

plot.h

```
#ifndef PLOT_H
#define PLOT_H
using namespace std;
class Plot {
private:
    int ny1;
public:
    Plot(int numy1 = 62)
    {
        ny1 = numy1; // constructor
        cout << "plot width: " << ny1 << "chars\n";
    }
    void minmax(float y[], int n,
                float& ymin, float& ymax);
    int cyfun(float y, int ny,
              float ymin, float ymax);
    void plotarrays(float x[], float y[], int n,
                   float ymin, float ymax);
    void plotarrays(float x[], float y[], int n);
};



---


#endif
```

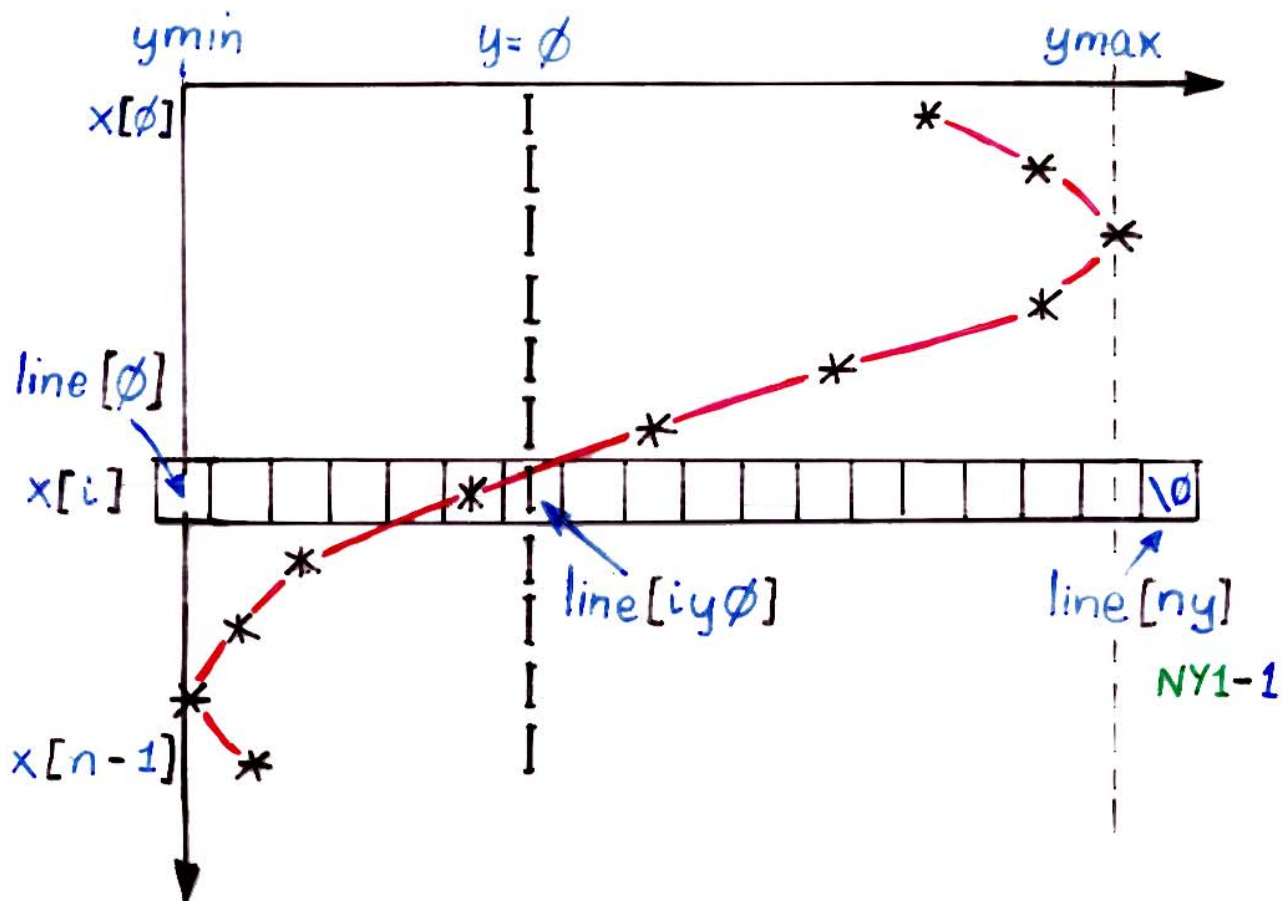
plot.h

```
#ifndef PLOT_H
#define PLOT_H
using namespace std;
class Plot {
private:
    int ny1;
public:
    Plot(int numy1 = 62)
    {
        ny1 = numy1; // constructor
        cout << "plot width: " << ny1 << "chars\n";
    }
    void minmax(float y[], int n,
                float& ymin, float& ymax);
};
```

```
void Plot::minmax(float y[], int n,
                  float& ymin, float& ymax)
{
    ymin = y[0];
    ymax = y[0];
    for (int i = 1; i < n; i++)
        if (y[i] < ymin)
            ymin = y[i];
        else if (y[i] > ymax)
            ymax = y[i];
}
#endif
```

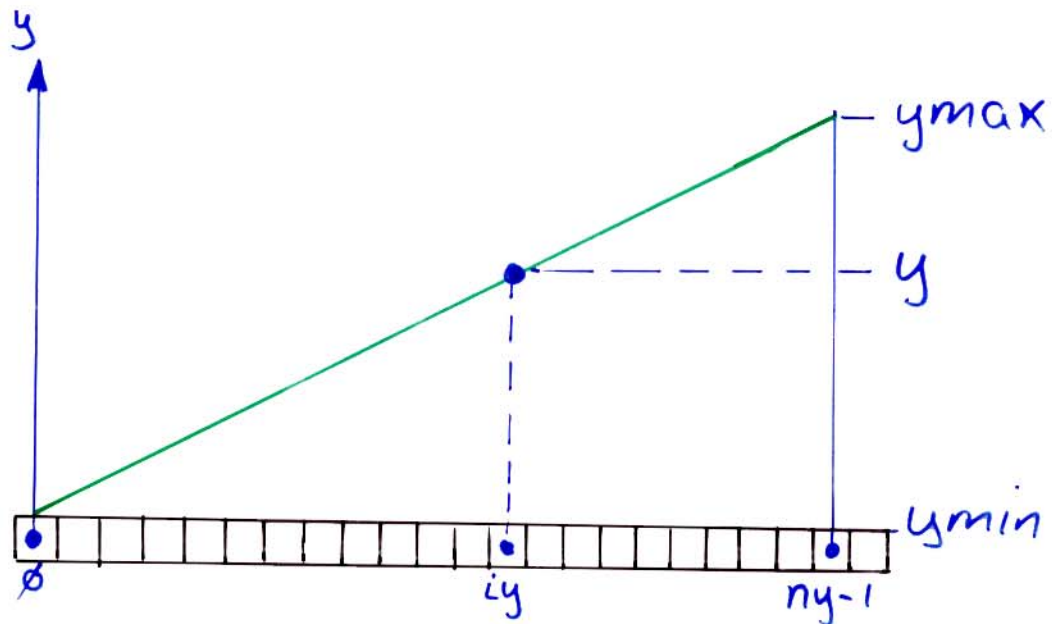
```
void Plot::plotarrays(float x[], float y[],
                    int n, float ymin, float ymax)
```

```
{
    const char blank = ' ';
    const char star = '*';
    const char ichar = 'I';
    const char tab = '\t';
    int i, iy0, iy, ny;
    char line[ny+1];
```



```
cout << "min" << ymin << ", max"
      << ymax << endl;
```

```
}
```



$$\frac{(i_y - 0)}{(n_y - 1) - 0} = \frac{(y - y_{\min})}{(y_{\max} - y_{\min})}$$

```

int Plot::iyfun(float y, int ny,
               float ymin, float ymax)
{
    int iy = static_cast<int>((ny - 1) *
                              (y - ymin) / (ymax - ymin) + 0.5);
    return iy;
}

```

plot.h

```
#ifndef PLOT_H
#define PLOT_H
using namespace std;
class Plot {
private:
    int ny1;
public:
    Plot(int numy1 = 62)
    {
        ny1 = numy1; // constructor
        cout << "plot width: " << ny1 << "chars\n";
    }
    void minmax(float y[], int n,
                float& ymin, float& ymax);
    int iyfun(float y, int ny,
              float ymin, float ymax);
};

int Plot::iyfun(float y, int ny,
                float ymin, float ymax)
{
    int iy = static_cast<int>((ny - 1) *
                              (y - ymin) / (ymax - ymin) + 0.5);
    return iy;
}

#endif
```

```
void Plot::plotarrays(float x[], float y[],  
                    int n, float ymin, float ymax)
```

```
{  
    const char blank = ' ';
```

```
    const char star = '*';
```

```
    const char ichar = 'I';
```

```
    const char tab = '\t';
```

```
    int i, iy0, iy, ny;
```

```
    char line[ny1];
```

```
    ny = ny1 - 1;
```

```
    line[ny] = '\0';
```

```
    iy0 = iyfun(0, ny, ymin, ymax);
```

```
    for (i = 0; i < n; i++) {
```

```
        for (iy = 0; iy < ny; iy++)
```

```
            line[iy] = blank;
```

```
        line[iy0] = ichar;
```

```
        iy = iyfun(y[i], ny, ymin, ymax);
```

```
        line[iy] = star;
```

```
        cout << x[i] << tab << line << endl;
```

```
    }
```

```
    cout << "min" << ymin << ", max"
```

```
        << ymax << endl;
```

```
}
```