```
#include <iostream>
#include <cstdlib>
#include <omp.h>

using namespace std;
int total = 0;
int result[4] = {};
void fillBuffer(int* buf, int length,int num){
    srand(time(NULL));
    omp_set_num_threads(num);

    int i=0;
    #pragma omp parallel private(i)
    {
        for(i=0; i<length;i++){
        buf[i] = rand() % 100;
        cout<<"Element "<<i<<":  "<<buf[i]<<"  assigned to Thread 
"<<omp_get_thread_num()<<endl;
        }
    }
}

void displayBuffer(int* buf, int length,int num){

    #pragma omp parallel
    {

        int total=0;
        for(int i=0; i<length; i++){

            cout<<"\n"<<buf[i]<<"\t";
            printf("Num thread : %d\t",omp_get_thread_num());

                switch(omp_get_thread_num()){
                    case 0: result[0]+= buf[i];
                    case 1: result[1]+= buf[i];
                    case 2: result[2]+= buf[i];
                    case 3: result[3]+= buf[i];
                }
                total += buf[i];

            cout<<total;
            cout<<endl;
        }

    }




}
```

```
int main(){
	int *Dataset = NULL;
	int datalength = 0;
	int numthread = 0;

	cout<<"Type in number of elements of the dataset\t";
	cin>> datalength;

	cout<<"Type in number of thread\t";
	cin>> numthread;



	Dataset = (int *) malloc(datalength * sizeof(int));
	fillBuffer(Dataset,datalength,numthread);

	displayBuffer(Dataset,datalength,numthread);
	for(int i=0; i<numthread;i++){
	cout<<"\nThread "<<i<<" : My average is "<<result[i]/datalength<<"\n";
	}

	free(Dataset);

	return 0;
}
```