) &&
         igual(a->dir, b->dir));
}
```

```
int ocorrencias_x (Arv* a, int x)
{
    if (a==NULL) return 0;
    if (a->info == x) return 1 + ocorrencias_x(a->dir, x);
    if (a->info < x) return ocorrencias_x(a->dir, x);
    return ocorrencias_x(a->esq, x);
}
```

```
int imprime_folhas (Arv* a)
{
    if (a==NULL) return 0;
    if ((a->esq == NULL) && (a->dir == NULL)) /* imprime só as folhas */
        {printf("%d",a->info); return 1;}
    imprime_folhas(a->dir); /* em ordem não-crescente */
    imprime_folhas(a->esq);
}
```

```

void imprime(Arv* a)
{
    /* supõe expressão não vazia */
    if ((a->esq == NULL) && (a->dir == NULL))
        printf("%g", a->valor);
    else {
        imprime(a->esq);
        imprime(a->dir);
        printf("%c", a->op);
    }
}

```

```

float avalia(Arv* a)
{
    /* supõe expressão não vazia */
    if ((a->esq == NULL) && (a->dir == NULL))
        return a->valor;
    else {
        if(a->op == '+') return avalia(a->esq) + avalia(a->dir);
        if(a->op == '-') return avalia(a->esq) - avalia(a->dir);
        if(a->op == '*') return avalia(a->esq) * avalia(a->dir);
        if(a->op == '/') return avalia(a->esq) / avalia(a->dir);
    }
}

```

```

int folhas (ArvVar* a)
{
    ArvVar* p;
    int n = 0;
    if (a->prim == NULL) return 1;
    for (p=a->prim; p!=NULL; p=p->prox)
        {n = n + folhas(p);};
    return n;
}

```

```
int igual (ArvVar* a, ArvVar* b)
{ ArvVar* p;
  ArvVar* q;
  int n = 0;
  if (a == NULL && b == NULL) return 1;
  for (p=a->prim, q=b->prim; p!=NULL && q!=NULL; p=p->prox, q=q->prox)
    {if (!igual(p,q)) return 0;};
  return 1;
}
```