高性能并行计算第 1 次作业

姓名: 张子栋　　　学号: 2020317210101

代码地址: `/home/2020317210101/work1` `https://github.com/Bluuur/MarkdownNotes/tree/main/高性能并行计算/Code1`

实验结果

1. 计算 N 维数据的最大值, 最小值, 平均值, 标准差

代码

```c
//
// Created by ZidongZh on 2022/9/23.
//

#include "stdio.h"
#include "math.h"

double getMax(double array[]) {
    double max = array[0];
    for (int i = 0; i < ((int) sizeof(array) / sizeof(*array)); ++i) {
        if (array[i] >= max) {
            max = array[i];
        }
    }
    return max;
}

double getMin(double array[]) {
    double min = array[0];
    for (int i = 0; i < ((int) sizeof(array) / sizeof(*array)); ++i) {
        if (array[i] <= min) {
            min = array[i];
        }
    }
    return min;
}

double getMean(double array[]) {
    int len = (int) sizeof(array) / sizeof(*array);
    double sum = 0.0;
    double mean = 0.0;
    for (int i = 0; i < len; ++i) {
        sum += array[i];
    }
    mean = sum / len;
    return mean;
}

double getSD(double array[]) {
    int len = (int) sizeof(array) / sizeof(*array);
    double mean = getMean(array);
    double SS = 0.0;
    for (int i = 0; i < len; ++i) {
        SS += pow(array[i] - mean, 2);
    }
    double SD = sqrt(SS / (len - 1));
    return SD;
}

int main() {

    // Get the length of data
    int length;

    printf("Enter the size of data:");
    scanf("%d", &length);

    // init array, get data
    double array[length];

    for (int i = 0; i < length; ++i) {
        printf("Enter data (%d of %d):", i + 1, length);
        scanf("%lf", &array[i]);
    }

    // Display data
    printf("Entered data:");

    for (int i = 0; i < length; ++i) {
        printf("%lf ", array[i]);
    }

    // Output
    printf("\nMax of data is: %lf\n", getMax(array));
    printf("Min of data is: %lf\n", getMin(array));
    printf("Mean of data is: %lf\n", getMean(array));
    printf("SD of data is: %lf\n", getSD(array));
}
```

结果

```
Enter the size of data:4
Enter data (1 of 4):12.6
Enter data (2 of 4):15.4
Enter data (3 of 4):19.8
Enter data (4 of 4):11.5
Entered data:12.600000 15.400000 19.800000 11.500000
Max of data is: 12.600000
Min of data is: 12.600000
Mean of data is: 12.600000
SD of data is: 9.970886
```

2. 计算 N 维向量点乘

代码:

```
//
// Created by ZidongZh on 2022/9/23.
//

#include <stdio.h>

int main() {
    int dimension;

    // Get the dimension of array
    printf("Enter the dimension of the two vectors:");
    scanf("%d", &dimension);

    // Initialize array
    double array1[dimension];
    double array2[dimension];

    // Get data
    for (int i = 0; i < dimension; ++i) {
        printf("Enter data (vector1, %d of %d):", i + 1, dimension);
        scanf("%lf", &array1[i]);
```

```c
    }
    printf("--------------------------\n");
    for (int i = 0; i < dimension; ++i) {
        printf("Enter data (vector2, %d of %d):", i + 1, dimension);
        scanf("%lf", &array2[i]);
    }

    // Display
    printf("\nVector 1:\n");
    for (int i = 0; i < dimension; ++i) {
        printf("%lf \n", array1[i]);
    }
    printf("\nVector 2:\n");
    for (int i = 0; i < dimension; ++i) {
        printf("%lf\n", array2[i]);
    }

    // Compute & Output
    double result = 0;
    for (int i = 0; i < dimension; ++i) {
        result += (array1[i] * array2[i]);
    }

    printf("\nDot product of the two vectors is %lf", result);

    return 0;
}

```

结果

```
Enter the dimension of the two vectors:4
Enter data (vector1, 1 of 4):1
Enter data (vector1, 2 of 4):2
Enter data (vector1, 3 of 4):3
Enter data (vector1, 4 of 4):4
--------------------------
Enter data (vector2, 1 of 4):5
Enter data (vector2, 2 of 4):6
Enter data (vector2, 3 of 4):7
Enter data (vector2, 4 of 4):8

Vector 1:
1.000000
```

```
2.000000
3.000000
4.000000

Vector 2:
5.000000
6.000000
7.000000
8.000000

Dot product of the two vectors is 70.000000
```

3. 计算 N 维矩阵点乘

```c
//
// Created by ZidongZh on 2022/9/23.
//

#include <stdio.h>

int main() {
    int dimension;

    // Get the dimension of array
    printf("Enter the dimension of the matrix:");
    scanf("%d", &dimension);

    // Initialize array
    double matrix1[dimension][dimension];
    double matrix2[dimension][dimension];

    //
    for (int i = 0; i < dimension; ++i) {
        for (int j = 0; j < dimension; ++j) {
            printf("Enter the number of matrix1 at row %d, column %d:", i + 1, j + 1);
            scanf("%lf", &matrix1[i][j]);
        }
    }
    printf("------------------------------\n");
    for (int i = 0; i < dimension; ++i) {
        for (int j = 0; j < dimension; ++j) {
            printf("Enter the number of matrix2 at row %d, column %d:", i + 1, j + 1);
            scanf("%lf", &matrix2[i][j]);
        }
    }

    printf("\nMatrix 1:\n");
    for (int i = 0; i < dimension; ++i) {
        for (int j = 0; j < dimension; ++j) {
            printf("%lf ", matrix1[i][j]);
        }
        printf("\n");
    }
    printf("\nMatrix 2:\n");
    for (int i = 0; i < dimension; ++i) {
        for (int j = 0; j < dimension; ++j) {
            printf("%lf ", matrix2[i][j]);
        }
        printf("\n");
    }

    // Compute
    double result = 0;
    for (int i = 0; i < dimension; ++i) {
        for (int j = 0; j < dimension; ++j) {
            result += (matrix1[i][j] * matrix2[i][j]);
        }
    }

    printf("\nDot product of the two vectors is %lf", result);
    return 0;
}
```

结果

```
Enter the dimension of the matrix:2
Enter the number of matrix1 at row 1, column 1:1
Enter the number of matrix1 at row 1, column 2:2
Enter the number of matrix1 at row 2, column 1:3
Enter the number of matrix1 at row 2, column 2:4
------------------------------
Enter the number of matrix2 at row 1, column 1:5
Enter the number of matrix2 at row 1, column 2:6
Enter the number of matrix2 at row 2, column 1:7
Enter the number of matrix2 at row 2, column 2:8

Matrix 1:
1.000000 2.000000
```

```
3.000000 4.000000

Matrix 2:
5.000000 6.000000
7.000000 8.000000

Dot product of the two vectors is 70.000000
Process finished with exit code 0

```

4. 大量随机数冒泡排序

```c
//
// Created by ZidongZh on 2022/9/23.
//

#include<stdio.h>
#include<stdlib.h>

void BubbleSort(int array[], int length) {
    int i, j, temp;
    for (i = 0; i < length - 1; i++) {
        for (j = 0; j < length - i - 1; j++) {
            if (array[j] > array[j + 1]) {
                temp = array[j + 1];
                array[j + 1] = array[j];
                array[j] = temp;
            }
        }
    }
}

int main() {
    // Get the length of data
    int length;

    printf("Enter the size of data:");
    scanf("%d", &length);

    int array[length];

    for (int i = 0; i < length; i++) {
        array[i] = rand();
    }
    BubbleSort(array, length);

    for (int j = 0; j < length; j++) {
        printf("%d ", array[j]);
    }

    return 0;
}
```

结果

```
Enter the size of data:10
41 6334 11478 15724 18467 19169 24464 26500 26962 29358
```