```
allForce (generic function with 1 method)
```

```julia
##################calculate all particles forces##################
function allForce(n1::Integer,n2::Integer,ωbTmp::Float64, modelTmp::FrequencySimulation, coefData::Matrix{ComplexF64})
coefDataTmp=deepcopy(coefData);
simModelTmp=modelTmp;
pNoTmp=length(simModelTmp.particles);
dimensionTmp=typeof(simModelTmp.source.medium).parameters[2];
allForceTmp=Array{Float64}(undef,0)
for iTmp in 1:pNoTmp
    forceTmp=force(iTmp,n1,n2,ωbTmp, modelTmp, coefDataTmp);
    allForceTmp=push!(allForceTmp,forceTmp[1],forceTmp[2],forceTmp[3]);
end
return allForceTmp
end
```

```
forcePackLow (generic function with 1 method)
```

```julia
function forcePackLow(RsTmp::Float64,parPos::Matrix{Float64})
parPosTmp=deepcopy(parPos);
freqIn=40000;
ω=2.0*π*freqIn;
dimension=3;
ρb=1.225;
ρp=29.0;
cb=343.0;
cp=900.0;
pNo=length(parPosTmp[:,1]);
coefOrder=6;
modelTmp=buildModelProto(dimension, ρb, cb, ρp, cp, ω, pNo, RsTmp, parPosTmp)
coefData=getCoefProto(ω, modelTmp, coefOrder);
forceTmp=allForce(24,48,ω, modelTmp, coefData);
return forceTmp
end
```

```
forcePackMiddle (generic function with 1 method)
```

```julia
function forcePackMiddle(RsTmp::Float64,parPos::Matrix{Float64})
parPosTmp=deepcopy(parPos);
freqIn=40000;
ω=2.0*π*freqIn;
dimension=3;
ρb=1.225;
ρp=29.0;
cb=343.0;
cp=900.0;
pNo=length(parPosTmp[:,1]);
coefOrder=8;
modelTmp=buildModelProto(dimension, ρb, cb, ρp, cp, ω, pNo, RsTmp, parPosTmp)
coefData=getCoefProto(ω, modelTmp, coefOrder);
forceTmp=allForce(30,60,ω, modelTmp, coefData);
return forceTmp
end
```

```
forcePackHigh (generic function with 1 method)
```

```julia
function forcePackHigh(RsTmp::Float64,parPos::Matrix{Float64})
parPosTmp=deepcopy(parPos);
freqIn=40000;
ω=2.0*π*freqIn;
dimension=3;
ρb=1.225;
ρp=29.0;
cb=343.0;
cp=900.0;
pNo=length(parPosTmp[:,1]);
coefOrder=10;
modelTmp=buildModelProto(dimension, ρb, cb, ρp, cp, ω, pNo, RsTmp, parPosTmp)
coefData=getCoefProto(ω, modelTmp, coefOrder);
forceTmp=allForce(36,72,ω, modelTmp, coefData);
return forceTmp
end
```

```
forcePackExtraHigh (generic function with 1 method)
```

```julia
function forcePackExtraHigh(RsTmp::Float64,parPos::Matrix{Float64})
parPosTmp=deepcopy(parPos);
freqIn=40000;
ω=2.0*π*freqIn;
dimension=3;
ρb=1.225;
ρp=29.0;
cb=343.0;
cp=900.0;
pNo=length(parPosTmp[:,1]);
coefOrder=14;
modelTmp=buildModelProto(dimension, ρb, cb, ρp, cp, ω, pNo, RsTmp, parPosTmp)
coefData=getCoefProto(ω, modelTmp, coefOrder);
forceTmp=allForce(54,108,ω, modelTmp, coefData);
return forceTmp
end
```