2)

a) O ponteiro p não foi inicializado. Nesse caso, ele aponta para um endereço de memória arbitrário.
b) O ponteiro p foi inicializado, porém não aponta para a variável x. Nesse caso, ele aponta para o endereço de memória retornado pelo malloc.
c) O ponteiro p foi inicializado como p = x. O correto seria inicializá-lo como p = &x (p recebe o endereço de memória de x).
d) Inicialização está correta. Porém, dentro do printf, os campos do struct x estão sendo acessados através de p com o operador ponto (p.inteiro, p.real...). O jeito ideal de acessar os valores de um struct a partir de um ponteiro é por meio do operador seta. Dessa forma, o correto seria p->inteiro, p->real, etc.
e) Não existe nenhum erro de sintaxe ou sintática no código. Porém, p é inicializado duas vezes: primeiro recebe o retorno de um malloc e depois recebe o endereço de memória de x. Ou seja, o malloc é trivial.
f) O erro está no fato do código estar fazendo aritmética de ponteiros em um ponteiro de tipo void. Isso não é permitido pois o compilador não sabe como somar o endereço de memória.

3)

De A para B:

```
p->pt = q;
```

De A para C:

```
p->pt = q;
q->pt = q;
```

4)

De A1 para A2:

```
pt1->b = pt2;
pt2->b = pt3;
```

De B1 para B2:

```
pt1->b = pt2;
pt1->a = pt2->a;
pt1->a->b = pt1;
pt2->a = pt1;
```

Cuidado com a ordem, não inicie com pt2->a=pt1

5)

```
p->esq->dir->esq = p->esq;
p->esq->dir->dir = p;
p->esq = p->esq->dir;
```

Cuidado com a ordem, não inicie alterando p->esq

6)

A)

```c
//Solução recursiva

int contaNodo_r(struct nodo *p) {
    if( p == NULL )
        return 0;

    return 1 + contaNodo_r(p->link);
}

//Solução iterativa

int contaNodo_i(struct nodo *p) {
    if ( p == NULL )
        return 0;

    int cont = 1;
    struct nodo *temp = p->link;

    while( temp != NULL ) {
        cont++;
        temp = temp->link;
    }

    return cont;
}
```

B)

```c
//Nesse caso, a solução recursiva é deselegante

//Solução iterativa

int contaNodo_i(struct nodo *p) {
    if ( p == NULL )
        return 0;

    struct nodo *inicio = p;
    int cont = 1;
    struct nodo *temp = p->link;

    while( temp != inicio ) {
        cont++;
        temp = temp->link;
    }

    return cont;
}
```

C)

```c
//Nesse caso, a solução recursiva é deselegante

//Solução iterativa

int contaNodo_i(struct nodo *p) {
    if ( p == NULL )
        return 0;

    int cont = 1;
    struct nodo *temp1 = p;
    struct nodo *temp2 = p->link;

    while( temp1 != temp2 ) {
        cont++;
        temp1 = temp2;
        temp2 = temp2->link;
    }

    return cont;
}
```