Solución Pregunta 1 (25%)

1. (3 puntos)

```
#Opcion 1: Usando ciclo While
def es_abecedaria(s):
        n = 1
        es_abc = False

        while n < len(s):
                if s[n-1] <= s[n]:
                        es_abc = True
                else:
                        return False
        n += 1

return es_abc

#Opcion 2: Usando listas
def es_abecedaria(s):
        ordenado = list(s)
        original = list(s)
        ordenado.sort()

        if ordenado == original:
                return True
        else:
                return False

# Opcion 3: Usando ciclo for
def es_abecedaria(s):
        es_abc = False

        for i in range(len(s) - 1):
                if s[i] <= s[i+1]:
                        es_abc = True
                else:
                        return False

        return es_abc
```

2. (3 puntos)

```
#Opcion 1 y 2: replace
def contar_abecedarias(s):
        #Sacamos caracteres especiales
        for c in ",.;:":
                s.replace(c,'')

        lista = s.split(" ") #Separamos en espacios
        contador = 0

        #Opcion 1: ciclo for
        for palabra in lista:
                if es_abecedaria(palabra):
                contador += 1

        #Opcion 2: ciclo while
        i = 0
        while i < len(lista):
                if es_abecedaria(lista[i]):
                        contador += 1
                i += 1

        return contador

#Opcion 3 y 4: usando strip
def contar_abecedarias(s):
        lista = s.split(" ") #Separamos en espacios

        for palabra in lista:
                lista.remove(palabra)
                        for c in ",.;:":
                                p = palabra.strip('c') #Sacamos caracteres especi
                lista.append(p)

        contador = 0
        #Opcion 3: ciclo for
        for palabra in lista:
                if es_abecedaria(palabra):
                        contador += 1

        #Opcion 4: ciclo while:
        i = 0
        while i < len(lista):
                if es_abecedaria(lista[i]):
                        contador += 1
                i += 1
        return contador
```

Solución Pregunta 2 (25%)

1. (1 punto)

```
def faltan(album):
        laminas_faltantes = []

        for i in range(1,349):
                if not(i in album):
                        laminas_faltantes.append(i)

        return laminas_faltantes
```

2. (2.5 puntos) .

```
def numero(album1,album2,reps1,reps2):
        traspaso_1 = 0
        traspaso_2 = 0

        #Eliminamos las repetidas dentro de las listas repetidas
        r1 = []
        r2 = []

        for lamina in reps1:
                if not(lamina in r1):
                        r1.append(lamina)

        for lamina in reps2:
                if not(lamina in r2):
                        r2.append(lamina)

        #Vemos cuanto puede traspasar 1 a 2:
        for lamina in r1:
                if not(lamina in album2):
                        traspaso_1 += 1

        #Vemos cuanto puede traspasar 2 a 1:
        for lamina in r2:
                if not(lamina in album1):
                        traspaso_2 += 1

        #Opcion 1: comparacion directa
        if traspaso_1 < traspaso_2:
                return traspaso_1
        else:
                return traspaso_2

        #Opcion 2: usando funcion min(x,y)
        return min(traspaso_1,traspaso_2)
```

3. (2.5 puntos)

```
def traspasar(album,reps,num):
        agregadas = []

        #Buscamos las laminas que sirven
        for lamina in reps:
                if not(lamina in album):
                        if num > 0:
                                album.append(lamina)
                                agregadas.append(lamina)
                                num -= 1
                        else:
                                break

        #Ahora eliminamos en reps:
        for lamina in agregadas:
                if lamina in reps:
                        reps.remove(lamina)

        #Ordenamos album
        album.sort()
```

Solución Pregunta 3 (25%)

Pendiente

Solución Pregunta 4 (25%)

1. (3 puntos)

```
def mezclar(s1,s2):
        if len(s1) >= 1 and len(s2) >= 1:
                if s1[0] < s2[0]:
                        return s1[0] + mezclar(s1[1:],s2)
                if s1[0] == s2[0]:
                        return s1[0]+s2[0]+mezclar(s1[1:],s2[1:])

                return s2[0] + mezclar(s1,s2[1:])

        if len(s1) == 0:
                return s2
        if len(s2) == 0:
                return s1
```

2. (3 puntos) Pendiente