

Nie do końca rozumiałem jak mam zrobić „drivera”. W celu sprawdzenia czy algorytmy poprawnie sortują wykorzystałem poniższą funkcję, znajdującą się w pliku Vector.cpp

```
bool CheckCorrect()
{
    int Counter = 0;
    int i;
    for(i=0;i<Y_Dime-1;i++)
    {
        if(Column[i]<=Column[i+1]){++Counter;}
    }
    if(Counter == Y_Dime-1)
    {
        return true;
    }
    else
    {
        cout<<"Bład na  " <<Counter<<endl;
        return false;
    }
}
```

Chcąc ją wywołać używałem `cout<<Mat[i].CheckCorrect();` umieszczając w pętli for poszczególnych plikach zawierających algorytmy sortowania, np. Quicksort.cpp jak widać na zdjęciu poniżej. Ukazany na filmach ciąg jedynek oznacza logiczną jedynkę, czyli prawdę, co potwierdza, że tablice sortują się poprawnie.

```
void ExecutionQuickSort(Matrix<Type> &Mat)
{
    chrono::steady_clock Clock;
    int DimMatrix = Mat.GetX_Dime();
    int DimVector = Mat[0].GetY_Dime();           //Zakladamy ze rozmiar jest taki sam
    int i;
    auto Start = Clock.now();
    for(i=0;i<DimMatrix;i++)
    {
        QuickSort(Mat[i],0,DimVector); //wywołujemy dla wszystkich komorek macierzy
    }
    auto End = Clock.now();
    auto Time_delay = static_cast<chrono::duration<double>>(End - Start);
    ShowStatistic(QuickType,Mat,Time_delay);    //wyswietlamy statystyke
}
```