

Exercise 3

Introduction to Computational Astrophysics, SoSe 2024

Seha Lee

Review

Solution. 1. (2) `main(){}
2. (2) for (int i=0; i<n ; ++i)`



Task 1. Importing data into arrays

Solution.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     double x[100], y[100];
6     int n;
7
8     // input data
9     cout << "How many data pairs?(max 100) ";
10    cin >> n;
11    if(n > 100) return 1;
12
13    cout << "Enter your pairs (x,y) " << endl;
14    for (int i = 0; i < n; i++) {
15        cout << "x[" << i << "], y[" << i << "]: ";
16        cin >> x[i] >> y[i];
17    }
18
19    cout << "Imported Pairs" << endl;
20    cout << "x\t-\ty" << endl;
21    for (int i = 0; i < n; i++) {
22        cout << x[i] << "\t-\t" << y[i] << endl;
23    }
24
25    double sum_x = 0, sum_y = 0, sum_xy = 0, sum_xx = 0;
26    // line of best fit
27    for (int i = 0; i < n; i++) {
28        sum_x += x[i];
```

```

29     sum_y += y[i];
30     sum_xy += x[i] * y[i];
31     sum_xx += x[i] * x[i];
32 }
33 double mean_x = sum_x / n;
34 double mean_y = sum_y / n;
35 double b = (sum_xy - n * mean_x * mean_y) / (sum_xx - n * mean_x
    * mean_x);
36 double a = mean_y - b * mean_x;
37
38 cout << "a = " << a << ", b = " << b << endl;
39 return 0;
40 }

```

□

Task 2. Pointers, references, arrays

Solution. a)

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     double radius;
6     double &diameterReference = radius;
7     double *diameterPointer = &radius;
8     cout << "Radius? ";
9     cin >> radius;
10    *diameterPointer *= 2;
11    cout << "Diameter = " << diameterReference << endl;
12 }

```

b) In this way the pointer does not points to an arbitrary unknown memory location.

c) Normally "" would be added;"parray = array;"

d) "parray[i]" means "*(parray + i)", which accesses the data i elements beyond the pointer's initial point.

e) As in d), "parray[2] = 6 ;" accesses two elements beyond the new pointer location set at array[3] in (5), resulting in array[5] being set to 6.

f) array[5] = 6.

□

Task 3. Catching invalid input

Solution.

```

1 #include <iostream>
2 #include <cmath>
3 #include <limits>
4 using namespace std;
5
6 int main() {
7     const double tempSun = 5778;

```

```

8     double logLsun;
9     double Teff;
10
11     cout << "Enter log(L*/Lsun): ";
12     cin >> logLsun;
13     cout << "Enter Teff (in Kelvin): ";
14     cin >> Teff;
15     while (Teff <= 0) {
16         cout << "Invalid input. Please enter a positive value for
17             Teff: ";
18         cin.clear();
19         cin.ignore(numeric_limits<streamsize>::max(), '\n');
20         cin >> Teff;
21     }
22     double radius = sqrt(pow(10, logLsun) * pow(tempSun / Teff, 4));
23     cout << "The radius is: " << radius << " R_sun" << endl;
24     return 0;
25 }

```

□