

آزمایشگاه سیستم عامل

اعضای گروه: محمد نورسته – ابوالفضل نعمتی – علی حیدری – سپهر نوروزی

سوال 4 ام:

در این بخش ابتدا یک فایل تولید کرده و در این فایل کد مربوط به تمرین را مینویسیم حالا با استفاده از کامپایلر gcc کد را کامپایل میکنیم.

از آنجایی که در کد از کتابخانه math استفاده میکنیم نیاز داریم پس در ادامه دستور از این فلگ استفاده میکنیم:

`gcc -o outputfile inputfile.c -lm`

```
at ~ via C v9.4.0-gcc via v3.8.10
gcc -o Cfile Cfile.c -lm

at ~ via C v9.4.0-gcc via v3.8.10
ls
25linecode.txt Cfile Cfile.c config.omp.json fl micro omp.cache pyScript.py temp

at ~ via C v9.4.0-gcc via v3.8.10
./Cfile
5
1
2
3
4
5
Mean: 3.000000
Standard Deviation: 1.414214
```

کد:

```
#include <stdio.h>
```

```
#include <math.h>
```

```
double Mean(int numbers[], int size){
```

```
    int sum = 0;
```

```
    for (int i = 0; i < size; i++){
```

```
        sum += numbers[i];
```

```
{
```

```
    return (double)sum / size;
```

```
{
```

```
double Deviation(int numbers[], int size, double mean){
```

```

double squaredDifferencesSum = 0.0;

for (int i = 0; i < size; i++){
    double difference = (double)numbers[i] - mean;
    squaredDifferencesSum += difference * difference;
}

double meanOfSquaredDifferences = squaredDifferencesSum / size;
return sqrt(meanOfSquaredDifferences);
}

int main} ()
    int n;
    scanf("%d", &n);

    int numbers[n];

    for (int i = 0; i < n; i++){
        scanf("%d", &numbers[i]);
    }

    double mean = Mean(numbers, n);
    double standardDeviation = Deviation(numbers, n, mean);

    printf("Mean: %f\n", mean);
    printf("Standard Deviation: %f\n", standardDeviation);

    return 0;
}

```

