

```

class Rocket { // copied from rockfile
    void funarrays(float tmax, float time[],
                  float vel[], float ht[], int npoints);
    void printarrays(float [], float [], float [], int);
    void savearrays(float [], float [], float [], int);
};
void getrocket(float& ms, float& vt, ..... );
int const SIZE = 100;


---


int main()
{
    float time[SIZE], vel[SIZE] ht[SIZE];
    float tmax;
    int npoints;
    cout << "enter elapsed time from liftoff\n";
    cin >> tmax;
    cout << "time entered is " << tmax << "secs\n";
    cout << "enter number of points\n";
    cin >> npoints;
    cout << "number of points: " << npoints << endl;

    myRocket.funarrays(tmax, time,
                       vel, ht, npoints);

    myRocket.printarrays(time, vel, ht, npoints);
    myRocket.savearrays(time, vel, ht, npoints);
    return 0;
}

```

```
void Rocket::funarrays (float tmax,  
    float time[], float vel[], float ht[],  
    int npoints )  
{  
    float delt = tmax / (npoints - 1);  
    for (int i = 0; i < npoints; i++) {  
        time[i] = i * delt;  
        findvelht (time[i], vel[i], ht[i]);  
    }  
}
```

```
void Rocket::printarrays (float time[],  
    float vel[], float ht[], int npoints )  
{  
    cout << "time(s) velocity(m/s) height(m)\n";  
    for (int i = 0; i < npoints; i++)  
        cout << time[i] << vel[i] << ht[i] << endl;  
}
```

```
void Rocket::savearrays(float time[],  
                        float vel[], float ht[], int npoints)  
{  
    char const tab = '\t';  
    ofstream outfile("rocket.out");  
    assert(outfile);  
    for(int i = 0; i < npoints; i++)  
        outfile << time[i] << tab << vel[i]  
                << tab << ht[i] << endl;  
    cout << npoints << " rows data to file\n";  
    outfile.close();  
}
```