COORD randCoordOne;

COORD randCoordTwo;

randCoordOne.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordOne.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

do

{

randCoordTwo.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordTwo.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

}

while(randCoordTwo.X > (randCoordOne.X - fireWorksVar.area.X) &&

randCoordTwo.X < (randCoordOne.X + fireWorksVar.area.X));

/\*\*two fire works\*\*/

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[4], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[5], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[6], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[7], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[8], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[4], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

ClearArea(hStdOut, clearSpaceArr, clearSpaceSize, randCoordOne, fireWorksVar.area);

do

{

randCoordOne.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordOne.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

}

while(randCoordOne.X > (randCoordTwo.X - fireWorksVar.area.X) &&

randCoordOne.X < (randCoordTwo.X + fireWorksVar.area.X));

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[5], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[6], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[7], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[8], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

ClearArea(hStdOut, clearSpaceArr, clearSpaceSize, randCoordTwo, fireWorksVar.area);

do

{

/\*\*second loop\*\*/

do

{

randCoordTwo.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordTwo.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

}

while(randCoordTwo.X > (randCoordOne.X - fireWorksVar.area.X) &&

randCoordTwo.X < (randCoordOne.X + fireWorksVar.area.X));

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[4], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[5], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[6], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[8], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

ClearArea(hStdOut, clearSpaceArr, clearSpaceSize, randCoordOne, fireWorksVar.area);

do

{

randCoordOne.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordOne.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

}

while(randCoordOne.X > (randCoordTwo.X - fireWorksVar.area.X) &&

randCoordOne.X < (randCoordTwo.X + fireWorksVar.area.X));

AddObject(hStdOut, fireWorksArr[4], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[5], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[6], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[7], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[8], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[4], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

ClearArea(hStdOut, clearSpaceArr, clearSpaceSize, randCoordTwo, fireWorksVar.area);

do

{

randCoordTwo.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordTwo.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

}

while(randCoordTwo.X > (randCoordOne.X - fireWorksVar.area.X) &&

randCoordTwo.X < (randCoordOne.X + fireWorksVar.area.X));

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[5], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[6], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[7], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordTwo);

AddObject(hStdOut, fireWorksArr[8], fireWorksVar.area, randCoordOne);

Sleep(sleepTime);

ClearArea(hStdOut, clearSpaceArr, clearSpaceSize, randCoordOne, fireWorksVar.area);

/\*\*third loop\*\*/

do

{

randCoordOne.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordOne.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

}

while(randCoordOne.X > (randCoordTwo.X - fireWorksVar.area.X) &&

randCoordOne.X < (randCoordTwo.X + fireWorksVar.area.X));

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[4], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[5], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[6], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[8], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

ClearArea(hStdOut, clearSpaceArr, clearSpaceSize, randCoordTwo, fireWorksVar.area);

do

{

randCoordTwo.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordTwo.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

}

while(randCoordTwo.X > (randCoordOne.X - fireWorksVar.area.X) &&

randCoordTwo.X < (randCoordOne.X + fireWorksVar.area.X));

AddObject(hStdOut, fireWorksArr[4], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[5], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[6], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[7], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[8], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[4], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

ClearArea(hStdOut, clearSpaceArr, clearSpaceSize, randCoordOne, fireWorksVar.area);

do

{

randCoordOne.X = (rand() % (static\_cast<int>((numX - fireWorksVar.area.X) \* .8))) + ((numX - fireWorksVar.area.X) \* .1);

randCoordOne.Y = (rand() % (static\_cast<int>(numY \* .33))) + 1;

}

while(randCoordOne.X > (randCoordTwo.X - fireWorksVar.area.X) &&

randCoordOne.X < (randCoordTwo.X + fireWorksVar.area.X));

AddObject(hStdOut, fireWorksArr[0], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[5], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[1], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[6], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[2], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[7], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

AddObject(hStdOut, fireWorksArr[3], fireWorksVar.area, randCoordOne);

AddObject(hStdOut, fireWorksArr[8], fireWorksVar.area, randCoordTwo);

Sleep(sleepTime);

ClearArea(hStdOut, clearSpaceArr, clearSpaceSize, randCoordTwo, fireWorksVar.area);

}

while (true);