# Assigment 2.

C-2.1

```
Algorithm findListMiddle(L)
    left = L.first();
    right = L.last();

    while !L.after(right) = left do
        left = L.after(left)
        right = L.before(right)

    return left
```

C-2.2.

```
Algorithm enqueue(el)
    en_stack.push(el)

Algorithm dequeue()
    if de_stack.isEmpty() then
        if en_stack.isEmpty() then
            throw Error
        else
            while !en_stack.isEmpty() do
                de_stack.push(en_stack.pop())

    return de_stack.pop()
```

C-2.3.

```
Algorithm        Push(o)
  queue.enqueue(o)
  for i := 1 to queue.size()-1 do
    queue.enqueue(queue.dequeue)
```

```
Algorithm        Pop()
  if queue.isEmpty() then
    error "stacks empty"
  return queue.dequeue();
```

```
Algorithm        Top()
  if queue.isEmpty() then
    error "stack empty"
  return queue.front();
```

```
Algorithm        IsEmpty()
  return queue.isEmpty()
```

C-2.3 B. Algorithm PowerSet(n)

```
S := new empty Sequence
s0 := new empty Sequence
s.insertLast(s0)

for i := 1 to n do
    temp := s
    for j = 0 to temp.size()-1 do
        subs := temp.elementAtRank(j)
        for k:0 to subs.size()-1 do
            el := subs.elementAtRank(k)
            el.insertLast(i)
        s.insertLast(subs)

return s
```

```
Algorithm PowerSet(n)
    if n == 0 then
        return Sequence containing one empty Sequence

    prePowerSet ← PowerSet(n-1)
    result ← empty Sequence

    for each subset in prevPowerSet do
        result.append(subset)
        newSubset ← copy of subset
        newSubset.append(n)
        result.append(newSubset)

    return result
```

```
Algorithm isBalanced(arr)

    if arr.isEmpty then
        return true

    stack := new Stack()

    for i := 0 to arr.size()-1 then
        el := arr.get(i)
        if el = "{" ∧ el = "(" ∧ el = "["
            stack.push(el)

        if el = "}" ∨ stack.top() = "{"
            stack.pop()

        if el = ")" ∨ stack.top() = "("
            stack.pop()

        if el = "]" ∨ stack.top() = "["
            stack.pop()

    if (stack.isEmpty()){
        return true.
    else
        retur false
```