



49 Skyscrapers

Seong Ki LEE
WS0607 MAS
CAAD **D**-ARCH ETH

Seong Ki LEE

Taipei 101

8 Sung-Chih Road, Hsin-Yi District
Taipei

Construction Dates

Started : 1999

Finished : 2004

Floor Count : 101

Basement Floors : 5

Floor Area : 412.500m²

Elevator Count : 61

Company

C.Y. Lee & Partners architect

Building Uses

mixed use

Structural Types

high-rise, tuned mass damper, pole

Architectural Style

pagoda style

Materials

glass, steel

Reference

<http://skyscraperpage.com/cities/?buildingID=18/>

<http://www.tfc101.com.tw>

<http://www.cylee.com/>



Pagoda

Tower-like multistoried structure of stone, brick, or wood, usually associated with a Buddhist temple complex and enshrining sacred relics.

The pagoda evolved from the Indian stupa. The pagoda's crowning ornament is pyramidal or conical in Myanmar (Burma), Thailand, Cambodia, and Laos and bottle-shaped in the Tibet Autonomous Region of China. In other parts of China and in Korea and Japan, a pagoda is a tall tower repeating a basic story unit in diminishing proportions. The stories may be circular, square, or polygonal. The pagoda form is intended mainly as a monument and has very little usable interior space

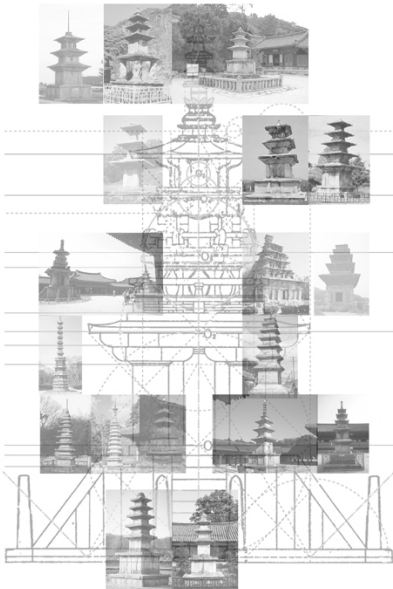
< source :: Encyclopedia Britannica : <http://concise.britannica.com/ebc/article-9374366> >

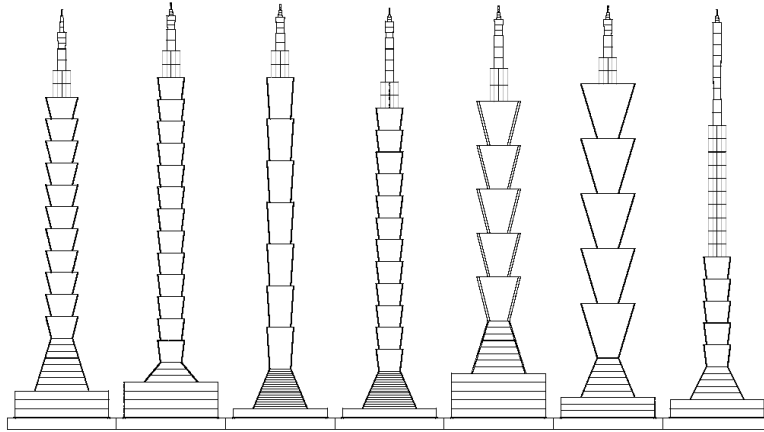
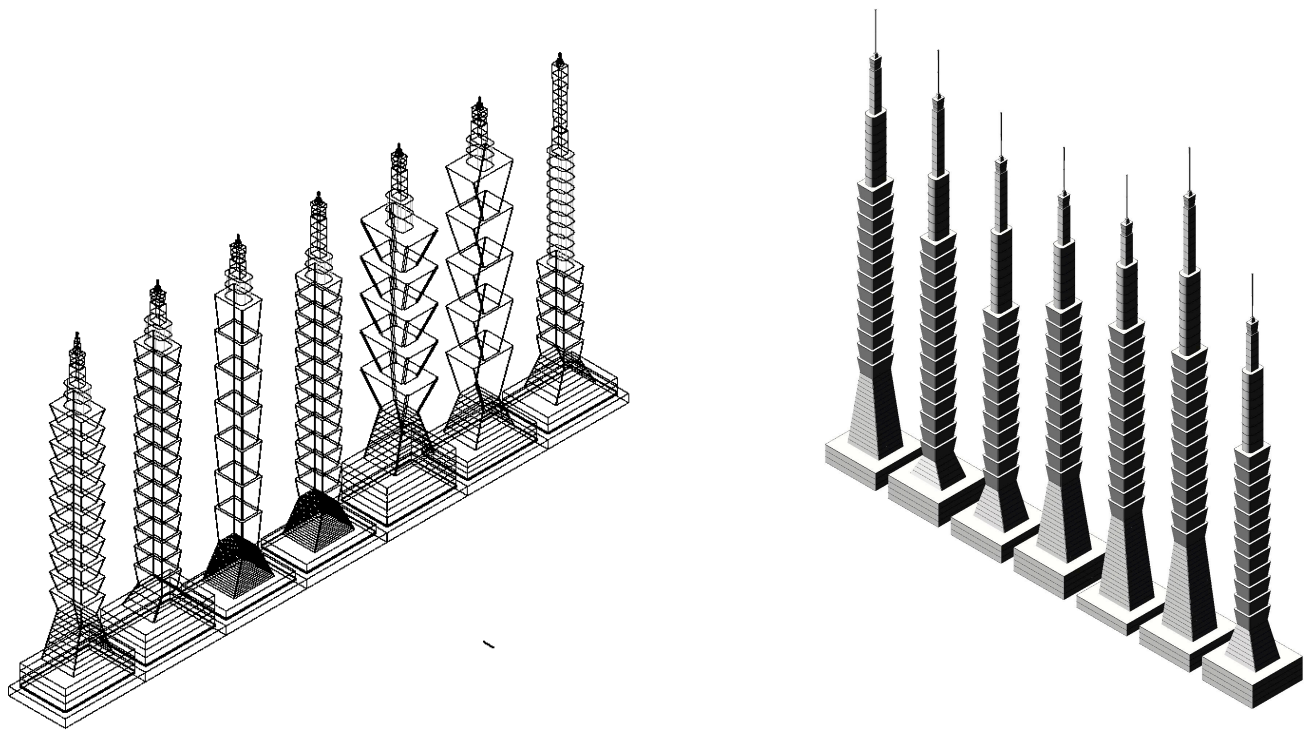
Concept

Shape (Geometric) properties of hermitages + Building function or structure = Innovative design, new symbol (Landmark)



Target





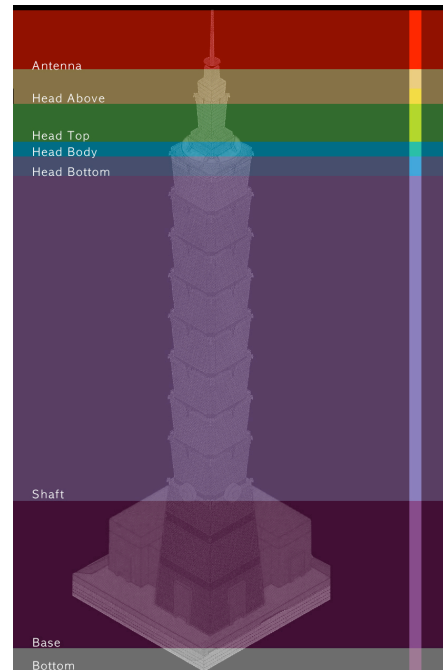
Parameters

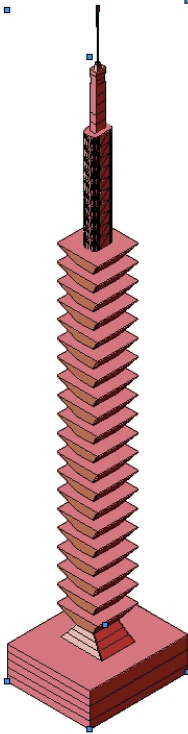
Name	Alternate Name	Type	Default
AntennaHeight	Height of Antenna	Number	30
Numb_antenna	Number of Antenna	Integer	2
Numb_headAbove	Number of HeadAbove	Integer	3
HeadAboveHeight	Height of HeadAbove	Number	2
Numb_headTop	Number of HeadTop	Integer	3
HeadTopHeight	Height of HeadTop	Number	5
Numb_headBody	Number of HeadBody	Integer	5
HeadBodyHeight	Height of HeadBody	Number	10
Numb_headBottom	Number of HeadBottom	Integer	5
HeadBottomHeight	Height of HeadBottom	Number	12
Numb_shaft	Number of Shaft	Integer	20
ShaftCornerDepth	Depth of Shaft Corner	Number	0.7
ShaftHeight	Height of Shaft	Number	20
Shaft2MyLength	Length of Shaft	Number	25
Shaft2MyWidth	Width of Shaft	Number	35
Numb_base	Number of Base	Integer	3
BaseHeight	Height of Base	Number	6
BaseCornerDepth	Depth of Base Corner	Number	0.5
Base2MyLength	Length of Base's top	Number	20
Base2MyWidth	Width of Base's top	Number	30
Base1MyLength	Length of Base's bottom	Number	50
Base1MyWidth	Width of Base's bottom	Number	40
Numb_underbase	Number of Underbase	Integer	5
UnderbaseHeight	Height of Underbase	Number	8
Underbase1MyLength	Length of Underbase	Number	100
Underbase1MyWidth	Width of Underbase	Number	120

Add... Edit... Remove Move Up Move Down

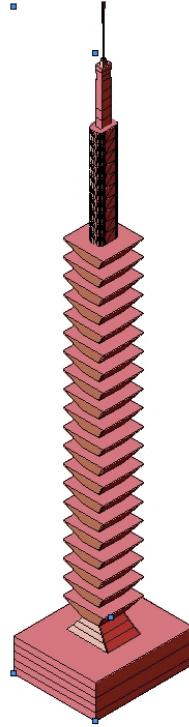
Press the Help key or Cmd+? for help.

Cancel OK

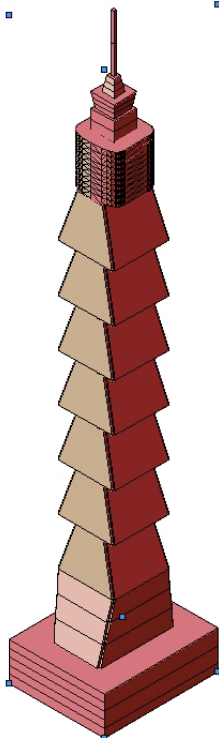




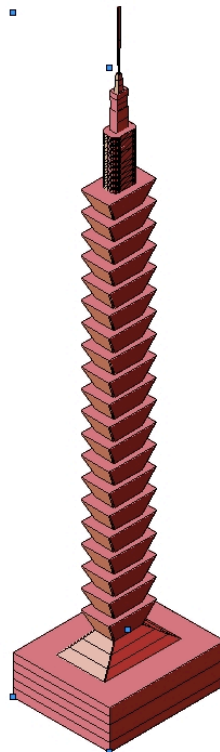
Obj Info		
Shape	Data	Render
parametric Taipei 101		
Class:	None	
Layer:	Design Lay...	
X:	47'6.735"	
Y:	51'9.889"	
Z:	-6'5.846"	
Rot:	0.00°	
Height of Antenna:	30	
Number of Antenna:	2	
Number of HeadAbove:	3	
Height of HeadAbove:	2	
Number of HeadTop:	3	
Height of HeadTop:	5	
Number of HeadBody:	5	
Height of HeadBody:	10	
Number of HeadBottom:	10	
Height of HeadBottom:	12	
Number of Shaft:	20	
Depth of Shaft Corner:	0.7	
Height of Shaft:	20	
Length of Shaft:	50	
Width of Shaft:	50	
Number of Base:	3	
Height of Base:	6	
Depth of Base Corner:	0.5	
Length of Base's top:	20	
Width of Base's top:	30	
Length of Base's ...:	40	
Width of Base's b...:	50	
Number of Underbase:	5	
Height of Underbase:	8	
Length of Underbase:	100	
Width of Underbase:	120	



Obj Info		
Shape	Data	Render
parametric Taipei 101		
Class:	None	
Layer:	Design Lay...	
X:	47'6.735"	
Y:	51'9.889"	
Z:	-6'5.846"	
Rot:	0.00°	
Height of Antenna:	30	
Number of Antenna:	2	
Number of HeadAbove:	3	
Height of HeadAbove:	2	
Number of HeadTop:	3	
Height of HeadTop:	5	
Number of HeadBody:	5	
Height of HeadBody:	10	
Number of HeadBottom:	10	
Height of HeadBottom:	12	
Number of Shaft:	20	
Depth of Shaft Corner:	0.7	
Height of Shaft:	20	
Length of Shaft:	50	
Width of Shaft:	50	
Number of Base:	3	
Height of Base:	6	
Depth of Base Corner:	0.5	
Length of Base's top:	20	
Width of Base's top:	30	
Length of Base's bottom:	50	
Width of Base's bottom:	40	
Number of Underbase:	5	
Height of Underbase:	8	
Length of Underbase:	100	
Width of Underbase:	120	



Obj Info		
Shape	Data	Render
parametric Taipei 101		
Class:	None	
Layer:	Design Lay...	
X:	47'6.735"	
Y:	51'9.889"	
Z:	-6'5.846"	
Rot:	0.00°	
Height of Antenna:	30	
Number of Antenna:	2	
Number of HeadAbove:	3	
Height of HeadAbove:	5	
Number of HeadTop:	3	
Height of HeadTop:	5	
Number of HeadBody:	2	
Height of HeadBody:	10	
Number of HeadBottom:	10	
Height of HeadBottom:	5	
Number of Shaft:	7	
Depth of Shaft Corner:	2	
Height of Shaft:	50	
Length of Shaft:	30	
Width of Shaft:	50	
Number of Base:	3	
Height of Base:	20	
Depth of Base Corner:	2	
Length of Base's top:	60	
Width of Base's top:	60	
Length of Base's bottom:	50	
Width of Base's bottom:	80	
Number of Underbase:	5	
Height of Underbase:	8	
Length of Underbase:	100	
Width of Underbase:	120	



Obj Info		
Shape	Data	Render
parametric Taipei 101		
Class:	None	
Layer:	Design Lay...	
X:	47'6.735"	
Y:	51'9.889"	
Z:	-6'5.846"	
Rot:	0.00°	
Height of Antenna:	30	
Number of Antenna:	2	
Number of HeadAbove:	3	
Height of HeadAbove:	5	
Number of HeadTop:	3	
Height of HeadTop:	5	
Number of HeadBody:	2	
Height of HeadBody:	10	
Number of HeadBottom:	10	
Height of HeadBottom:	5	
Number of Shaft:	20	
Depth of Shaft Corner:	0.7	
Height of Shaft:	20	
Length of Shaft:	30	
Width of Shaft:	50	
Number of Base:	3	
Height of Base:	6	
Depth of Base Corner:	0.5	
Length of Base's top:	20	
Width of Base's top:	30	
Length of Base's bottom:	50	
Width of Base's bottom:	80	
Number of Underbase:	5	
Height of Underbase:	8	
Length of Underbase:	100	
Width of Underbase:	120	

PROCEDURE paraBuilding_Taipei101;

TYPE

```
BASICMODULE = STRUCTURE
    myLength : REAL;
    myWidth : REAL;
END;
```

VAR

```
underbase : DYNARRAY[] OF BASICMODULE;
base : DYNARRAY[] OF BASICMODULE;
shaft : DYNARRAY[] OF BASICMODULE;
headBottom : DYNARRAY[] OF BASICMODULE;
headBody : DYNARRAY[] OF BASICMODULE;
headTop : DYNARRAY[] OF BASICMODULE;
headAbove : DYNARRAY[] OF BASICMODULE;
antenna : DYNARRAY[] OF BASICMODULE;
```

```
numb_underbase : INTEGER;
numb_base : INTEGER;
numb_shaft : INTEGER;
numb_headBottom : INTEGER;
numb_headBody : INTEGER;
numb_headTop : INTEGER;
numb_headAbove : INTEGER;
numb_antenna : INTEGER;
```

```
underbase_nCH : DYNARRAY[] OF HANDLE;
base_nCH : DYNARRAY[] OF HANDLE;
shaft_nCH : DYNARRAY[] OF HANDLE;
headBottom_nCH : DYNARRAY[] OF HANDLE;
headBody_nCH : DYNARRAY[] OF HANDLE;
headTop_nCH : DYNARRAY[] OF HANDLE;
headAbove_nCH : DYNARRAY[] OF HANDLE;
antenna_nCH : DYNARRAY[] OF HANDLE;
```

```
underbase_groupHandle : HANDLE;
base_groupHandle : HANDLE;
shaft_groupHandle : HANDLE;
headBottom_groupHandle : HANDLE;
headBody_groupHandle : HANDLE;
headTop_groupHandle : HANDLE;
headAbove_groupHandle : HANDLE;
antenna_groupHandle : HANDLE;
```

```
bRule, bClose, bSolid : BOOLEAN;
```

```
i : INTEGER;
```

```
underbaseHeight : REAL;
baseHeight : REAL;
shaftHeight : REAL;
headBottomHeight : REAL;
headBodyHeight : REAL;
headTopHeight : REAL;
headAboveHeight : REAL;
antennaHeight : REAL;
```

```
sumOfUnderbaseHeight : REAL;
sumOfBaseHeight : REAL;
sumOfShaftHeight : REAL;
sumOfHeadBottomHeight : REAL;
sumOfHeadBodyHeight : REAL;
sumOfHeadTopHeight : REAL;
sumOfHeadAboveHeight : REAL;
sumOfHeadHeight : REAL;
sumOfAntennaHeight : REAL;
```

```
sumOfUnderHeadBottom : REAL;
sumOfUnderHeadBody : REAL;
sumOfUnderHeadTop : REAL;
sumOfUnderHeadAbove : REAL;
sumOfUnderAntenna : REAL;
```

```
averIncOfmyL : REAL;
averIncOfmyW : REAL;
```

```
averIncOfHeadTopL : REAL;  
averIncOfHeadTopW : REAL;  
averIncOfHeadAboveL : REAL;  
averIncOfHeadAboveW : REAL;  
averIncOfAntennaL : REAL;  
averIncOfAntennaW : REAL;
```

```
baseCornerDepth : REAL;  
shaftCornerDepth : REAL;
```

BEGIN

```
numb_underbase := pNumb_underbase;  
numb_base := pNumb_base;  
numb_shaft := pNumb_shaft;  
numb_headBottom := pNumb_headBottom;  
numb_headBody := pNumb_headBody;  
numb_headTop := pNumb_headTop;  
numb_headAbove := pNumb_headAbove;  
numb_antenna := pNumb_antenna;
```

```
ALLOCATE underbase[1..numb_underbase];  
ALLOCATE base[1..numb_base];  
ALLOCATE shaft[1..numb_shaft];  
ALLOCATE headBottom[1..numb_headBottom];  
ALLOCATE headBody[1..numb_headBody];  
ALLOCATE headTop[1..numb_headTop];  
ALLOCATE headAbove[1..numb_headAbove];  
ALLOCATE antenna[1..numb_antenna];
```

```
ALLOCATE underbase_nCH[0..numb_underbase];  
ALLOCATE base_nCH[0..numb_base];  
ALLOCATE shaft_nCH[0..numb_shaft];  
ALLOCATE headBottom_nCH[0..numb_headBottom];  
ALLOCATE headBody_nCH[0..numb_headBody];  
ALLOCATE headTop_nCH[0..numb_headTop];  
ALLOCATE headAbove_nCH[0..numb_headAbove];  
ALLOCATE antenna_nCH[0..numb_antenna];
```

```
bRule := TRUE;  
bClose := TRUE;  
bSolid := TRUE;
```

```
underbaseHeight := pUnderbaseHeight;  
baseHeight := pBaseHeight;  
shaftHeight := pShaftHeight;  
headBottomHeight := pHeadBottomHeight;  
headBodyHeight := pHeadBodyHeight;  
headTopHeight := pHeadTopHeight;  
headAboveHeight := pHeadAboveHeight;  
antennaHeight := pAntennaHeight;
```

```
baseCornerDepth := pBaseCornerDepth;  
shaftCornerDepth := pShaftCornerDepth;
```

```
underbase[1].myLength := pUnderbase1MyLength;  
underbase[1].myWidth := pUnderbase1MyWidth;
```

```
base[1].myLength := pBase1MyLength;  
base[1].myWidth := pBase1MyWidth;  
base[2].myLength := pBase2MyLength;  
base[2].myWidth := pBase2MyWidth;
```

```
shaft[1].myLength := base[2].myLength;  
shaft[1].myWidth := base[2].myWidth;  
shaft[2].myLength := pShaft2MyLength;  
shaft[2].myWidth := pShaft2MyWidth;
```

```
headBottom[1].myLength := shaft[1].myLength*0.8;  
headBottom[1].myWidth := shaft[1].myWidth*0.8;
```

```
headBody[1].myLength := headBottom[1].myLength*0.5;  
headBody[1].myWidth := headBottom[1].myWidth*0.5;
```

```
headTop[1].myLength := headBody[1].myLength*0.8;  
headTop[1].myWidth := headBody[1].myWidth*0.8;  
headTop[2].myLength := headBody[1].myLength;  
headTop[2].myWidth := headBody[1].myWidth;
```



```

headAbove[1].myLength := headTop[2].myLength*0.5;
headAbove[1].myWidth := headTop[2].myWidth*0.5;
headAbove[2].myLength := headTop[2].myLength*0.3;
headAbove[2].myWidth := headTop[2].myWidth*0.3;

antenna[1].myLength := headAbove[1].myLength * 0.2;
antenna[1].myWidth := headAbove[1].myWidth * 0.2;
antenna[2].myLength := headAbove[2].myLength * 0.2;
antenna[2].myWidth := headAbove[2].myWidth * 0.2;

sumOfUnderbaseHeight := numb_underbase * underbaseHeight;
sumOfBaseHeight := numb_base * baseHeight;
sumOfShaftHeight := numb_shaft * shaftHeight;
sumOfHeadBottomHeight := numb_headBottom * headBottomHeight;
sumOfHeadBodyHeight := numb_headBody * headBodyHeight;
sumOfHeadTopHeight := numb_headTop * headTopHeight;
sumOfHeadAboveHeight := numb_headAbove * headAboveHeight;
sumOfAntennaHeight := numb_antenna * antennaHeight;
sumOfHeadHeight := sumOfHeadBottomHeight + sumOfHeadBodyHeight + sumOfHeadTopHeight + sumOfAntennaHeight;

FOR i := 1 TO numb_underbase DO BEGIN
  BeginGroup;
  underbase_nCH[i-1] := CreateNurbsCurve(-0.5*underbase[1].myLength, -0.5*underbase[1].myWidth,
underbaseHeight*(i-1), true, 1);
  AddVertex3D(underbase_nCH[i-1], 0.5*underbase[1].myLength, -0.5*underbase[1].myWidth,
underbaseHeight*(i-1));
  AddVertex3D(underbase_nCH[i-1], 0.5*underbase[1].myLength, 0.5*underbase[1].myWidth,
underbaseHeight*(i-1));
  AddVertex3D(underbase_nCH[i-1], -0.5*underbase[1].myLength, 0.5*underbase[1].myWidth,
underbaseHeight*(i-1));
  AddVertex3D(underbase_nCH[i-1], -0.5*underbase[1].myLength, -0.5*underbase[1].myWidth,
underbaseHeight*(i-1));

  underbase_nCH[i] := CreateNurbsCurve(-0.5*underbase[1].myLength, -0.5*underbase[1].myWidth,
underbaseHeight*(i), true, 1);
  AddVertex3D(underbase_nCH[i], 0.5*underbase[1].myLength, -0.5*underbase[1].myWidth,
underbaseHeight*(i));
  AddVertex3D(underbase_nCH[i], 0.5*underbase[1].myLength, 0.5*underbase[1].myWidth,
underbaseHeight*(i));
  AddVertex3D(underbase_nCH[i], -0.5*underbase[1].myLength, 0.5*underbase[1].myWidth,
underbaseHeight*(i));
  AddVertex3D(underbase_nCH[i], -0.5*underbase[1].myLength, -0.5*underbase[1].myWidth,
underbaseHeight*(i));

  EndGroup;
  underbase_groupHandle := LNewObj;
  underbase_groupHandle := CreateLoftSurfaces(underbase_groupHandle, bRule, bClose, bSolid);
END;

averIncOfmyL := (base[1].myLength - base[2].myLength) / numb_base;
averIncOfmyW := (base[1].myWidth - base[2].myWidth) / numb_base;

FOR i := 1 TO numb_base DO BEGIN
  BeginGroup;

  base_nCH[i-1] := CreateNurbsCurve(-0.5*(base[1].myLength-(averIncOfmyL)*(i-1)),
-0.5*(base[1].myWidth-(averIncOfmyW)*(i-1))+ baseCornerDepth, baseHeight*(i-1)+sumOfUnderbaseHeight, true, 1);
  AddVertex3D(base_nCH[i-1], -0.5*(base[1].myLength-(averIncOfmyL)*(i-1))+ baseCornerDepth, -0.5*(base[1].myWidth-(averIncOfmyW)*(i-1))+ baseCornerDepth, baseHeight*(i-1)+sumOfUnderbaseHeight);
  AddVertex3D(base_nCH[i-1], -0.5*(base[1].myLength-(averIncOfmyL)*(i-1))+ baseCornerDepth, -0.5*(base[1].myWidth-(averIncOfmyW)*(i-1)), baseHeight*(i-1)+sumOfUnderbaseHeight);
  AddVertex3D(base_nCH[i-1], 0.5*(base[1].myLength-(averIncOfmyL)*(i-1))-baseCornerDepth, -0.5*(base[1].myWidth-(averIncOfmyW)*(i-1))+ baseCornerDepth, baseHeight*(i-1)+sumOfUnderbaseHeight);
  AddVertex3D(base_nCH[i-1], 0.5*(base[1].myLength-(averIncOfmyL)*(i-1))-baseCornerDepth, -0.5*(base[1].myWidth-(averIncOfmyW)*(i-1)), baseHeight*(i-1)+sumOfUnderbaseHeight);
  AddVertex3D(base_nCH[i-1], 0.5*(base[1].myLength-(averIncOfmyL)*(i-1))-baseCornerDepth, 0.5*(base[1].myWidth-(averIncOfmyW)*(i-1))+ baseCornerDepth, baseHeight*(i-1)+sumOfUnderbaseHeight);
  AddVertex3D(base_nCH[i-1], 0.5*(base[1].myLength-(averIncOfmyL)*(i-1))-baseCornerDepth, 0.5*(base[1].myWidth-(averIncOfmyW)*(i-1)), baseHeight*(i-1)+sumOfUnderbaseHeight);
  AddVertex3D(base_nCH[i-1], -0.5*(base[1].myLength-(averIncOfmyL)*(i-1))+ baseCornerDepth, 0.5*(base[1].myWidth-(averIncOfmyW)*(i-1)), baseHeight*(i-1)+sumOfUnderbaseHeight);

```



```

        AddVertex3D(base_nCH[i-1], -0.5*(base[1].myLength-(averIncOfmyL)*(i-1))+ baseCorner-
Depth, 0.5*(base[1].myWidth-(averIncOfmyW)*(i-1))-baseCornerDepth, baseHeight*(i-1)+sumOfUnderbaseHeight);
        AddVertex3D(base_nCH[i-1], -0.5*(base[1].myLength-(averIncOfmyL)*(i-1)),
0.5*(base[1].myWidth-(averIncOfmyW)*(i-1))-baseCornerDepth, baseHeight*(i-1)+sumOfUnderbaseHeight);

        AddVertex3D(base_nCH[i-1], -0.5*(base[1].myLength-(averIncOfmyL)*(i-1)),
-0.5*(base[1].myWidth-(averIncOfmyW)*(i-1))+ baseCornerDepth, baseHeight*(i-1)+sumOfUnderbaseHeight);

        base_nCH[i] := CreateNurbsCurve(-0.5*(base[1].myLength-(averIncOfmyL)*i),
-0.5*(base[1].myWidth-(averIncOfmyW)*i)+ baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight, true, 1);
        AddVertex3D(base_nCH[i], -0.5*(base[1].myLength-(averIncOfmyL)*i)+ baseCornerDepth,
-0.5*(base[1].myWidth-(averIncOfmyW)*i)+ baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight);
        AddVertex3D(base_nCH[i], -0.5*(base[1].myLength-(averIncOfmyL)*i)+ baseCornerDepth,
-0.5*(base[1].myWidth-(averIncOfmyW)*i), baseHeight*i+sumOfUnderbaseHeight);

        AddVertex3D(base_nCH[i], 0.5*(base[1].myLength-(averIncOfmyL)*i)-baseCornerDepth,
-0.5*(base[1].myWidth-(averIncOfmyW)*i), baseHeight*i+sumOfUnderbaseHeight);
        AddVertex3D(base_nCH[i], 0.5*(base[1].myLength-(averIncOfmyL)*i)-baseCornerDepth,
-0.5*(base[1].myWidth-(averIncOfmyW)*i)+ baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight);
        AddVertex3D(base_nCH[i], 0.5*(base[1].myLength-(averIncOfmyL)*i),
-0.5*(base[1].myWidth-(averIncOfmyW)*i)+ baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight);

        AddVertex3D(base_nCH[i], 0.5*(base[1].myLength-(averIncOfmyL)*i),
0.5*(base[1].myWidth-(averIncOfmyW)*i)-baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight);
        AddVertex3D(base_nCH[i], 0.5*(base[1].myLength-(averIncOfmyL)*i)-baseCornerDepth,
0.5*(base[1].myWidth-(averIncOfmyW)*i)-baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight);
        AddVertex3D(base_nCH[i], 0.5*(base[1].myLength-(averIncOfmyL)*i)-baseCornerDepth,
0.5*(base[1].myWidth-(averIncOfmyW)*i), baseHeight*i+sumOfUnderbaseHeight);

        AddVertex3D(base_nCH[i], -0.5*(base[1].myLength-(averIncOfmyL)*i)+ baseCornerDepth,
0.5*(base[1].myWidth-(averIncOfmyW)*i), baseHeight*i+sumOfUnderbaseHeight);
        AddVertex3D(base_nCH[i], -0.5*(base[1].myLength-(averIncOfmyL)*i)+ baseCornerDepth,
0.5*(base[1].myWidth-(averIncOfmyW)*i)-baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight);
        AddVertex3D(base_nCH[i], -0.5*(base[1].myLength-(averIncOfmyL)*i),
0.5*(base[1].myWidth-(averIncOfmyW)*i)-baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight);

        AddVertex3D(base_nCH[i], -0.5*(base[1].myLength-(averIncOfmyL)*i),
-0.5*(base[1].myWidth-(averIncOfmyW)*i)+ baseCornerDepth, baseHeight*i+sumOfUnderbaseHeight);

        EndGroup;
        base_groupHandle := LNewObj;

        base_groupHandle := CreateLoftSurfaces(base_groupHandle, bRule, bClose, bSolid);
    END;

    FOR i := 1 TO numb_shaft DO BEGIN
        BeginGroup;

        shaft_nCH[i-1] := CreateNurbsCurve(-0.5*shaft[1].myLength, -0.5*shaft[1].myWidth+shaftCornerDepth,
shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight, true, 1);
        AddVertex3D(shaft_nCH[i-1], -0.5*shaft[1].myLength+shaftCornerDepth,
-0.5*shaft[1].myWidth+shaftCornerDepth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);
        AddVertex3D(shaft_nCH[i-1], -0.5*shaft[1].myLength+shaftCornerDepth,
-0.5*shaft[1].myWidth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);

        AddVertex3D(shaft_nCH[i-1], 0.5*shaft[1].myLength-shaftCornerDepth,
-0.5*shaft[1].myWidth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);
        AddVertex3D(shaft_nCH[i-1], 0.5*shaft[1].myLength-shaftCornerDepth,
-0.5*shaft[1].myWidth+shaftCornerDepth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);
        AddVertex3D(shaft_nCH[i-1], 0.5*shaft[1].myLength,
-0.5*shaft[1].myWidth+shaftCornerDepth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);

        AddVertex3D(shaft_nCH[i-1], 0.5*shaft[1].myLength, 0.5*shaft[1].myWidth-shaftCornerDepth,
shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);
        AddVertex3D(shaft_nCH[i-1], 0.5*shaft[1].myLength-shaftCornerDepth,
0.5*shaft[1].myWidth-shaftCornerDepth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);
        AddVertex3D(shaft_nCH[i-1], 0.5*shaft[1].myLength-shaftCornerDepth, 0.5*shaft[1].myWidth,
shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);

        AddVertex3D(shaft_nCH[i-1], -0.5*shaft[1].myLength+shaftCornerDepth,
0.5*shaft[1].myWidth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);
        AddVertex3D(shaft_nCH[i-1], -0.5*shaft[1].myLength+shaftCornerDepth,
0.5*shaft[1].myWidth-shaftCornerDepth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);
        AddVertex3D(shaft_nCH[i-1], -0.5*shaft[1].myLength,
0.5*shaft[1].myWidth-shaftCornerDepth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);

        AddVertex3D(shaft_nCH[i-1], -0.5*shaft[1].myLength,
-0.5*shaft[1].myWidth+shaftCornerDepth, shaftHeight*(i-1)+sumOfUnderbaseHeight+sumOfBaseHeight);
    
```

```

shaft_nCH[i] := CreateNurbsCurve(-0.5*shaft[2].myLength, -0.5*shaft[2].myWidth+shaftCornerDepth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight, true, 1);
AddVertex3D(shaft_nCH[i], -0.5*shaft[2].myLength+shaftCornerDepth,
-0.5*shaft[2].myWidth+shaftCornerDepth, shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);
AddVertex3D(shaft_nCH[i], -0.5*shaft[2].myLength+shaftCornerDepth, -0.5*shaft[2].myWidth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);

AddVertex3D(shaft_nCH[i], 0.5*shaft[2].myLength-shaftCornerDepth, -0.5*shaft[2].myWidth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);
AddVertex3D(shaft_nCH[i], 0.5*shaft[2].myLength-shaftCornerDepth,
-0.5*shaft[2].myWidth+shaftCornerDepth, shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);
AddVertex3D(shaft_nCH[i], 0.5*shaft[2].myLength, -0.5*shaft[2].myWidth+shaftCornerDepth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);

AddVertex3D(shaft_nCH[i], 0.5*shaft[2].myLength, 0.5*shaft[2].myWidth-shaftCornerDepth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);
AddVertex3D(shaft_nCH[i], 0.5*shaft[2].myLength-shaftCornerDepth,
0.5*shaft[2].myWidth-shaftCornerDepth, shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);
AddVertex3D(shaft_nCH[i], 0.5*shaft[2].myLength-shaftCornerDepth, 0.5*shaft[2].myWidth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);

AddVertex3D(shaft_nCH[i], -0.5*shaft[2].myLength+shaftCornerDepth, 0.5*shaft[2].myWidth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);
AddVertex3D(shaft_nCH[i], -0.5*shaft[2].myLength+shaftCornerDepth,
0.5*shaft[2].myWidth-shaftCornerDepth, shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);
AddVertex3D(shaft_nCH[i], -0.5*shaft[2].myLength, 0.5*shaft[2].myWidth-shaftCornerDepth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);

AddVertex3D(shaft_nCH[i], -0.5*shaft[2].myLength, -0.5*shaft[2].myWidth+shaftCornerDepth,
shaftHeight*i+sumOfUnderbaseHeight+sumOfBaseHeight);

EndGroup;

shaft_groupHandle := LNewObj;

shaft_groupHandle := CreateLoftSurfaces(shaft_groupHandle, bRule, bClose, bSolid);

END;

sumOfUnderHeadBottom := sumOfUnderbaseHeight + sumOfBaseHeight + sumOfShaftHeight;

FOR i := 1 TO numb_headBottom DO BEGIN
BeginGroup;

headBottom_nCH[i-1] := CreateNurbsCurve(0, -0.5*headBottom[1].myWidth,
sumOfUnderHeadBottom+(i-1)*headBottomHeight, true, 2);
AddVertex3D(headBottom_nCH[i-1], 0.5*headBottom[1].myLength,
-0.5*headBottom[1].myWidth, sumOfUnderHeadBottom+(i-1)*headBottomHeight);
AddVertex3D(headBottom_nCH[i-1], 0.5*headBottom[1].myLength, 0,
sumOfUnderHeadBottom+(i-1)*headBottomHeight);
AddVertex3D(headBottom_nCH[i-1], 0.5*headBottom[1].myLength,
0.5*headBottom[1].myWidth, sumOfUnderHeadBottom+(i-1)*headBottomHeight);
AddVertex3D(headBottom_nCH[i-1], 0, 0.5*headBottom[1].myWidth,
sumOfUnderHeadBottom+(i-1)*headBottomHeight);
AddVertex3D(headBottom_nCH[i-1], -0.5*headBottom[1].myLength,
0.5*headBottom[1].myWidth, sumOfUnderHeadBottom+(i-1)*headBottomHeight);
AddVertex3D(headBottom_nCH[i-1], -0.5*headBottom[1].myLength, 0,
sumOfUnderHeadBottom+(i-1)*headBottomHeight);
AddVertex3D(headBottom_nCH[i-1], -0.5*headBottom[1].myLength,
-0.5*headBottom[1].myWidth, sumOfUnderHeadBottom+(i-1)*headBottomHeight);
AddVertex3D(headBottom_nCH[i-1], 0, -0.5*headBottom[1].myWidth,
sumOfUnderHeadBottom+(i-1)*headBottomHeight);

headBottom_nCH[i] := CreateNurbsCurve(0, -0.5*headBottom[1].myWidth, sumOfUnderHeadBot-
tom+i*headBottomHeight, true, 2);
AddVertex3D(headBottom_nCH[i], 0.5*headBottom[1].myLength,
-0.5*headBottom[1].myWidth, sumOfUnderHeadBottom+i*headBottomHeight);
AddVertex3D(headBottom_nCH[i], 0.5*headBottom[1].myLength, 0, sumOfUnderHeadBot-
tom+i*headBottomHeight);
AddVertex3D(headBottom_nCH[i], 0.5*headBottom[1].myLength,
0.5*headBottom[1].myWidth, sumOfUnderHeadBottom+i*headBottomHeight);
AddVertex3D(headBottom_nCH[i], 0, 0.5*headBottom[1].myWidth, sumOfUnderHeadBot-
tom+i*headBottomHeight);
AddVertex3D(headBottom_nCH[i], -0.5*headBottom[1].myLength,
0.5*headBottom[1].myWidth, sumOfUnderHeadBottom+i*headBottomHeight);
AddVertex3D(headBottom_nCH[i], -0.5*headBottom[1].myLength, 0, sumOfUnderHeadBot-
tom+i*headBottomHeight);
AddVertex3D(headBottom_nCH[i], -0.5*headBottom[1].myLength,
-0.5*headBottom[1].myWidth, sumOfUnderHeadBottom+i*headBottomHeight);

```

```

        AddVertex3D(headBottom_nCH[i], 0, -0.5*headBottom[1].myWidth, sumOfUnderHeadBot-
tom+i*headBottomHeight);
    EndGroup;

    headBottom_groupHandle := LNewObj;

    headBottom_groupHandle := CreateLoftSurfaces(headBottom_groupHandle, bRule, bClose, bSolid);

END;

sumOfUnderHeadBody := sumOfUnderbaseHeight + sumOfBaseHeight + sumOfShaftHeight + sumOfHeadBot-
tomHeight;

FOR i := 1 TO numb_headBody DO BEGIN
    BeginGroup;
    headBody_nCH[i-1] := CreateNurbsCurve(-0.5*headBody[1].myLength, -0.5*headBody[1].myWidth,
sumOfUnderHeadBody+(i-1)*headBodyHeight, true, 1);
    AddVertex3D(headBody_nCH[i-1], 0.5*headBody[1].myLength, -0.5*headBody[1].myWidth,
sumOfUnderHeadBody+(i-1)*headBodyHeight);
    AddVertex3D(headBody_nCH[i-1], 0.5*headBody[1].myLength, 0.5*headBody[1].myWidth,
sumOfUnderHeadBody+(i-1)*headBodyHeight);
    AddVertex3D(headBody_nCH[i-1], -0.5*headBody[1].myLength, 0.5*headBody[1].myWidth,
sumOfUnderHeadBody+(i-1)*headBodyHeight);
    AddVertex3D(headBody_nCH[i-1], -0.5*headBody[1].myLength, -0.5*headBody[1].myWidth,
sumOfUnderHeadBody+(i-1)*headBodyHeight);

    headBody_nCH[i] := CreateNurbsCurve(-0.5*headBody[1].myLength, -0.5*headBody[1].myWidth, su-
mOfUnderHeadBody+i*headBodyHeight, true, 1);
    AddVertex3D(headBody_nCH[i], 0.5*headBody[1].myLength, -0.5*headBody[1].myWidth,
sumOfUnderHeadBody+i*headBodyHeight);
    AddVertex3D(headBody_nCH[i], 0.5*headBody[1].myLength, 0.5*headBody[1].myWidth, su-
mOfUnderHeadBody+i*headBodyHeight);
    AddVertex3D(headBody_nCH[i], -0.5*headBody[1].myLength, 0.5*headBody[1].myWidth,
sumOfUnderHeadBody+i*headBodyHeight);
    AddVertex3D(headBody_nCH[i], -0.5*headBody[1].myLength, -0.5*headBody[1].myWidth,
sumOfUnderHeadBody+i*headBodyHeight);
    EndGroup;

    headBody_groupHandle := LNewObj;

    headBody_groupHandle := CreateLoftSurfaces(headBody_groupHandle, bRule, bClose, bSolid);

END;

sumOfUnderHeadTop := sumOfUnderbaseHeight + sumOfBaseHeight + sumOfShaftHeight + sumOfHeadBot-
tomHeight + sumOfHeadBodyHeight;
averIncOfHeadTopL := (headTop[1].myLength-headTop[2].myLength)/numb_headTop;
averIncOfHeadTopW := (headTop[1].myWidth-headTop[2].myWidth)/numb_headTop;

FOR i := 1 TO numb_headTop DO BEGIN
    BeginGroup;
    headTop_nCH[i-1] := CreateNurbsCurve(-0.5*(headTop[1].myLength-averIncOfHeadTopL*(i-1)),
-0.5*(headTop[1].myWidth-averIncOfHeadTopW*(i-1)), sumOfUnderHeadTop+(i-1)*headTopHeight, true, 1);
    AddVertex3D(headTop_nCH[i-1], 0.5*(headTop[1].myLength-averIncOfHeadTopL*(i-1)),
-0.5*(headTop[1].myWidth-averIncOfHeadTopW*(i-1)), sumOfUnderHeadTop+(i-1)*headTopHeight);
    AddVertex3D(headTop_nCH[i-1], 0.5*(headTop[1].myLength-averIncOfHeadTopL*(i-1)),
0.5*(headTop[1].myWidth-averIncOfHeadTopW*(i-1)), sumOfUnderHeadTop+(i-1)*headTopHeight);
    AddVertex3D(headTop_nCH[i-1], -0.5*(headTop[1].myLength-averIncOfHeadTopL*(i-1)),
0.5*(headTop[1].myWidth-averIncOfHeadTopW*(i-1)), sumOfUnderHeadTop+(i-1)*headTopHeight);
    AddVertex3D(headTop_nCH[i-1], -0.5*(headTop[1].myLength-averIncOfHeadTopL*(i-1)),
-0.5*(headTop[1].myWidth-averIncOfHeadTopW*(i-1)), sumOfUnderHeadTop+(i-1)*headTopHeight);

    headTop_nCH[i] := CreateNurbsCurve(-0.5*(headTop[1].myLength-averIncOfHeadTopL*i),
-0.5*(headTop[1].myWidth-averIncOfHeadTopW*i), sumOfUnderHeadTop+i*headTopHeight, true, 1);
    AddVertex3D(headTop_nCH[i], 0.5*(headTop[1].myLength-averIncOfHeadTopL*i),
-0.5*(headTop[1].myWidth-averIncOfHeadTopW*i), sumOfUnderHeadTop+i*headTopHeight);
    AddVertex3D(headTop_nCH[i], 0.5*(headTop[1].myLength-averIncOfHeadTopL*i),
0.5*(headTop[1].myWidth-averIncOfHeadTopW*i), sumOfUnderHeadTop+i*headTopHeight);
    AddVertex3D(headTop_nCH[i], -0.5*(headTop[1].myLength-averIncOfHeadTopL*i),
0.5*(headTop[1].myWidth-averIncOfHeadTopW*i), sumOfUnderHeadTop+i*headTopHeight);
    AddVertex3D(headTop_nCH[i], -0.5*(headTop[1].myLength-averIncOfHeadTopL*i),
-0.5*(headTop[1].myWidth-averIncOfHeadTopW*i), sumOfUnderHeadTop+i*headTopHeight);
    EndGroup;

    headTop_groupHandle := LNewObj;

    headTop_groupHandle := CreateLoftSurfaces(headTop_groupHandle, bRule, bClose, bSolid);

END;

```

```

sumOfUnderHeadAbove := sumOfUnderbaseHeight + sumOfBaseHeight + sumOfShaftHeight + sumOfHeadBot-
tomHeight + sumOfHeadBodyHeight + sumOfHeadTopHeight;
averIncOfHeadAboveL := (headAbove[1].myLength-headAbove[2].myLength)/numb_headAbove;
averIncOfHeadAboveW := (headAbove[1].myWidth-headAbove[2].myWidth)/numb_headAbove;

```

```

FOR i := 1 TO numb_headAbove DO BEGIN
  BeginGroup;
  headAbove_nCH[i-1] :=
CreateNurbsCurve(-0.5*(headAbove[1].myLength-averIncOfHeadAboveL*(i-1)),
-0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*(i-1)), sumOfUnderHeadAbove+(i-1)*headAboveHeight, true, 1);
  AddVertex3D(headAbove_nCH[i-1],
0.5*(headAbove[1].myLength-averIncOfHeadAboveL*(i-1)), -0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*(i-1)),
sumOfUnderHeadAbove+(i-1)*headAboveHeight);
  AddVertex3D(headAbove_nCH[i-1],
0.5*(headAbove[1].myLength-averIncOfHeadAboveL*(i-1)), 0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*(i-1)),
sumOfUnderHeadAbove+(i-1)*headAboveHeight);
  AddVertex3D(headAbove_nCH[i-1],
-0.5*(headAbove[1].myLength-averIncOfHeadAboveL*(i-1)), 0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*(i-1)),
sumOfUnderHeadAbove+(i-1)*headAboveHeight);
  AddVertex3D(headAbove_nCH[i-1],
-0.5*(headAbove[1].myLength-averIncOfHeadAboveL*(i-1)), -0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*(i-1)),
sumOfUnderHeadAbove+(i-1)*headAboveHeight);
  headAbove_nCH[i] := CreateNurbsCurve(-0.5*(headAbove[1].myLength-averIncOfHeadAboveL*i),
-0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*i), sumOfUnderHeadAbove+i*headAboveHeight, true, 1);
  AddVertex3D(headAbove_nCH[i], 0.5*(headAbove[1].myLength-averIncOfHeadAboveL*i),
-0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*i), sumOfUnderHeadAbove+i*headAboveHeight);
  AddVertex3D(headAbove_nCH[i], 0.5*(headAbove[1].myLength-averIncOfHeadAboveL*i),
0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*i), sumOfUnderHeadAbove+i*headAboveHeight);
  AddVertex3D(headAbove_nCH[i], -0.5*(headAbove[1].myLength-averIncOfHeadAboveL*i),
0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*i), sumOfUnderHeadAbove+i*headAboveHeight);
  AddVertex3D(headAbove_nCH[i], -0.5*(headAbove[1].myLength-averIncOfHeadAboveL*i),
-0.5*(headAbove[1].myWidth-averIncOfHeadAboveW*i), sumOfUnderHeadAbove+i*headAboveHeight);
  EndGroup;

  headAbove_groupHandle := LNewObj;

  headAbove_groupHandle := CreateLoftSurfaces(headAbove_groupHandle, bRule, bClose, bSolid);
END;

```

```

sumOfUnderAntenna := sumOfUnderbaseHeight + sumOfBaseHeight + sumOfShaftHeight + sumOfHeadBot-
tomHeight + sumOfHeadBodyHeight + sumOfHeadTopHeight + sumOfHeadAboveHeight;
averIncOfAntennaL := (antenna[2].myLength - antenna[1].myLength) / numb_antenna;
averIncOfAntennaW := (antenna[2].myWidth - antenna[1].myWidth) / numb_antenna;

```

```

FOR i := 1 TO numb_antenna DO BEGIN
  BeginGroup;
  antenna_nCH[i-1] := CreateNurbsCurve(-0.5*(antenna[1].myLength-averIncOfAntennaL*(i-1)),
-0.5*(antenna[1].myWidth-averIncOfAntennaW*(i-1)), sumOfUnderAntenna +(i-1)*antennaHeight, true, 1);
  AddVertex3D(antenna_nCH[i-1], 0.5*(antenna[1].myLength-averIncOfAntennaL *(i-1)),
-0.5*(antenna[1].myWidth-averIncOfAntennaW *(i-1)), sumOfUnderAntenna +(i-1)*antennaHeight);
  AddVertex3D(antenna_nCH[i-1], 0.5*(antenna[1].myLength-averIncOfAntennaL *(i-1)),
0.5*(antenna[1].myWidth-averIncOfAntennaW *(i-1)), sumOfUnderAntenna +(i-1)*antennaHeight);
  AddVertex3D(antenna_nCH[i-1], -0.5*(antenna[1].myLength-averIncOfAntennaL *(i-1)),
0.5*(antenna[1].myWidth-averIncOfAntennaW *(i-1)), sumOfUnderAntenna +(i-1)*antennaHeight);
  AddVertex3D(antenna_nCH[i-1], -0.5*(antenna[1].myLength-averIncOfAntennaL *(i-1)),
-0.5*(antenna[1].myWidth-averIncOfAntennaW *(i-1)), sumOfUnderAntenna +(i-1)*antennaHeight);
  antenna_nCH[i] := CreateNurbsCurve(-0.5*(antenna[1].myLength-averIncOfAntennaL *i),
-0.5*(antenna[1].myWidth-averIncOfAntennaW *i), sumOfUnderAntenna +i*antennaHeight, true, 1);
  AddVertex3D(antenna_nCH[i], 0.5*(antenna[1].myLength-averIncOfAntennaL *i),
-0.5*(antenna[1].myWidth-averIncOfAntennaW *i), sumOfUnderAntenna +i*antennaHeight);
  AddVertex3D(antenna_nCH[i], 0.5*(antenna[1].myLength-averIncOfAntennaL *i),
0.5*(antenna[1].myWidth-averIncOfAntennaW *i), sumOfUnderAntenna +i*antennaHeight);
  AddVertex3D(antenna_nCH[i], -0.5*(antenna[1].myLength-averIncOfAntennaL *i),
0.5*(antenna[1].myWidth-averIncOfAntennaW *i), sumOfUnderAntenna +i*antennaHeight);
  AddVertex3D(antenna_nCH[i], -0.5*(antenna[1].myLength-averIncOfAntennaL *i),
-0.5*(antenna[1].myWidth-averIncOfAntennaW *i), sumOfUnderAntenna +i*antennaHeight);
  EndGroup;

  antenna_groupHandle := LNewObj;

  antenna_groupHandle := CreateLoftSurfaces(antenna_groupHandle, bRule, bClose, bSolid);
END;
DoMenuTextByName('Standard Views', 8);

```

END;

RUN(paraBuilding_Taipei101);

Seong Ki LEE



WS0607 MAS CAAD Module01 : 49 Skyscraper