```
/*-----------------------------------------------------------------
*
* This code is based on the original UCLA model, and is modified
* by CIRCS group of Northeastern University.
*
* Contact Information:
*
* Center for interdisciplinary research on complex systems
* Departments of Physics, Northeastern University
*
* Alain Karma           a.karma (at) northeastern.edu
* Mingwang Zhong        mingwang.zhong (at) gmail.com
*
* The code was used to reproduce simulations in
* Transient outward K+ current (Ito) underlies the right ventricular
* initiation of polymorphic ventricular tachycardia in a transgenic
* rabbit model of long QT type 1, Bum-Rak Choi, Weiyan Li, Dmitry
* Terentyev, Anatoli Kabkov, Mingwang Zhong, Colin M Rees, Radmila
* Terentyeva, Tae Yun Kim, Zhilin Qu, Xuwen Peng, Alain Karma,
* and Gideon Koren (2018).
*----------------------------------------------------------------- */

// Information of original UCLA model:
/*--------------------- UCLA Model ver 1.00 ----------------------
*
* Contact Information
*
* Departments of Medicine (Cardiology)
* David Geffen School of Medicine at UCLA
*
* Daisuke Sato           dasato (at) mednet.ucla.edu
* Yohannes Shiferaw      yshiferaw (at) csun.edu
* James N Weiss          JWeiss (at) mednet.ucla.edu
*
* The code was used to produce simulations in
* A. Mahajan, Y. Shiferaw, D. Sato, A. Baher, R. Olcese, L.-H. Xie,
* M.-J. Yang, P.-S. Chen, J. G. Restrepo, A. Karma, A. Garfinkel,
* Z. Qu, and J. N. Weiss, A rabbit ventricular action potential model
* replicating cardiac dynamics at rapid heart rates, Biophysical Journal,
* 94 (2008), pp. 392¿410.
*----------------------------------------------------------------- */

#include <iostream>
#include <cstdlib>
#include <fstream>
#include <cmath>
#include <queue>

#define ___USE_VAR_FOR_CONST
#include "cell.cpp"

using namespace std;



int main( int argc, char** argv)
{

        CCell *cell;
        cell = new CCell; // creat the object with name "cell", with class type "CCe
ll"

        double BCL=4000; // pacing cycle length
        int NBeat = 5 + 15*(BCL<2000); // number of beats before output
        int step_wait = (int)( 200/(cell->getdt())+0.5 ); // waiting time at the beg
inning, 200ms
        int TotalStep = (int)( BCL*(NBeat+3)/(cell->getdt())+0.5 ) + step_wait;
        int step_1ms = (int)( 1/(cell->getdt())+0.5 ); // number of steps in 1ms
        int step_BCL = BCL*step_1ms; // number of steps in 1 BCL
        int step_output = (int)( 1/(cell->getdt())+0.5 ); // number of steps for out
put

        ofstream currsfile("output.txt",ios::out);
```

```
        for (int step=0; step<TotalStep; step++)
        {
                double t = step*cell->getdt();
                ////////////////////// Normal pacing or voltage clamp /////////////
/////////
                if( argc < 9 ) // normal pacing
                {
                        if ( (step-step_wait)%step_BCL < step_1ms && (step-step_wait
)>0 )
                                cell->Pace(50.0);
                        else
                                cell->Pace();
                }
                else  // voltage clamp
                {
                        if ( t < BCL*NBeat+100 )
                                cell->PaceVClamp(-50);
                        else if ( t < BCL*NBeat+600 )
                                cell->PaceVClamp(atof(argv[9]));
                        else if ( t < BCL*NBeat+1000 )
                                cell->PaceVClamp(0);
                        else
                                cell->PaceVClamp(-86);
                }

                ////////////////////// output to file ////////////////////////////
////
                if (t>=BCL*NBeat && step%step_output==0)
                {
                        // currsfile << t-BCL*NBeat << " " << cell->v << " " << endl
;
                        // << cell->ci << " " << cell->cj << endl;

                        currsfile << t-BCL*NBeat << " " << cell->v << " "
                                        << cell->_ical << " " << cell->_ikr << " "
                                        << cell->_itof << " " << cell->_itos << " "
                                        << cell->_inaca << " " << cell->_inak << " "
                                        << cell->_ik1 << " " << cell->_ina << " "

                                        << cell->_iks << " " << cell->_svipca << " "
                                        << cell->_iup << " " << cell->_ir << " "
                                        << cell->cs << " " << cell->ci << " "
                                        << cell->cjp << " " << cell->cj << " "
                                        << cell->ica <<" " << cell->fspark << " "

                                        << cell->hd << " " << cell->hf << " "
                                        << cell->hf_ca << " " << cell->IKrO << " "
                                        << cell->IKrI << " " << endl;
                }
        }

        delete cell;
        currsfile.close();

        return 0;
}
```