

# **Stock Market**

**Michał Drab**

# 1. Task topic

Stock Market

## 2. Project Analysis

First I created a schedule of smaller tasks to perform:

- Checking input parameters
- Loading input file
- Calculations
- Saving output files

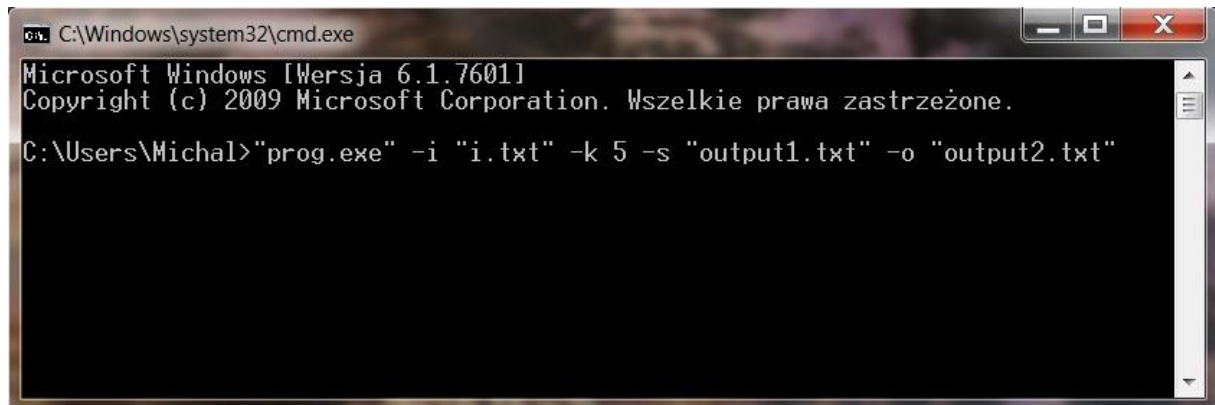
Data structures:

- typedef struct Struktura- This structure stores a pointer, which will indicate the memory allocated to our dynamic table, those are "float\*Element" and its size that is "int rozmiar".
- void PrepareStruktura(Struktura\* Element) - Prepares the table to perform.
- void Push(Struktura \*Element, float f) – This function adds new element, takes greater memory, increases the variable that "remembers" the size.
- void ZwolnijStruktura(Struktura \*Element)- cleans the memory allocated.
- void Argumenty(char\*\* argv,int argc, int\* input\_file, int\* step, int\* output\_srednia, int\* odchylenie) - assigns indexes from argv table to variables (parameters).
- void WczytajZPliku(char\* input\_filename,Struktura\* v)- reads a file, checks if file is not equal the end of file.
- int StrNalnt(char\* str) - converts numbers from string form to integers.
- float SredniaRuch(int step, int t,Struktura\* v)- this function calculates the moving average.
- float Odchylenie(float avg, int step, int t, Struktura\* v)- this function calculates the standard deviation.
- void StworzPlik(char\* out)- this function creates the output file.
- void BledneMozliwosci(char\* out) – this function extracts improper cases.
- void ZapiszDane(char\* str,float data)- records data to the file.
- int ZleDane(int input\_file, int step, int output\_srednia, int odchylenie, int argc)- it is responsible for three cases. The first checks if there is any parameter missing. The second works, when there is a file missing . The third is for a case, when there is a file name missing and instead of it there is given a next parameter.

### 3. Internal analysis

What does the program do? The program is made to calculate moving average and standard deviation then it prints results to output files. How should the input file look like?

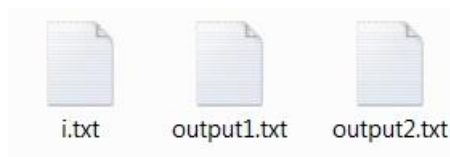
It should be a txt file in which is placed a list of numbers. First user has to run this program in command line – he has to give 5 parameters – name of program, name of input, step, output file name 1 and 2 (in which average and deviation would be saved).



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

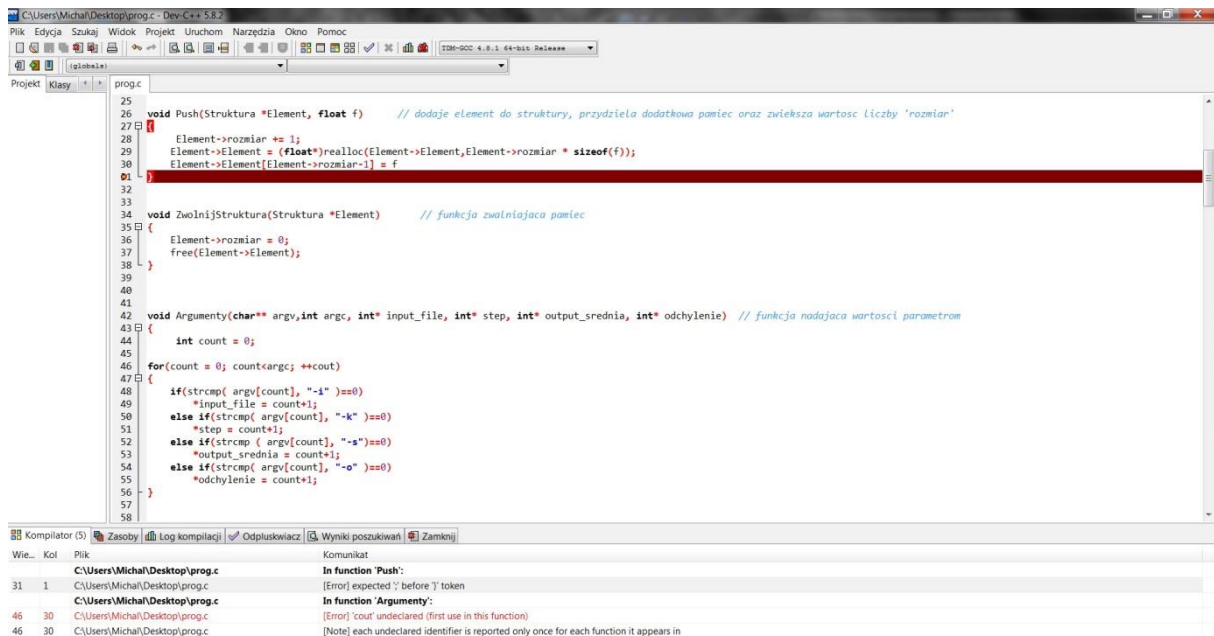
C:\Users\Michal>"prog.exe" -i "i.txt" -k 5 -s "output1.txt" -o "output2.txt"
```

Next program takes data from input, copies numbers from input to dynamic structure and calculates moving average and a trailing standard deviation. At the end program prints all results to output files. Output files have a .txt extension like an input file.



## 4. Testing

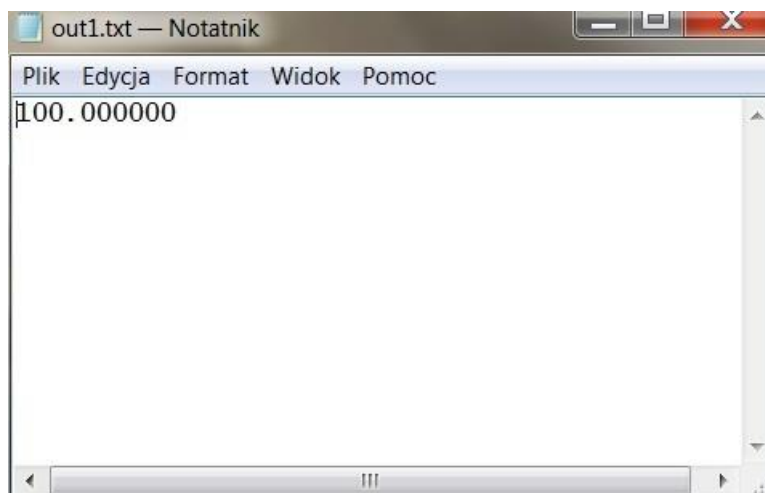
I ran my program a few times. There were small errors, for example sometimes I forgot to place the semicolon or I made mistakes in variable names.



The screenshot shows a C++ IDE with a file named `prog.c`. The code defines a linked list structure and functions to push, free, and parse arguments. The `main` function uses `argc` and `argv` to parse command-line arguments. The compiler output at the bottom shows two errors:

```
31 1 C:\Users\Michal\Desktop\prog.c In function 'Push':  
[Error] expected ';' before '}' token  
46 30 C:\Users\Michal\Desktop\prog.c In function 'Argumenty':  
[Error] 'cout' undeclared (first use in this function)  
46 30 C:\Users\Michal\Desktop\prog.c [Note] each undeclared identifier is reported only once for each function it appears in
```

For the first time my program calculated normal average. I had to search in the Internet what exactly moving average means and how to calculate it.



I also checked if I can run program if I give it wrong parameters. We know that program will crash so I have to write a function which will ask user to give parameters again.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Michał>"prog.exe" -i "i.txt" -o "output1.txt" -s "output2.txt"
Za malo argumentow
C:\Users\Michał>
```

Now the program prints “Za malo argumentow” which means “Too less arguments” and user has to give parameters again.

After this all my program works correctly.

```
i.txt — Notatnik
Plik  Edycja  Format  Widok  Pomoc
100
120
100
80
160
40
```

← this is the input file

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Michał>"prog.exe" -i "i.txt" -k 4 -s "average.txt" -o "deviation.txt"
Wszystko poszlo tak jak trzeba, otworz pliki by zobaczyc wyniki!
C:\Users\Michał>
```

```
deviation.txt — Notatnik
Plik  Edycja  Format  Widok  Pomoc
Blad - nie wystarczajaco danych do obliczenia.
Blad - nie wystarczajaco danych do obliczenia.
Blad - nie wystarczajaco danych do obliczenia.
14.142136
29.580399
43.301270

average.txt — Notatnik
Plik  Edycja  Format  Widok  Pomoc
Blad - nie wystarczajaco danych do obliczenia.
Blad - nie wystarczajaco danych do obliczenia.
Blad - nie wystarczajaco danych do obliczenia.
100.000000
115.000000
95.000000
```

There are an output files.