Basic Settings : 사용할 때는 무시하세요

In [1]:
```
# !pip install matplotlib
# !pip install forallpeople
```

In [2]:
```
import csv
from dataclasses import dataclass, field, asdict, astuple
from typing import List
from collections import namedtuple
from functools import reduce
from functools import partial
# import forallpeople as si ## 단위 변환 관련 패키지(아직 미사용)
import math
import matplotlib.pyplot as plt
import numpy as np
# import matplotlib.transforms as transforms
```

In [3]:
```
##################################Module For Functional Programing#####################
curry = lambda f: lambda a,*args: f(a, *args) if (len(args)) else lambda *args: f(

filter = curry(filter)
map = curry(map)

def _take(length, iter):
    res = []
    for a in iter:
        res.append(a)
        if len(res) == length:
            return res

take = curry(_take)
reduce = curry(reduce)

go = lambda *args: reduce(lambda a,f: f(a), args) ## 함수도 축약 가능 ##
########################################################################################
```

In [4]:
```
## 함수형 프로그래밍 코드 사용 예제

add = curry(lambda a,b: a + b)

example = lambda _list: goW
(_list,
    filter(lambda a: not a % 2),
    filter(lambda a: not a % 4),
    map(lambda a: a ),
    reduce(add)
)
```

In [5]:
```
example([1,2,3,4,5,6])
```

Out[5]: 4

Definition Code Part

In [6]:
```
## 사용 예시
#########################################################################################################
    ####### 공통 입력부 #######
input_path = 'Section Profile.csv'

input_dsgnMode = "LRFD"

input_fy = 344.738 ## 50 ksi
input_fu = 448.159 ## 65 ksi
input_E = 199900 ## 29000

# input_DL = 6.567  ## 휨 검토때
# input_LL = 10.945
# input_length = 10670

input_DL = 622.751 ##140 kips ## 압축 검토때
input_LL = 1868.253 ## 420 kips

input_Pu = 1779 ## kN (400 kip)
input_Mux = 338.954 ## kN*m
input_Muy = 108.465 ## kN*m
input_length = 4267 ## 14 ft

    ####### 휨 입력부 #######
input_cb_mode = "Cb고려"
input_table_mode = "continuous"
input_brace_idx = 2

    ####### 압축 입력부 #######
input_Comp_mode = "recommended" ### 압축 유효길이 팩터 테이블 산정 모드
input_cond = "d"  ### 압축 유효길이 팩터 테이블 상 지점 조건
input_effLength = [4267] ### 압축 유효길이

    ####### 인장 입력부 #######
input_dia_bolt=20
input_length_bolt=228
input_n_bolt=4
input_gap_bolt=2

# input_dia_bolt = input_length_bolt = input_n_bolt = input_gap_bolt = None

#########################################################################################################
```

0. 단면 정보 import (선언부)

In [7]:
```
def importCSV(_path):
    with open(_path,'r') as f:
        reader  = csv.reader(f) # csv의 행별로 읽어옴
        res = [x for x in reader]

    return res
```

In [8]:
```
def exportCSV(_filename, _datas):
    f = open(_filename, 'w', newline='') # 자동줄바꿈 방지 header 이후 첫행 빈행 생성
    csv_writer = csv.writer(f)

    for x in _datas:
```

```
        csv_writer.writerow(x)
    f.close()
```

In [9]:
```
dfSectionProfile = importCSV(input_path)
dfHeader = dfSectionProfile.pop(0) #pop(0)의 의미 첫행을 값을 반환하고 나머지 리스트는
```

1. 단면 자료형 Setter (선언부)

In [10]:
```
def makeSectionForm(args):

    Form = namedtuple('SectionForm', ['ID','shape','h','bf','tw','tf','k'])
    form = W
        Form(ID=args[0], shape=args[1],
            h=float(args[2]), bf=float(args[3]),
            tw=float(args[4]), tf=float(args[5]), k=float(args[6]))

    return form


### 예시 ###
# sForm = makeSectionForm(dfSectionProfile[294])
# sForm

# targetSection = makeSectionForm(dfSectionProfile[160])
_targetSection = list(filter(lambda x: makeSectionForm(x).ID == "W14X99", dfSectionP
targetSection = makeSectionForm(_targetSection)
targetSection
```

Out[10]: SectionForm(ID='W14X99', shape='W', h=361.0, bf=371.0, tw=12.3, tf=19.8, k=35.1)

2. 단면 속성 Setter (선언부)

In [47]:
```
def setSectionProp(sForm):

    def defineC():
        if _shape == "W" or _shape == "H": return 1
        elif _shape == "C": return (_h0/2) * (_Iy/_Cw)**0.5
        else: return 1 ## 임시 대처


    Prop = namedtuple(
        'SectionProperty',
        ['ID', 'h', 'bf', 'tw', 'tf', 'k', 'shape', 'Area', 'Weight',
         'Ix', 'Sx', 'Zx', 'rx', 'Iy', 'Sy', 'Zy', 'ry',
         'Cw', 'J', 'rts', 'h0', 'SInd_Flange', 'SInd_Web', 'C'])

    def chkSlenderness(shape, mode):
        if _shape == 'H' or 'W':
            if mode == 'flange':
                if _bf/(2*_tf) < 0.56*(input_E/input_fy)**0.5:
                    return "nonslender"
                else:
                    return "slender"
```

```
            elif mode == 'web':
                if (_h-2*_k)/_tw < 1.49*(input_E/input_fy)**0.5:
                    return "nonslender"
                else:
                    return "slender"
            else:
                pass

        elif _shape == 'BH':
                if mode == 'flange':
                    _kc = min(4/(_h/_tw)**0.5, 0.35)
                    if _bf/(2*_tf) < 0.64*(_kc*input_E/input_fy)**0.5:
                        return "nonslender"
                    else:
                        return "slender"

                elif mode == 'web':
                    if (_h-2*_self.k)/_tw < 1.49*(input_E/input_fy)**0.5:
                        return "nonslender"
                    else:
                        return "slender"
                else:
                    pass

    (_ID,_h,_bf,_tw,_tf,_k,_shape) = (sForm.ID,sForm.h,sForm.bf,sForm.tw,sForm.tf,sFo
    _Area = 2*_tf*_bf+(_h-2*_tf)*_tw
    _Weight = _Area*77.22/10**6
    _Ix = (2*(_bf*_tf**3/12+_bf*_tf*((_h-2*_tf)/2+_tf/2)**2)+_tw*(_h-2*_tf)**3/12)
    _Sx = _Ix/(_h/2)
    _Zx = _bf*_tf*(_h-_tf)+0.25*(_h-2*_tf)**2*_tw
    _rx = (_Ix/_Area)**0.5
    _Iy = 2*(_tf*(_bf)**3/12)+(_h-2*_tf)*(_tw)**3/12
    _Sy = _Iy/(_bf/2)
    _Zy = 0.5*(_bf)**2*_tf+0.25*(_h-2*_tf)*(_tw)**2
    _ry = (_Iy/_Area)**0.5
    _Cw = (_h-_tf)**2*_bf**3*_tf/24
    _SInd_Flange = chkSlenderness(_shape, 'flange')
    _SInd_Web = chkSlenderness(_shape, 'web')
    (_J,_rts,_h0,_C) = ((2*_bf*_tf**3+(_h-_tf)*_tw**3)/3, ((_Iy*_Cw)**0.5/_Sx)**0.5


    prop = Prop(
        ID= _ID, h= _h, bf= _bf, tw= _tw, tf= _tf, k= _k, shape= _shape, Area= _Area,
        Ix= _Ix, Sx= _Sx, Zx= _Zx, rx= _rx, Iy= _Iy, Sy= _Sy, Zy= _Zy, ry= _ry,
        Cw= _Cw, J= _J, rts= _rts, h0= _h0, SInd_Flange= _SInd_Flange, SInd_Web= _SIn

    return prop

### 예시 ###
sProp = setSectionProp(targetSection)
sProp
```

Out[47]:
```
SectionProperty(ID='W14X99', h=361.0, bf=371.0, tw=12.3, tf=19.8, k=35.1, shape='W', A
rea=18644.82, Weight=1.4397530004, Ix=462099553.4006, Sx=2560108.329089197, Zx=282402
8.187, rx=157.430428275296, Iy=168563716.52115002, Sy=908699.2804374665, Zy=1374802.05
15, ry=95.08301657417651, Cw=4904488520830.668, J=2131540.8948000004, rts=105.97667362
617395, h0=341.2, SInd_Flange='nonslender', SInd_Web='nonslender', C=1)
```

3. 재료 속성 Setter (선언부)

In [12]:
```
def setMaterialProp(_fy, _fu, _E):
    Prop = namedtuple('MaterialProperty', ['fy', 'fu', 'E'])
    prop = Prop(fy=_fy, fu=_fu, E=_E)

    return prop

### 예시 ###
mProp = setMaterialProp(344.738, 448.159, 199900)
mProp
```

Out[12]: MaterialProperty(fy=344.738, fu=448.159, E=199900)

4-1. 프레임 해석 // 2023년도 구현 예정 //

In [13]:
```
def setRequiredStrength(_DL, _LL, _Pu, _Mux, _Muy):
    def calcPu(_DL, _LL): ## 추후 구현
        pass
    def calcMu(_DL, _LL): ## 추후 구현
        pass
    _Load = namedtuple('LoadInform', ['DL', 'LL'])
    RqStr = namedtuple('requiredStrength', ['Load', 'Pu', 'Mux', 'Muy'])

    return RqStr(Load= _Load(DL=_DL, LL=_LL), Pu=_Pu, Mux=_Mux, Muy=_Muy)

rqStr = setRequiredStrength(input_DL, input_LL, input_Pu, input_Mux, input_Muy)
rqStr
```

Out[13]: requiredStrength(Load=LoadInform(DL=622.751, LL=1868.253), Pu=1779, Mux=338.954, Muy=108.465)

4-2. 디자인 베이스 Setter (선언부)

In [14]:
```
def setDesignBase(_dsgnMode, _rqStr, _length, _useMode):

    Base = namedtuple('DesignBase', ['DL', 'LL', 'length', 'Pr', 'Mrx', 'Mry'])

    def calcRequired(_dsgnMode):
        _DL = _rqStr.Load.DL
        _LL = _rqStr.Load.LL
        def _calcStr():
            if _dsgnMode == "LRFD":
                result = 1.2*_DL + 1.6*_LL
            elif _dsgnMode == "ASD":
                result = _DL + _LL
            else:
                result = "check the DesignMode"
            return result

        def _calcMoment():
            result = ((_calcStr() * _length**2) / 8) / 1000**2
            return result

        return (_calcStr(), _calcMoment())

    if _useMode == "useLoad":
        base = Base(
```

```
            DL=_rqStr.Load.DL, LL=_rqStr.Load.LL,
            length=_length, Pr=_calcRequired(_dsgnMode)[0], Mrx=calcRequired(_dsgnMod
    elif _useMode == "usePuMu":
        base = Base(
            DL=_rqStr.Load.DL, LL=_rqStr.Load.LL,
            length=_length, Pr=_rqStr.Pu, Mrx=_rqStr.Mux, Mry=_rqStr.Muy)

    return base

### 예시 ###
dBase = setDesignBase(input_dsgnMode, rqStr, input_length, "usePuMu")
dBase
```

Out[14]: DesignBase(DL=622.751, LL=1868.253, length=4267, Pr=1779, Mrx=338.954, Mry=108.465)

5. 서브 디자인 베이스 Setter (선언부)

In [15]:

```
def setFlexureBase(cb_mode, table_mode, _brace_idx):
    Base = namedtuple('SubBase_flx', ['brace_idx', 'Cb'])

    def findCb():
        _none = {
            "1p": [[1.32]],
            "2p": [[1.14]],
            "3p": [[1.14]],
            "continuous": [[1.14]] }

        _atLoad = {
            "1p": [1.67, 1.67],
            "2p": [1.67, 1.00, 1.67],
            "3p": [1.67, 1.11, 1.11, 1.67],
            "continuous":
                [[1.30, 1.30],
                [1.45, 1.01, 1.45],
                [1.52, 1.06, 1.06, 1.52],
                [1.56, 1.12, 1.00, 1.12, 1.56]] }
        if cb_mode == "Cb고려":
            if _brace_idx == 0:
                result = _none[table_mode][0]#[_brace_idx]
            else:
                if table_mode == "continuous":
                    result = _atLoad[table_mode][_brace_idx-1]
                else:
                    result = _atLoad[table_mode]
        elif cb_mode == "Cb미고려":
            result = [ 1.00 ]

        return result

    base = Base(brace_idx= _brace_idx, Cb=findCb())
    return base

### 예시 ###
fBase = setFlexureBase("Cb고려", "continuous", 2)
fBase
```

Out[15]: SubBase_flx(brace_idx=2, Cb=[1.45, 1.01, 1.45])

In [16]:

```
def setCompressureBase(mode, cond, *_length):
    Base = namedtuple('SubBase_Comp', ['unbracedLength_x', 'unbracedLength_y', 'factor
```

```
    def EffectiveLength():

        if len(_length) == 1:
            _unbracedLength_x = _length[0]
            _unbracedLength_y = _length[0]
        elif len(_length) == 2:
            _unbracedLength_x = _length[0]
            _unbracedLength_y = _length[1]

        EffectiveLength = [_unbracedLength_x, _unbracedLength_y]
        return EffectiveLength

    def factorK():
        _factorTable = {
            "a": (0.5, 0.65),
            "b": (0.7, 0.80),
            "c": (1.0, 1.2),
            "d": (1.0, 1.0),
            "e": (2.0, 2.1),
            "f": (2.0, 2.0),
        }

        if mode == "theoretical":
            factorK = _factorTable[cond][0]

        elif mode == "recommended":
            factorK = _factorTable[cond][1]

        else:
            factorK = 'Please select mode("theoretical" or "recommended")'

        return factorK

    base = Base(unbracedLength_x= EffectiveLength()[0], unbracedLength_y= EffectiveLe
    return base

### 예시 ###
cBase = setCompressureBase("recommended", "d", 4267)
cBase
```

Out[16]:
```
SubBase_Comp(unbracedLength_x=4267, unbracedLength_y=4267, factorK=1.0)
```

In [17]:
```
def setTensileBase(_dia_bolt, _len_bolt, _n_bolt, _gap_bolt):

    Base = namedtuple('SubBase_Tensile', ['dia_bolt', 'length_bolt', 'n_bolt', 'gap_bo
    base = Base(dia_bolt=_dia_bolt, length_bolt=_len_bolt, n_bolt=_n_bolt, gap_bolt=_

    return base

### 예시 ###
tBase = setTensileBase(20, 228, 4, 2)
tBase
```

Out[17]:
```
SubBase_Tensile(dia_bolt=20, length_bolt=228, n_bolt=4, gap_bolt=2)
```

In [18]:
```
def setCombinedBase(_cBase, _fBase):
    Base = namedtuple('SubBase_Combined', ['cBase', 'fBase'])
    return Base(cBase=_cBase, fBase=_fBase)

### 예시 ###
```

```
comBase = setCombinedBase(cBase, fBase)
comBase
```

Out[18]:
```
SubBase_Combined(cBase=SubBase_Comp(unbracedLength_x=4267, unbracedLength_y=4267, fact
orK=1.0), fBase=SubBase_flx(brace_idx=2, Cb=[1.45, 1.01, 1.45]))
```

6.a 서브 디자인 결과 Checker_휨 (선언부)

In [19]:
```
def checkFlexure(_dsgnMode, _sProp, _mProp, _dBase, _subBase):
    ResultForm = namedtuple('flexResult', ['Mcx', 'Mcy'])

    def findLp():
        return 1.76 * _sProp.ry * (_mProp.E/_mProp.fy)**0.5

    def findLr():
        return 1.95*_sProp.rts*_mProp.E/(0.7*_mProp.fy)*(_sProp.J/(_sProp.Sx*(_sPro

    def findMp():
        ### for Strong Axis ###
        Mp_x = _mProp.fy * _sProp.Zx
        ### for Weak Axis ###
        Mp_y = min([(_mProp.fy * _sProp.Zy), (1.6*_mProp.fy*_sProp.Sy)])

        return {"Mp_x": Mp_x, "Mp_y": Mp_y}

    def findMn(): ### for Strong Axis ###
        (Mp,Lp,Lr,Lb) = (findMp()["Mp_x"], findLp(), findLr(), _dBase.length/(_subBas

        if Lb <= Lp:
            Mn = Mp
        elif Lp < Lb <= Lr:
            Mn = min(map(lambda x: x * (Mp-(Mp-0.7*_mProp.fy*_sProp.Sx)*((Lb-Lp) /
        elif Lb > Lr:
            Fcr = min(map(lambda x: (x * (math.pi**2 * _mProp.E)/((Lb/_sProp.rts)**
            Mn = min((Fcr)*_sProp.Sx, Mp)

        return Mn / 1000**2

    def findMn_weekAxis():
        Mn = findMp()["Mp_y"] ## 약축 휨 Buckling 무시: 해당 부재 많지 않고, 적용하는
        return Mn / 1000**2


    def calcNominal(_Mn):
        if _dsgnMode == "LRFD":
            result = 0.90 * _Mn
        elif _dsgnMode == "ASD":
            result = _Mn / 1.67
        else:
            result = "check the DesignMode"

        return result

    return ResultForm(Mcx=calcNominal(findMn()), Mcy=calcNominal(findMn_weekAxis()))

### 예시
fCheck = checkFlexure(input_dsgnMode, sProp, mProp, dBase, fBase)
fCheck
```

Out[19]:
```
flexResult(Mcx=876.1948462170054, Mcy=426.55185866700634)
```

6.b 서브 디자인 결과 Checker_압축 (선언부)

In [20]:

```
def checkCompressure(_dsgnMode, _sProp, _mProp, _dBase, _subBase):

    def find_Lc():
        Lc_x = _subBase.factorK * _subBase.unbracedLength_x
        Lc_y = _subBase.factorK * _subBase.unbracedLength_y
        return (Lc_x, Lc_y)

    def find_r(): ## 회전 반경
        rx = _sProp.rx
        ry = _sProp.ry
        return (rx, ry)

    def find_Lc_r(): ## 세장비 (effective slenderness ratio)
        Lc_rx = find_Lc()[0]/find_r()[0]
        Lc_ry = find_Lc()[1]/find_r()[1]

        def chk_IsLcExceed():
            if Lc_rx < 200 and Lc_ry < 200:
                return (Lc_rx, Lc_ry)
            else:
                return (Lc_rx, Lc_ry, "ratio exceed 200")

        return chk_IsLcExceed()

    def find_Fe(): ## 탄성 좌굴 응력
        r = find_r()
        Lc_rx = find_Lc_r()[0]
        Lc_ry = find_Lc_r()[1]

        Fe_x = math.pi**2*_mProp.E/Lc_rx**2
        Fe_y = math.pi**2*_mProp.E/Lc_ry**2

        return (Fe_x, Fe_y)

    def find_Fcr(): ## 좌굴임계응력

        Fe_x = find_Fe()[0]
        Fe_y = find_Fe()[1]

        if find_Lc_r()[0] > 4.71*(_mProp.E/_mProp.fy)**0.5:
            Fcr_x = 0.877*Fe_x
        else:
            Fcr_x = 0.658**(_mProp.fy/Fe_x)*_mProp.fy

        if find_Lc_r()[1] > 4.71*(_mProp.E/_mProp.fy)**0.5:
            Fcr_y = 0.877*Fe_y
        else:
            Fcr_y = 0.658**(_mProp.fy/Fe_y)*_mProp.fy

        return min(Fcr_x, Fcr_y)

    def find_slender_web():

        if _sProp.shape == "H" or "W" or "BH":
            c1 = 0.18
        else:
            c1 = 0.22

        c2 = (1-(1-4*c1)**0.5)/(2*c1)
        h = _sProp.h-(2*_sProp.tf)
        λ = _sProp.h/_sProp.tw
```

```
        λr = 1.49*(_mProp.E/_mProp.fy)**0.5
        Fel = (c2*λr/λ)**2 * _mProp.fy

        if λ<=λr*(_mProp.fy/find_Fcr())**0.5:
            he_effh = _sProp.h
        else:
            he_effh = _sProp.h*(1-c1*(Fel/find_Fcr())**0.5)*(Fel/find_Fcr())**0.5

        return he_effh


    def find_slender_flange(): ###### slender 부재일경우
        shape = _sProp.shape

        kc = 4/(_sProp.h/_sProp.tw)**0.5
        λ = _sProp.bf/(2*_sProp.tf)
        b = _sProp.bf/2


        if shape == "H" or "W" or "BH":
            if shape == "H" or "W":
                c1 = 0.18
                λr = 0.56*(kc*_mProp.E/_mProp.fy)**0.5

            elif Shape == "BH":
                c1 = 0.18
                λr = 0.64*(kc*_mProp.E/_mProp.fy)**0.5

        else:
            c1 = 0.22

        c2 = (1-(1-4*c1)**0.5)/(2*c1)
        Fel = (c2*λr/λ)**2 * _mProp.fy

        if λ<=λr*(_mProp.fy/find_Fcr())**0.5:
            be_effb = b

        else:
            be_effb = b*(1-c1*(Fel/find_Fcr())**0.5)*(Fel/find_Fcr())**0.5

        return be_effb

    def find_Ae():
        SInd_Web = _sProp.SInd_Web
        SInd_Flange = _sProp.SInd_Flange
        he_effh = find_slender_web()
        be_effb = find_slender_flange()

        if SInd_Web == "nonslender" and SInd_Flange == "nonslender":
            Ae = _sProp.Area
        else:
            Ae = he_effh*_sProp.tw + 2*_sProp.tf*(2*be_effb)
        return Ae

    def find_Area_ratio():
        Area_ratio = find_Ae()/_sProp.Area
        return Area_ratio

    def find_Pn(): ## 공칭 강도 (nominal compressive strength)
        return find_Fcr() * find_Ae() / 1000

    def calcNominal():
        if _dsgnMode == "LRFD":
            result = 0.90 * find_Pn()
```

```
        elif _dsgnMode == "ASD":
            result = find_Pn() / 1.67
        else:
            result = "check the DesignMode"

        return result

    return calcNominal()

### 예시
cCheck = checkCompressure(input_dsgnMode, sProp, mProp, dBase, cBase)
cCheck
```

Out[20]: 4992.568533077785

6.c 서브 디자인 결과 Checker_인장 (선언부)

In [21]:

```
def checkTensile(_dsgnMode, _sProp, _mProp, _dBase, _subBase):
    ( d, bf, tw, tf, Ag ) = ( _sProp.h, _sProp.bf, _sProp.tw, _sProp.tf, _sProp.Area
    if all([_subBase.dia_bolt, _subBase.length_bolt, _subBase.n_bolt, _subBase.gap_bo
        ( db, boltN, length_bolt, gap ) = ( _subBase.dia_bolt, _subBase.n_bolt, _subB
    else: pass

    def findAn():
        An = Ag - boltN*(db + gap)*tf
        return An

    def findU_forShealag():
        l = length_bolt
        x = (bf*tf*tf/2 + (d/2 - tf)*tw*((d/2-tf)/2 + tf)) / (bf*tf + (d/2 - tf)*tw
        u1 = (2*bf*tf) / Ag

        u2 = 1 - (x/l)

        if boltN >= 3:
            if bf < 2/3*d:
                u3 = 0.85
            else:
                u3 = 0.9
        else:
            u3 = 0

        return max(u1, u2, u3)

    def calc_Ae():
        U = findU_forShealag()
        return findAn()*U

    def find_Pn():
        ( Fy, Fu ) = ( _mProp.fy, _mProp.fu )
        Result = namedtuple("Result",["Fy_Ag", "Fu_Ae"])
        unitModif = 1000**1

        if all([_subBase.dia_bolt, _subBase.length_bolt, _subBase.n_bolt, _subBase.ga
            Ae = calc_Ae() /unitModif
            result = Result(Fy_Ag=Fy*Ag, Fu_Ae=Fu*Ae)
            return result
        else:
            Ae = 0.75 * Ag /unitModif
            result = Result(Fy_Ag=Fy*Ag, Fu_Ae=Fu*Ae)
            return result
```

```
    def calcNominal_tensile():
        ( Φ1, Φ2 ) = ( 0.9, 0.75 )
        ( Ω1, Ω2 ) = ( 1.67, 2.00 )

        if _dsgnMode == "LRFD":
            result = min(Φ1 * find_Pn().Fy_Ag, Φ2*find_Pn().Fu_Ae)
        elif _dsgnMode == "ASD":
            result = min(find_Pn().Fy_Ag/Ω1, Φ2*find_Pn().Fu_Ae/Ω2)
        else:
            result = "check the DesignMode"

        return result

    return calcNominal_tensile()
```

6.d 서브 디자인 결과 Checker_복합력 (선언부)

In [22]:

```
def checkCombined(_dsgnMode, _sProp, _mProp, _dBase, _subBase, _fCheck, _cCheck):
    ( Pr, Mrx, Mry ) = ( _dBase.Pr, _dBase.Mrx, _dBase.Mry )
    Pc = _cCheck(_dsgnMode, _sProp, _mProp, _dBase, _subBase.cBase)

    def findMcx():
        return _fCheck(_dsgnMode, _sProp, _mProp, _dBase, _subBase.fBase).Mcx


    def findMcy():
        return _fCheck(_dsgnMode, _sProp, _mProp, _dBase, _subBase.fBase).Mcy
    ( Mcx, Mcy ) =  ( findMcx(), findMcy() )
    def calcRatio_combined():
        ( Mcx, Mcy ) =  ( findMcx(), findMcy() )
        if (Pr/Pc) >= 0.2:
            res = (Pr/Pc) + (8/9)*((Mrx/Mcx) + (Mry/Mcy))
        elif (Pr/Pc) < 0.2:
            res = (Pr/(2*Pc)) + ((Mrx/Mcx) + (Mry/Mcy))
        else:
            pass

        return res

    return calcRatio_combined()

### 예시 ###
comCheck = checkCombined(input_dsgnMode, sProp, mProp, dBase, comBase, checkFlexure, c
comCheck
```

Out[22]: 0.9262238424378537

In [23]:

```
mkSubBaseColl = namedtuple('subBaseColl', 'fBase, cBase, tBase, comBase')

fBase = setFlexureBase(input_cb_mode, input_table_mode, input_brace_idx)
cBase = setCompressureBase(input_Comp_mode, input_cond, *input_effLength)
tBase = setTensileBase(input_dia_bolt, input_length_bolt, input_n_bolt, input_gap_bolt
comBase = setCombinedBase(cBase, fBase)

subBaseColl = mkSubBaseColl(fBase=fBase, cBase=cBase, tBase=tBase, comBase=comBase)
subBaseColl

mkSubCheckColl = namedtuple('subCheckColl', 'fCheck, cCheck, tCheck, comCheck')
```

```
subCheckColl = mkSubCheckColl(fCheck=checkFlexure, cCheck=checkCompressure, tCheck=ch

subCheckColl
```

Out[23]:
```
subCheckColl(fCheck=<function checkFlexure at 0x7f6c14b35e50>, cCheck=<function checkC
ompressure at 0x7f6c14ac1a60>, tCheck=<function checkTensile at 0x7f6c14b21ca0>, comCh
eck=<function checkCombined at 0x7f6c14b194c0>)
```

7. 디자인 결과 Checker (선언부)

In [24]:
```
def checkDesignResult(_dsgnMode, _sProp, _mProp, _dBase, _subBaseColl, _subCheckColl)

    ( Pr, Mrx, Mry ) = ( _dBase.Pr, _dBase.Mrx, _dBase.Mry )

    ( fCheck, fBase ) = ( _subCheckColl.fCheck, _subBaseColl.fBase )
    ( cCheck, cBase ) = ( _subCheckColl.cCheck, _subBaseColl.cBase )
    ( tCheck, tBase ) = ( _subCheckColl.tCheck, _subBaseColl.tBase )
    ( comCheck, comBase ) = ( _subCheckColl.comCheck, _subBaseColl.comBase )

    def findDesignResult(_dsgnMode, _sProp, _mProp, _dBase, _subBase, _subCheck):
        FormResult = namedtuple("resForm", "checkName, ID, NomStr, ReqStr, Result")
        checkName = _subCheck.__name__
        if checkName == "checkFlexure":
            nomStr = _subCheck(_dsgnMode,_sProp, _mProp, _dBase, _subBase)
            if nomStr.Mcx > Mrx:
                return FormResult(
                            checkName = _subCheck.__name__,
                            ID = _sProp.ID,
                            NomStr = nomStr.Mcx,
                            ReqStr = Mrx,
                            Result = "O.K.")
            else:
                return FormResult(
                            checkName = _subCheck.__name__,
                            ID = _sProp.ID,
                            NomStr = nomStr.Mcx,
                            ReqStr = Mrx,
                            Result = "N.G.")

        elif checkName == "checkCombined":
            nomStr = _subCheck(_dsgnMode,_sProp, _mProp, _dBase, _subBase, fCheck, cCh
            reqStr = 1.0
            if nomStr <= reqStr:
                return FormResult(
                            checkName = _subCheck.__name__,
                            ID = _sProp.ID,
                            NomStr = nomStr,
                            ReqStr = reqStr,
                            Result = "O.K.")
            else:
                return FormResult(
                            checkName = _subCheck.__name__,
                            ID = _sProp.ID,
                            NomStr = nomStr,
                            ReqStr = reqStr,
                            Result = "N.G.")

        else:
            nomStr = _subCheck(_dsgnMode,_sProp, _mProp, _dBase, _subBase)
            if nomStr > Pr:
                return FormResult(
                            checkName = _subCheck.__name__,
```

```
                            ID = _sProp.ID,
                            NomStr = nomStr,
                            ReqStr = Pr,
                            Result = "O.K.")
            else:
                return FormResult(
                            checkName = _subCheck.__name__,
                            ID = _sProp.ID,
                            NomStr = nomStr,
                            ReqStr = Pr,
                            Result = "N.G.")


    result_flx = findDesignResult(_dsgnMode, _sProp, _mProp, _dBase, fBase, fCheck)
    result_comp = findDesignResult(_dsgnMode, _sProp, _mProp, _dBase, cBase, cCheck)
    result_tensile = findDesignResult(_dsgnMode, _sProp, _mProp, _dBase, tBase, tCheck
    result_combined = findDesignResult(_dsgnMode, _sProp, _mProp, _dBase, comBase, con

    FinalForm = namedtuple("FinalForm", "result_flx, result_comp, result_tensile, resu

    return FinalForm(result_flx=result_flx, result_comp=result_comp, result_tensile=r

### 예시
kk = checkDesignResult(input_dsgnMode, sProp, mProp, dBase, subBaseColl, subCheckColl)
kk
```

Out[24]:
```
FinalForm(result_flx=resForm(checkName='checkFlexure', ID='W14X99', NomStr=876.1948462
170054, ReqStr=338.954, Result='O.K.'), result_comp=resForm(checkName='checkCompressur
e', ID='W14X99', NomStr=4992.568533077785, ReqStr=1779, Result='O.K.'), result_tensile
=resForm(checkName='checkTensile', ID='W14X99', NomStr=5113.105860226499, ReqStr=1779,
Result='O.K.'), result_combined=resForm(checkName='checkCombined', ID='W14X99', NomStr
=0.9262238424378537, ReqStr=1.0, Result='O.K.'))
```

In [25]:
```
## 결과 출력용 함수 ##
def showResult(ress, _checkMode):
    def mkResult(res):
        if res.NomStr > res.ReqStr: ineqSign = '>'
        else: ineqSign = '<'

        return f"for {res.checkName},,, {res.ID} - nominal: '{res.NomStr:.2f}'  {ineq
    ### 결과에 단위 포함 출력은 차주 구현 예정

    if _checkMode == "all":
        result = list(map(mkResult, ress))

    else:
        result = list(map(mkResult,
                          filter(lambda x: x.checkName == _checkMode, ress)))
    return result
```

In [26]:
```
showResult(kk, "checkCombined")
```

Out[26]:
```
["for checkCombined,,, W14X99 - nominal: '0.93'  < required: '1.00'  -> O.K."]
```

Client Code Part

<<< 단일 부재 검토 Mode >>>

Given:

Select a W-shape beam for span and uniform dead and live loads as shown in Figure F.1-1A. Limit the member to a maximum nominal depth of 18 in. Limit the live load deflection to $L/360$. The beam is simply supported and continuously braced. The beam is ASTM A992 material.

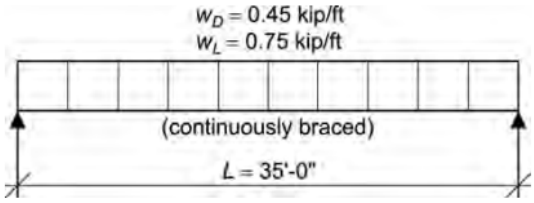


Fig. F.1-1A. Beam loading and bracing diagram.

Solution:

From AISC *Manual* Table 2-4, the material properties are as follows:

ASTM A992
F_y = 50 ksi
F_u = 65 ksi

------- 사용자 입력부 -------

In [27]:

```
## 사용 예시
#########################################################################################################
    ####### 공통 입력부 #######
input_path = 'Section Profile.csv'

input_dsgnMode = "LRFD"

input_fy = 344.738 ## 50 ksi
input_fu = 448.159 ## 65 ksi
input_E = 199900 ## 29000

# input_DL = 6.567  ## 휨 검토때
# input_LL = 10.945
# input_length = 10670

input_DL = 622.751 ##140 kips ## 압축 검토때
input_LL = 1868.253 ## 420 kips

input_Pu = 1779 ## kN (400 kip)
input_Mux = 338.954 ## kN*m
input_Muy = 108.465 ## kN*m
input_length = 4267 ## 14 ft

    ####### 휨 입력부 #######
input_cb_mode = "Cb고려"
input_table_mode = "continuous"
input_brace_idx = 2

    ####### 압축 입력부 #######
input_Comp_mode = "recommended" ### 압축 유효길이 팩터 테이블 산정 모드
input_cond = "d"  ### 압축 유효길이 팩터 테이블 상 지점 조건
input_effLength = [4267] ### 압축 유효길이

    ####### 인장 입력부 #######
input_dia_bolt=20
input_length_bolt=228
```

```
input_n_bolt=4
input_gap_bolt=2

# input_dia_bolt = input_length_bolt = input_n_bolt = input_gap_bolt = None

#########################################################################################
```

---

0. 단면 정보 import (사용부)

In [48]:
```
dfSectionProfile = importCSV(input_path)
dfSectionProfile.remove(dfSectionProfile[0])
dfSectionProfile[0]
```

Out[48]:
```
['W44X335', 'W', '1120', '404', '26.2', '45', '65']
```

1. 단면 자료형 Setter (사용부)

In [29]:
```
# targetSection = makeSectionForm(dfSectionProfile[160])
_targetSection = list(filter(lambda x: makeSectionForm(x).ID == "W14X99", dfSectionP
targetSection = makeSectionForm(_targetSection)
targetSection
```

Out[29]:
```
SectionForm(ID='W14X99', shape='W', h=361.0, bf=371.0, tw=12.3, tf=19.8, k=35.1)
```

----------중략----------

2. 단면 속성 Setter (사용부)

In [30]:
```
sProp = setSectionProp(targetSection)
sProp
```

Out[30]:
```
SectionProperty(ID='W14X99', h=361.0, bf=371.0, tw=12.3, tf=19.8, k=35.1, shape='W', A
rea=18644.82, Weight=1.4397530004, Ix=462099553.4006, Sx=2560108.329089197, Zx=282402
8.187, rx=157.430428275296, Iy=168563716.52115002, Sy=908699.2804374665, Zy=1374802.05
15, ry=95.08301657417651, Cw=4904488520830.668, J=2131540.8948000004, rts=105.97667362
617395, h0=341.2, SInd_Flange='nonslender', SInd_Web='nonslender', C=1)
```

3. 재료 속성 Setter (사용부)

In [31]:
```
mProp = setMaterialProp(input_fy, input_fu, input_E)
mProp
```

Out[31]:
```
MaterialProperty(fy=344.738, fu=448.159, E=199900)
```

4. 디자인 베이스 Setter (사용부)

In [32]:
```
rqStr = setRequiredStrength(input_DL, input_LL, input_Pu, input_Mux, input_Muy)

dBase = setDesignBase(input_dsgnMode, rqStr, input_length, "usePuMu")
dBase
```

Out[32]:
```
DesignBase(DL=622.751, LL=1868.253, length=4267, Pr=1779, Mrx=338.954, Mry=108.465)
```

5. 서브 디자인 베이스 Setter (사용부)

In [33]:
```
mkSubBaseColl = namedtuple('subBaseColl', 'fBase, cBase, tBase, comBase')

fBase = setFlexureBase(input_cb_mode, input_table_mode, input_brace_idx)
cBase = setCompressureBase(input_Comp_mode, input_cond, *input_effLength)
tBase = setTensileBase(input_dia_bolt, input_length_bolt, input_n_bolt, input_gap_bolt
comBase = setCombinedBase(cBase, fBase)

subBaseColl = mkSubBaseColl(fBase=fBase, cBase=cBase, tBase=tBase, comBase=comBase)
subBaseColl
```

Out[33]:
```
subBaseColl(fBase=SubBase_flx(brace_idx=2, Cb=[1.45, 1.01, 1.45]), cBase=SubBase_Comp
(unbracedLength_x=4267, unbracedLength_y=4267, factorK=1.0), tBase=SubBase_Tensile(dia
_bolt=20, length_bolt=228, n_bolt=4, gap_bolt=2), comBase=SubBase_Combined(cBase=SubBa
se_Comp(unbracedLength_x=4267, unbracedLength_y=4267, factorK=1.0), fBase=SubBase_flx
(brace_idx=2, Cb=[1.45, 1.01, 1.45])))
```

6.a, b, c. 서브 디자인 결과 Checker (사용부)

In [34]:
```
mkSubCheckColl = namedtuple('subCheckColl', 'fCheck, cCheck, tCheck, comCheck')

subCheckColl = mkSubCheckColl(fCheck=checkFlexure, cCheck=checkCompressure, tCheck=ch

subCheckColl
```

Out[34]:
```
subCheckColl(fCheck=<function checkFlexure at 0x7f6c14b35e50>, cCheck=<function checkC
ompressure at 0x7f6c14ac1a60>, tCheck=<function checkTensile at 0x7f6c14b21ca0>, comCh
eck=<function checkCombined at 0x7f6c14b194c0>)
```

---

6. 디자인 결과 Checker (사용부)

In [35]:
```
res = checkDesignResult(input_dsgnMode, sProp, mProp, dBase, subBaseColl, subCheckCol
res

# print(showResult(res, 'checkFlexure'))
# print(showResult(res, 'checkCompressure'))
# print(showResult(res, 'checkTensile'))
# print(showResult(res, 'checkCombined'))

showResult(res, 'all')
```

Out[35]:
```
["for checkFlexure,,, W14X99 - nominal: '876.19'  > required: '338.95'  -> O.K.",
 "for checkCompressure,,, W14X99 - nominal: '4992.57'  > required: '1779.00'  -> O.
K.",
 "for checkTensile,,, W14X99 - nominal: '5113.11'  > required: '1779.00'  -> O.K.",
 "for checkCombined,,, W14X99 - nominal: '0.93'  < required: '1.00'  -> O.K."]
```

<<< 다중 부재 검토 Mode >>>

------- 사용자 입력부 -------

In [36]:
```
## 사용 예시
##########################################################################################
```

```
    ####### 공통 입력부 #######
input_path = 'Section Profile.csv'

input_dsgnMode = "LRFD"

input_fy = 344.738 ## 50 ksi
input_fu = 448.159 ## 65 ksi
input_E = 199900 ## 29000

# input_DL = 6.567  ## 휨 검토때
# input_LL = 10.945
# input_length = 10670

input_DL = 622.751 ##140 kips ## 압축 검토때
input_LL = 1868.253 ## 420 kips

input_Pu = 1779 ## kN (400 kip)
input_Mux = 338.954 ## kN*m
input_Muy = 108.465 ## kN*m
input_length = 4267 ## 14 ft

    ####### 휨 입력부 #######
input_cb_mode = "Cb고려"
input_table_mode = "continuous"
input_brace_idx = 2

    ####### 압축 입력부 #######
input_Comp_mode = "recommended" ### 압축 유효길이 팩터 테이블 산정 모드
input_cond = "d"  ### 압축 유효길이 팩터 테이블 상 지점 조건
input_effLength = [4267] ### 압축 유효길이

    ####### 인장 입력부 #######
# input_dia_bolt=20
# input_length_bolt=228
# input_n_bolt=4
# input_gap_bolt=2

input_dia_bolt = input_length_bolt = input_n_bolt = input_gap_bolt = None

##########################################################################################
```

---

0 ~ 1. 다중 단면 자료형 세팅

In [37]:

```
## 여기서 makeSectionForms는 데이터를 저장하지 않고 함수 객체를 저장한다. C#, JAVA에서
makeSectionForms = lambda _list: goW
(
    _list,
    map(makeSectionForm),

list)
```

In [38]:

```
#### 0. 단면 정보 import
dfSectionProfile = importCSV(input_path)
dfSectionProfile.remove(dfSectionProfile[0])

#### 1. 단면 자료형 setter
targetSections = makeSectionForms(dfSectionProfile)
# targetSections
```

2 ~ 6. 각종 Setter들 및 Checker들 고정 조건 세팅

In [39]:
```
### 고정 조건 세팅

    #### 3. 재료속성
mProp = setMaterialProp(input_fy, input_fu, input_E)

    #### 4. 디자인 베이스 setter
rqStr = setRequiredStrength(input_DL, input_LL, input_Pu, input_Mux, input_Muy)

dBase = setDesignBase(input_dsgnMode, rqStr, input_length, "usePuMu")

    #### 5. 서브 디자인 베이스 setter
mkSubBaseColl = namedtuple('subBaseColl', 'fBase, cBase, tBase, comBase')

fBase = setFlexureBase(input_cb_mode, input_table_mode, input_brace_idx)
cBase = setCompressureBase(input_Comp_mode, input_cond, *input_effLength)
tBase = setTensileBase(input_dia_bolt, input_length_bolt, input_n_bolt, input_gap_bolt
comBase = setCombinedBase(cBase, fBase)

subBaseColl = mkSubBaseColl(fBase=fBase, cBase=cBase, tBase=tBase, comBase=comBase)

    #### 6.a~d 서브디자인 checker
mkSubCheckColl = namedtuple('subCheckColl', 'fCheck, cCheck, tCheck, comCheck')

subCheckColl = mkSubCheckColl(fCheck=checkFlexure, cCheck=checkCompressure, tCheck=ch


subBaseColl
```

Out[39]:
```
subBaseColl(fBase=SubBase_flx(brace_idx=2, Cb=[1.45, 1.01, 1.45]), cBase=SubBase_Comp
(unbracedLength_x=4267, unbracedLength_y=4267, factorK=1.0), tBase=SubBase_Tensile(dia
_bolt=None, length_bolt=None, n_bolt=None, gap_bolt=None), comBase=SubBase_Combined(cB
ase=SubBase_Comp(unbracedLength_x=4267, unbracedLength_y=4267, factorK=1.0), fBase=Sub
Base_flx(brace_idx=2, Cb=[1.45, 1.01, 1.45])))
```

디자인 결과 Checker 용 합성함수 제조

In [40]:
```
### 가변 조건 검토를 위한 함수 합성

multiChecker1 = lambda _list: goW
(
    _list,
    map(makeSectionForm),
    map(setSectionProp),
    map(lambda x: checkDesignResult(input_dsgnMode, x, mProp, dBase, subBaseColl, sub
    filter(lambda x: x.result_comp.Result == "O.K." ),
    map(lambda x: showResult(x, "checkCombined")),

list)
```

In [41]:
```
multiChecker1(targetSections)
```

Out[41]:
```
[["for checkCombined,, W44X335 - nominal: '0.19'  < required: '1.00'  -> O.K."],
 ["for checkCombined,, W44X290 - nominal: '0.22'  < required: '1.00'  -> O.K."],
 ["for checkCombined,, W44X262 - nominal: '0.24'  < required: '1.00'  -> O.K."],
 ["for checkCombined,, W44X230 - nominal: '0.28'  < required: '1.00'  -> O.K."],
 ["for checkCombined,, W40X655 - nominal: '0.09'  < required: '1.00'  -> O.K."],
 ["for checkCombined,, W40X593 - nominal: '0.10'  < required: '1.00'  -> O.K."],
```

```
["for checkCombined,,, W40X503 - nominal: '0.12'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X431 - nominal: '0.14'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X397 - nominal: '0.15'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X372 - nominal: '0.16'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X362 - nominal: '0.17'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X324 - nominal: '0.19'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X297 - nominal: '0.21'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X277 - nominal: '0.23'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X249 - nominal: '0.25'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X215 - nominal: '0.30'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X199 - nominal: '0.41'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X392 - nominal: '0.19'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X331 - nominal: '0.23'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X327 - nominal: '0.24'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X294 - nominal: '0.27'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X278 - nominal: '0.28'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X264 - nominal: '0.30'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X235 - nominal: '0.34'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X211 - nominal: '0.46'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X183 - nominal: '0.55'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X167 - nominal: '0.62'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W40X149 - nominal: '0.75'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X925 - nominal: '0.06'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X853 - nominal: '0.06'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X802 - nominal: '0.07'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X723 - nominal: '0.08'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X652 - nominal: '0.09'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X529 - nominal: '0.11'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X487 - nominal: '0.12'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X441 - nominal: '0.13'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X395 - nominal: '0.15'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X361 - nominal: '0.17'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X330 - nominal: '0.18'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X302 - nominal: '0.20'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X282 - nominal: '0.21'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X262 - nominal: '0.23'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X247 - nominal: '0.25'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X231 - nominal: '0.27'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X256 - nominal: '0.30'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X232 - nominal: '0.33'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X210 - nominal: '0.38'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X194 - nominal: '0.49'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X182 - nominal: '0.53'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X170 - nominal: '0.58'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X160 - nominal: '0.62'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X150 - nominal: '0.68'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W36X135 - nominal: '0.79'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X387 - nominal: '0.16'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X354 - nominal: '0.17'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X318 - nominal: '0.19'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X291 - nominal: '0.21'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X263 - nominal: '0.24'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X241 - nominal: '0.26'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X221 - nominal: '0.29'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X201 - nominal: '0.32'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X169 - nominal: '0.58'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X152 - nominal: '0.65'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X141 - nominal: '0.73'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X130 - nominal: '0.81'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W33X118 - nominal: '0.93'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X391 - nominal: '0.16'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X357 - nominal: '0.18'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X326 - nominal: '0.20'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X292 - nominal: '0.22'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X261 - nominal: '0.25'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X235 - nominal: '0.28'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X211 - nominal: '0.31'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X191 - nominal: '0.35'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X173 - nominal: '0.46'  < required: '1.00'  -> O.K."],
```

```
["for checkCombined,,, W30X148 - nominal: '0.70'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X132 - nominal: '0.82'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X124 - nominal: '0.88'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X116 - nominal: '0.96'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W30X108 - nominal: '1.06'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W30X99 - nominal: '1.20'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W30X90 - nominal: '1.37'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W27X539 - nominal: '0.12'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X368 - nominal: '0.18'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X336 - nominal: '0.20'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X307 - nominal: '0.22'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X281 - nominal: '0.24'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X258 - nominal: '0.26'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X235 - nominal: '0.29'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X217 - nominal: '0.32'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X194 - nominal: '0.36'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X178 - nominal: '0.47'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X161 - nominal: '0.52'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X146 - nominal: '0.58'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X129 - nominal: '0.83'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X114 - nominal: '0.96'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W27X102 - nominal: '1.11'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W27X94 - nominal: '1.24'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W27X84 - nominal: '1.43'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W24X370 - nominal: '0.19'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X335 - nominal: '0.21'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X306 - nominal: '0.23'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X279 - nominal: '0.26'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X250 - nominal: '0.29'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X229 - nominal: '0.32'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X207 - nominal: '0.36'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X192 - nominal: '0.39'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X176 - nominal: '0.50'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X162 - nominal: '0.54'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X146 - nominal: '0.61'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X131 - nominal: '0.68'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X117 - nominal: '0.78'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X104 - nominal: '0.89'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W24X103 - nominal: '1.15'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W24X94 - nominal: '1.27'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W24X84 - nominal: '1.47'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W24X76 - nominal: '1.67'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W24X68 - nominal: '1.94'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W21X275 - nominal: '0.27'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X248 - nominal: '0.30'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X223 - nominal: '0.34'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X201 - nominal: '0.38'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X182 - nominal: '0.49'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X166 - nominal: '0.55'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X147 - nominal: '0.63'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X132 - nominal: '0.70'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X122 - nominal: '0.76'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X111 - nominal: '0.85'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X101 - nominal: '0.92'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W21X93 - nominal: '1.42'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W21X83 - nominal: '1.59'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W21X73 - nominal: '1.81'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W21X68 - nominal: '1.98'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W21X62 - nominal: '2.23'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W18X311 - nominal: '0.25'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X283 - nominal: '0.28'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X258 - nominal: '0.31'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X234 - nominal: '0.35'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X211 - nominal: '0.39'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X192 - nominal: '0.50'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X175 - nominal: '0.55'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X158 - nominal: '0.62'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X143 - nominal: '0.69'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X130 - nominal: '0.76'  < required: '1.00'  -> O.K."],
```

```
["for checkCombined,,, W18X119 - nominal: '0.84'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X106 - nominal: '0.96'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W18X97 - nominal: '1.05'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W18X86 - nominal: '1.18'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W18X76 - nominal: '1.35'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W18X71 - nominal: '2.05'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W18X65 - nominal: '2.24'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W16X100 - nominal: '1.07'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W16X89 - nominal: '1.21'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W16X77 - nominal: '1.41'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W16X67 - nominal: '1.63'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W14X873 - nominal: '0.07'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X808 - nominal: '0.08'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X730 - nominal: '0.09'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X665 - nominal: '0.10'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X605 - nominal: '0.11'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X550 - nominal: '0.12'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X500 - nominal: '0.14'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X455 - nominal: '0.15'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X426 - nominal: '0.17'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X398 - nominal: '0.18'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X370 - nominal: '0.19'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X342 - nominal: '0.21'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X311 - nominal: '0.24'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X283 - nominal: '0.26'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X257 - nominal: '0.29'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X233 - nominal: '0.32'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X211 - nominal: '0.36'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X193 - nominal: '0.40'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X176 - nominal: '0.44'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X159 - nominal: '0.55'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X145 - nominal: '0.61'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X132 - nominal: '0.69'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X120 - nominal: '0.76'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X109 - nominal: '0.84'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X99 - nominal: '0.93'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W14X90 - nominal: '1.03'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W14X82 - nominal: '1.38'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W14X74 - nominal: '1.52'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W14X68 - nominal: '1.68'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W14X61 - nominal: '1.88'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W12X336 - nominal: '0.24'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X305 - nominal: '0.27'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X279 - nominal: '0.30'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X252 - nominal: '0.34'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X230 - nominal: '0.37'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X210 - nominal: '0.41'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X190 - nominal: '0.46'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X170 - nominal: '0.58'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X152 - nominal: '0.66'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X136 - nominal: '0.75'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X120 - nominal: '0.86'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X106 - nominal: '0.98'  < required: '1.00'  -> O.K."],
["for checkCombined,,, W12X96 - nominal: '1.08'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W12X87 - nominal: '1.21'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W12X79 - nominal: '1.33'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W12X72 - nominal: '1.48'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W12X65 - nominal: '1.64'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W12X58 - nominal: '2.01'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W12X53 - nominal: '2.24'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W10X112 - nominal: '1.04'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W10X100 - nominal: '1.18'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W10X88 - nominal: '1.35'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W10X77 - nominal: '1.56'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W10X68 - nominal: '1.79'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W10X60 - nominal: '2.03'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W10X54 - nominal: '2.29'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W10X49 - nominal: '2.53'  > required: '1.00'  -> N.G."],
["for checkCombined,,, W8X67 - nominal: '2.15'  > required: '1.00'  -> N.G."],
```

```
 ["for checkCombined,,, W8X58 - nominal: '2.51'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, H-248×249x8x13 - nominal: '2.86'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-250×250x9x14 - nominal: '2.62'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-250×255x14x14 - nominal: '2.42'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-294×302x12x12 - nominal: '2.04'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-298×299x9x14 - nominal: '1.88'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-300×300x10x15 - nominal: '1.74'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-300×305x15x15 - nominal: '1.61'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-304×301x11x17 - nominal: '1.53'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-310×305x15x20 - nominal: '1.25'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-310×310x20x20 - nominal: '1.17'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-336×249x8x12 - nominal: '2.61'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-340×250x9x14 - nominal: '2.25'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-338×351x13x13 - nominal: '1.44'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-344×348x10x16 - nominal: '1.26'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-344×354x16x16 - nominal: '1.16'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-350×350x12x19 - nominal: '1.05'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-350×357x19x19 - nominal: '0.96'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-386×299x9x14 - nominal: '1.64'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-390×300x10x16 - nominal: '1.44'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-388×402x15x15 - nominal: '0.99'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-394×398x11x18 - nominal: '0.89'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-394×405x18x18 - nominal: '0.81'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-400×400x13x21 - nominal: '0.76'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-400×408x21x21 - nominal: '0.69'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-406×403x16x24 - nominal: '0.65'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-414×405x18x28 - nominal: '0.55'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-428×407x20x35 - nominal: '0.38'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-458×417x30x50 - nominal: '0.26'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-498×432x45x70 - nominal: '0.17'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-434×299x10x15 - nominal: '1.45'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-440×300x11x18 - nominal: '1.22'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-506×201x11x19 - nominal: '2.02'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-482×300x11x15 - nominal: '1.35'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-488×300x11x18 - nominal: '1.16'  > required: '1.00'  -> N.
G."],
```

```
 ["for checkCombined,,, H-600×200x11x17 - nominal: '2.14'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-606×201x12x20 - nominal: '1.80'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-612×202x13x23 - nominal: '1.56'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-582×300x12x17 - nominal: '1.12'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-588×300x12x20 - nominal: '0.98'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-594×302x14x23 - nominal: '0.82'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-692×300x13x20 - nominal: '0.91'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-700×300x13x24 - nominal: '0.78'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-708×302x15x28 - nominal: '0.65'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-792×300x14x22 - nominal: '0.80'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-800×300x14x26 - nominal: '0.69'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-808×302x16x30 - nominal: '0.59'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-890×299x15x23 - nominal: '0.74'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-900×300x16x28 - nominal: '0.61'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-912×302x18x34 - nominal: '0.49'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, H-918×303x19x37 - nominal: '0.45'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-1000x400x15x30 - nominal: '0.31'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-2200x350x10x30 - nominal: '0.48'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-700x350x10x30 - nominal: '0.56'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-2200x300x10x25 - nominal: '0.73'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-1500x300x10x25 - nominal: '0.73'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-1800x300x10x25 - nominal: '0.73'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-700x300x10x25 - nominal: '0.81'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-2200x350x10x25 - nominal: '0.57'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-1500x350x10x25 - nominal: '0.58'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-1200x300x10x25 - nominal: '0.74'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-700x350x10x25 - nominal: '0.65'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-2200x350x15x30 - nominal: '0.43'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-1500x350x15x30 - nominal: '0.44'  < required: '1.00'  -> O.
K."],
 ["for checkCombined,,, BH-1800x350x15x30 - nominal: '0.43'  < required: '1.00'  -> O.
K."]]
```

In [42]:

```
### 가변 조건 검토를 위한 함수 합성

multiChecker2 = lambda _list: goW
(
    _list,
    map(makeSectionForm),
    map(setSectionProp),
```

```
    map(lambda x: checkDesignResult(input_dsgnMode, x, mProp, dBase, subBaseColl, sub
    filter(lambda x: x.result_flx.Result == "N.G." ),
    map(lambda x: showResult(x, "checkCombined")),

list)
```

In [43]:
```
multiChecker2(targetSections)
```

Out[43]:
```
[["for checkCombined,,, W18X35 - nominal: '6.49'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W16X36 - nominal: '4.67'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W16X31 - nominal: '7.80'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W16X26 - nominal: '9.94'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W14X38 - nominal: '4.37'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W14X34 - nominal: '4.99'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W14X30 - nominal: '5.89'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W14X26 - nominal: '10.36'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W14X22 - nominal: '13.26'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X45 - nominal: '3.12'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X40 - nominal: '3.53'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X35 - nominal: '4.80'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X30 - nominal: '5.75'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X26 - nominal: '6.76'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X22 - nominal: '18.01'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X19 - nominal: '21.89'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X16 - nominal: '29.49'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W12X14 - nominal: '34.75'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X54 - nominal: '2.29'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X49 - nominal: '2.53'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X45 - nominal: '3.16'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X39 - nominal: '3.71'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X33 - nominal: '4.55'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X30 - nominal: '6.71'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X26 - nominal: '7.93'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X22 - nominal: '9.69'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X19 - nominal: '20.51'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X17 - nominal: '23.92'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X15 - nominal: '29.11'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W10X12 - nominal: '38.56'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X58 - nominal: '2.51'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X48 - nominal: '3.07'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X40 - nominal: '3.78'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X35 - nominal: '4.33'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X31 - nominal: '4.99'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X28 - nominal: '6.48'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X24 - nominal: '7.61'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X21 - nominal: '11.34'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X18 - nominal: '13.98'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X15 - nominal: '26.66'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X13 - nominal: '32.76'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W8X10 - nominal: '42.63'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W6X25 - nominal: '8.33'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W6X20 - nominal: '10.62'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W6X15 - nominal: '14.98'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W6X16 - nominal: '22.36'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W6X12 - nominal: '32.35'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W6X9 - nominal: '44.47'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W6X8.5 - nominal: '48.99'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W5X19 - nominal: '14.01'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W5X16 - nominal: '17.06'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, W4X13 - nominal: '29.28'  > required: '1.00'  -> N.G."],
 ["for checkCombined,,, H-100×100x6x8 - nominal: '35.01'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-125×125x6.5x9 - nominal: '17.94'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-150×75x5x7 - nominal: '68.92'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-148×100x6x9 - nominal: '27.32'  > required: '1.00'  -> N.
```

```
G."],
 ["for checkCombined,,, H-150×150x7x10 - nominal: '10.48'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-198×99x4.5x7 - nominal: '32.89'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-200×100x5.5x8 - nominal: '28.34'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-194×150x6x9 - nominal: '10.41'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-200×200x8x12 - nominal: '4.72'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-200×204x12x12 - nominal: '4.39'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-208×202x10x16 - nominal: '3.48'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-248×124x5x8 - nominal: '16.18'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-250×125x6x9 - nominal: '14.01'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-244×175x7x11 - nominal: '5.84'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-244×252x11x11 - nominal: '3.15'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-248×249x8x13 - nominal: '2.86'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-250×250x9x14 - nominal: '2.62'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-250×255x14x14 - nominal: '2.42'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-298×149x5.5x8 - nominal: '10.14'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-300×150x6.5x9 - nominal: '8.82'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-294×200x8x12 - nominal: '3.92'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-298×201x9x14 - nominal: '3.34'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-346×174x6x9 - nominal: '6.27'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-350×175x7x11 - nominal: '5.05'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-354×176x8x13 - nominal: '4.22'  > required: '1.00'  -> N.
G."],
 ["for checkCombined,,, H-396×199x7x11 - nominal: '3.87'  > required: '1.00'  -> N.
G."]]
```

In [44]:

```python
def showGraph(y_value, threshold=None):
    x = range(len(y_value))
    y = y_value
    fig, ax = plt.subplots(figsize=(20,20))
    if threshold:
        ax.axhline(threshold, 0, 1, color='red', linestyle='--', linewidth=2)

    plt.plot(x,y,'or')
    plt.show()

#     return y_value
```

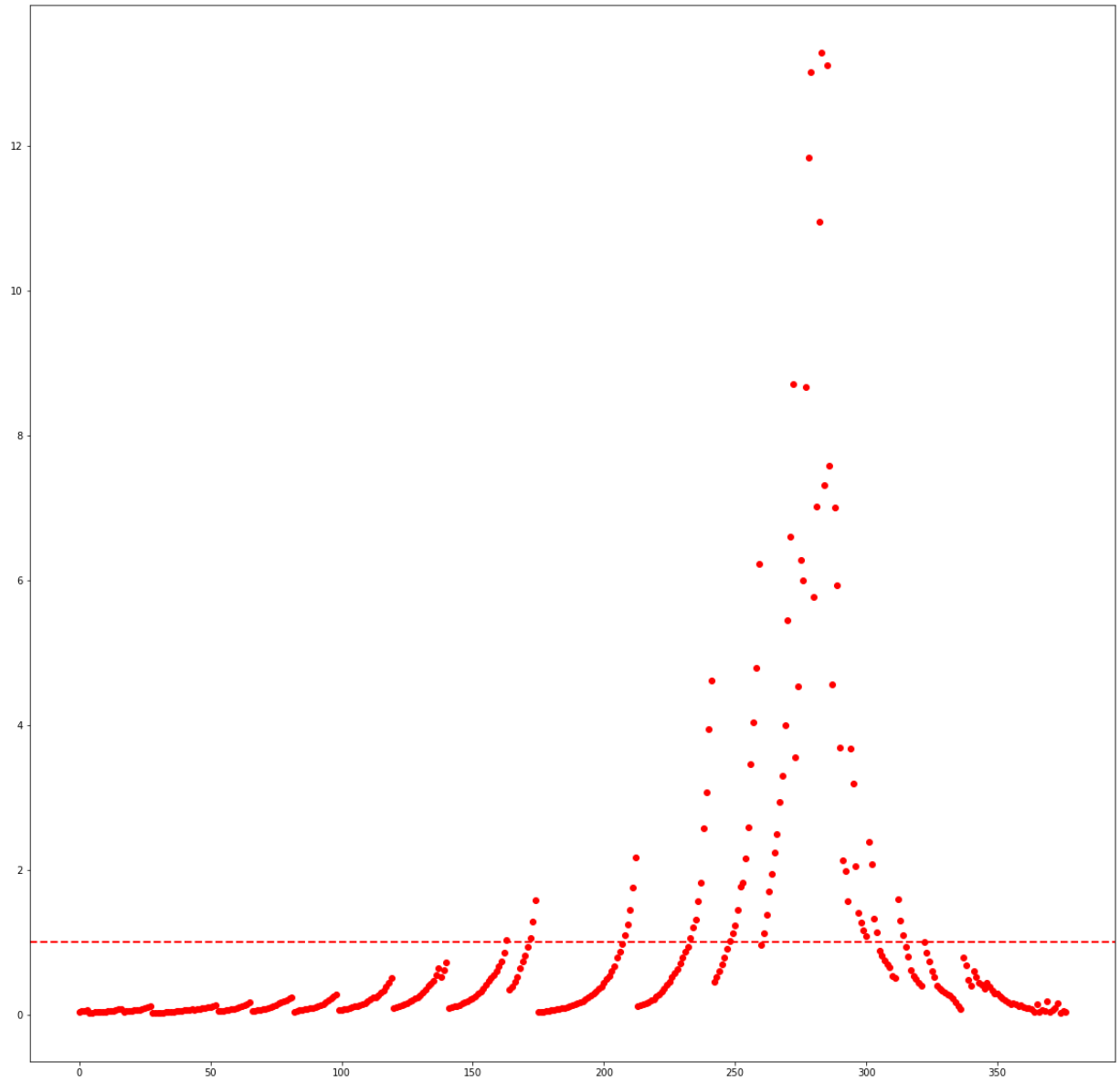
In [46]:

```python
multiChecker_forGraph = lambda _list: goW
(
    _list,
    map(makeSectionForm),
    map(setSectionProp),
    map(lambda x: checkDesignResult(input_dsgnMode, x, mProp, dBase, subBaseColl, sub
#     filter(lambda x: x.result_flx.Result == "O.K." ),
#     map(lambda x: x.result_flx.NomStr),list,
```

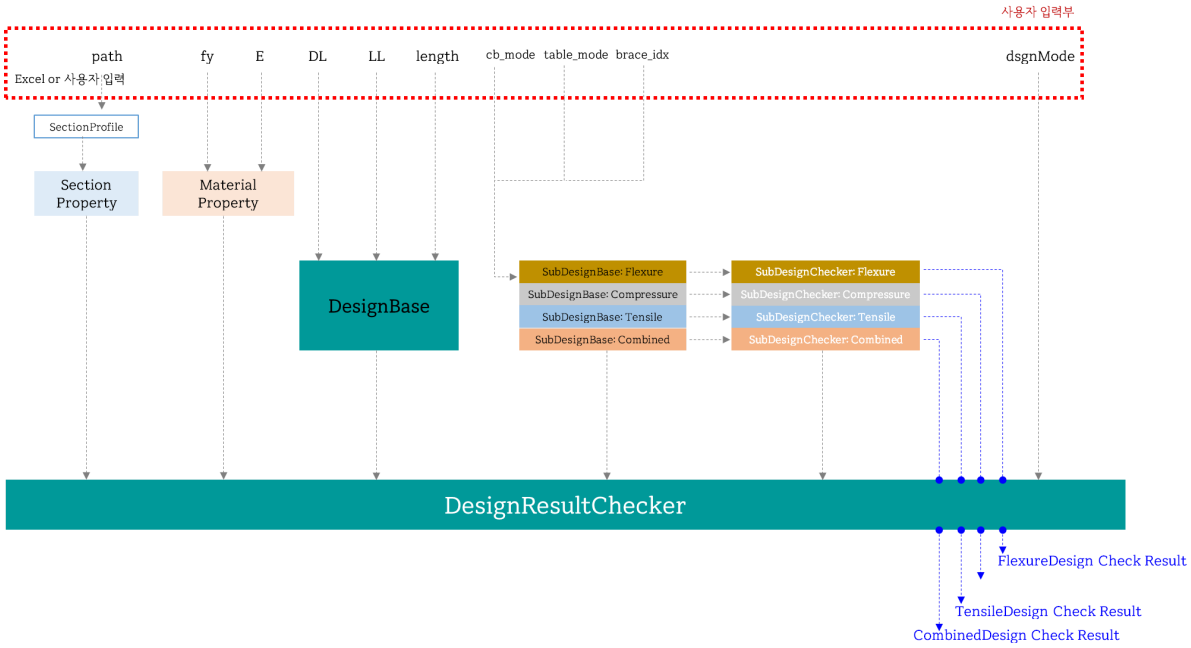
```
#     map(lambda x: x.result_flx.ReqStr),list,
    map(lambda x: x.result_flx.ReqStr/x.result_flx.NomStr),list,

#     showGraph,
    lambda x: showGraph(x, 1)
)

multiChecker_forGraph(targetSections)
```



Code Structure Diagram

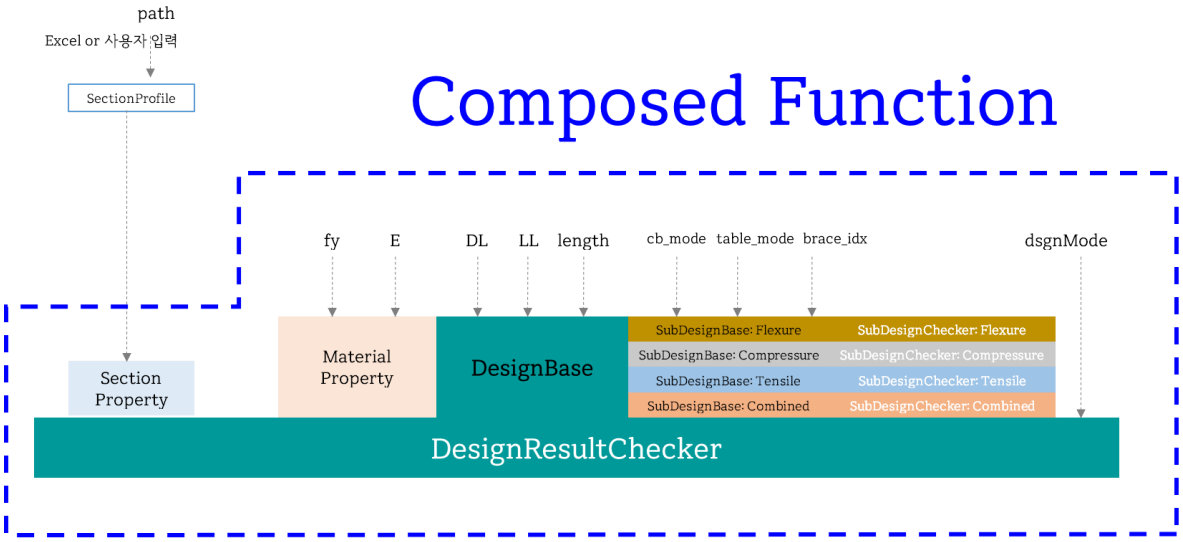


$$(g \circ f)(x) = g(f(x))$$

Data와 모델링을 추상화 하는 것 보다,

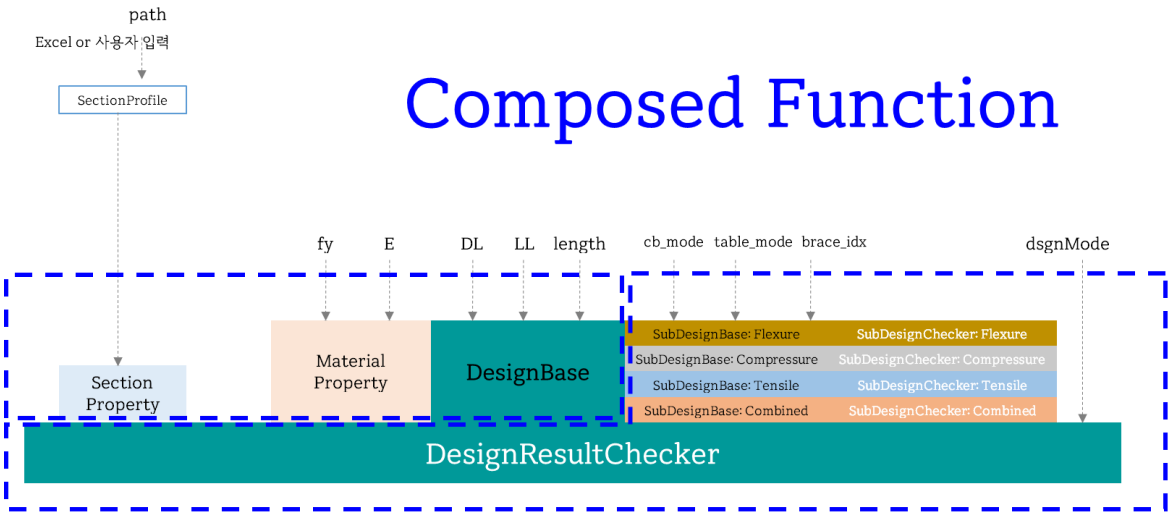
연산을 추상화하는 것이

복잡성을 다루는 데 더 좋다



하나의 연산으로 추상화된 12가지의 연산(함수)

※추상화 범위는 얼마든지 변경이 가능하다.



※추상화 범위는 얼마든지 변경이 가능하다.

In [ ]: