1.TP0

```
#include <stdio.h>
#include <stdlib.h>
int main(int /*argc*/, char ** /*argv*/)
 {  printf("Hello World");
    getchar();
    return 0; }
```

2.TP1

Sphere

```
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
#include <vtkSphereSource.h>
#include <vtkPolyDataMapper.h>
#include <vtkActor.h>
 #include <vtkProperty.h>
 #include <vtkRenderWindowInteractor.h>

int main()
{     vtkRenderer *renderer = vtkRenderer::New();
   renderer->SetBackground(0.8,0.7,0.1); //couleur d'arriere plan est beige
    vtkRenderWindow *renderWindow = vtkRenderWindow::New();
  renderWindow->AddRenderer(renderer);  //an interactor
   vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
   iren->SetRenderWindow(renderWindow);
  vtkSphereSource *sphere = vtkSphereSource::New();
  sphere->SetRadius(1.0);
   sphere->SetThetaResolution(18);
   sphere->SetPhiResolution(18);    // graphics library
    vtkPolyDataMapper *map = vtkPolyDataMapper::New();
   map->SetInput(sphere->GetOutput());
 vtkActor *aSphere = vtkActor::New();
```

```
 aSphere->SetMapper(map);  }
```

Cylindre :

```
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
 #include <vtkPolyDataMapper.h>
#include <vtkActor.h>
 #include <vtkProperty.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkCylinderSource.h>

int main() {     vtkRenderer *renderer = vtkRenderer::New();
   renderer->SetBackground(0.8,0.7,0.1);
    vtkRenderWindow *renderWindow = vtkRenderWindow::New();
   renderWindow->AddRenderer(renderer);
   vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
   iren->SetRenderWindow(renderWindow);
 vtkCylinderSource *cylinder = vtkCylinderSource::New();
  cylinder->SetCenter(0.0, 0.0, 0.0);
 cylinder->SetRadius(5.0);
  cylinder->SetHeight(7.0);
 cylinder->SetResolution(100);

   //graphics library    vtkPolyDataMapper *map = vtkPolyDataMapper::New();
     map->SetInput(cylinder->GetOutput());
 vtkActor *aCylinder = vtkActor::New();
   aCylinder->SetMapper(map);
 aCylinder->GetProperty()->SetColor(0,0,1); //cylindre color blue
  //add the actor to the scene
renderer->AddActor(aCylinder);
renderWindow->Render();
  iren->Start();
```

```
   return 0; }
```

Cube :

```
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
 #include <vtkPolyDataMapper.h>
#include <vtkActor.h>
 #include <vtkProperty.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkCubeSource.h>

int main() {     vtkRenderer *renderer = vtkRenderer::New();
  renderer->SetBackground(0.8,0.7,0.1);
   vtkRenderWindow *renderWindow = vtkRenderWindow::New();
  renderWindow->AddRenderer(renderer);    //an interactor
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renderWindow);
vtkCubeSource *cube = vtkCubeSource::New();
 cube->SetCenter(0.0, 0.0, 0.0);
cube->SetXLength(7.0);
  cube->SetYLength(7.0);
cube->SetZLength(7.0);

   // graphics library
    vtkPolyDataMapper *map = vtkPolyDataMapper::New();
    map->SetInput(cube->GetOutput());
  vtkActor *aCube = vtkActor::New();
    aCube->SetMapper(map);
  aCube->GetProperty()->SetColor(0,0,1); //cube color blue
  //add the actor to the scene
    renderer->AddActor(aCube);
```

```
 renderWindow->Render();
 iren->Start();
  return 0; }
```

Cône :

```
 #include <vtkRenderWindow.h>
 #include <vtkRenderer.h>
#include <vtkPolyDataMapper.h>
 #include <vtkActor.h>
 #include <vtkProperty.h>
 #include <vtkRenderWindowInteractor.h>
 #include <vtkConeSource.h>

int main() {    vtkRenderer *renderer = vtkRenderer::New();
   renderer->SetBackground(0.8,0.7,0.1);
 //couleur d'arriere plan
 vtkRenderWindow *renderWindow = vtkRenderWindow::New();
    renderWindow->AddRenderer(renderer);
    //an interactor     vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
   iren->SetRenderWindow(renderWindow);
    vtkConeSource *cone = vtkConeSource::New();

  cone->SetResolution(100);
     cone->SetCenter(0.0, 7.0, 0.0);
   cone->SetResolution(100);
  cone->SetHeight (7.0);
   cone->SetRadius(6.0);
   cone->SetDirection(0,90,0);

   //graphics library
 vtkPolyDataMapper *map = vtkPolyDataMapper::New();
```

```
    map->SetInput(cone->GetOutput()) ;
   vtkActor *aCone = vtkActor::New();
   aCone->SetMapper(map);
 aCone->GetProperty()->SetColor(1,0,0); //cone color rouge    //add the actor to the scene
renderer->AddActor(aCone);
  renderWindow->Render();
   iren->Start();
   return 0; }
```

Cône + Cylindre :

```
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
#include <vtkPolyDataMapper.h>
#include <vtkActor.h>
#include <vtkProperty.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkCylinderSource.h>
#include <vtkConeSource.h>

int main() {    vtkRenderer *renderer = vtkRenderer::New();
   renderer->SetBackground(0.8,0.7,0.1);
vtkRenderWindow *renderWindow = vtkRenderWindow::New();
   renderWindow->AddRenderer(renderer);
 //an interactor    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
 iren->SetRenderWindow(renderWindow);
vtkConeSource *cone = vtkConeSource::New();

 cone->SetResolution(100);
 cone->SetCenter(0.0, 7.0, 0.0);
 cone->SetResolution(100);
 cone->SetHeight (7.0);
cone->SetRadius(6.0);
```

```
cone->SetDirection(0,90,0);
 vtkCylinderSource *cylinder = vtkCylinderSource::New();
   cylinder->SetCenter(0.0, 0.0, 0.0);
  cylinder->SetRadius(6.0);
   cylinder->SetHeight(7.0);
  cylinder->SetResolution(100);
   // graphics library
 vtkPolyDataMapper *map = vtkPolyDataMapper::New();
     vtkPolyDataMapper *map1 = vtkPolyDataMapper::New();
  map->SetInput(cone->GetOutput());
  vtkActor *aCone = vtkActor::New();
    aCone->SetMapper(map);
  aCone->GetProperty()->SetColor(1,0,0);
renderer->AddActor(aCone);
   map1->SetInput(cylinder->GetOutput());
  vtkActor *aCylinder = vtkActor::New();
  aCylinder->SetMapper(map1);
 aCylinder->GetProperty()->SetColor(0,0,1);
renderer->AddActor(aCylinder);
renderWindow->Render();
  iren->Start();
 return 0; }


  aSphere->GetProperty()->SetColor(0,0,1);
  renderer->AddActor(aSphere);
  renderWindow->Render();
  iren->Start();
   return 0; }
```