Pauta Auxiliar 4

Pregunta 1

```
// Swap en un arreglo
void Q( int a [ ] , int i , int j ){
  int aux1 = a [i];
  int aux2 = a [j];
  a[i] = aux2;
  a[j] = aux1;
}

int P( int a [ ] , int x, int y ){
  if(x >= y){
    return x;
  }
  while(true){
    if(a[x] >= 0){
      Q(a,x,y);
      y = y-1;
      if(x < y){
        #no hace nada
      }
      #claramente seria mejor poner solamente
      else{
        # if(x >= y) return x;
        return x;
        #no lo haremos solo para seguir mas literalmente el codigo de assembler
      }
    }
    else{
      x = x+1;
      if(x < y){
      #no hacer nada
      }
      else{
         return x;
      }
    }
  }
}
```

Pregunta 2

```
 .globl p
p:
pushl %ebp
movl %esp, %ebp
pushl %edi
pushl %esi
pushl %ebx
movl 8(%ebp), %esi
movl 12(%ebp), %edi
movl 16(%ebp), %ecx
decl %ecx # %ecx--
movl $0, %ebx
.L2:
leal (%ebx,%ecx), %eax
# eax= %ebx+%ecx
shrl %eax # %eax>>= 1 sin signo
movl (%edi,%eax,4), %edx
cmpl %esi, %edx
je .L5
cmpl %esi, %edx
jge .L6
leal 1(%eax), %ebx # %ebx= %eax+1
jmp .L8
.L6:
leal -1(%eax), %ecx # %ecx= %eax-1
.L8:
cmpl %ecx, %ebx
jbe .L2
movl $-1, %eax
.L5:
popl %ebx
popl %esi
popl %edi
popl %ebp
ret
```