

[25.03.2020] Proof of concept – automat komórkowy

Andrzej Czechowski 307335 Bartosz Zakrzewski 307403

grupa 12, dziekańska grupa 3

Jest to trzeci dokument do projektu "Gra w Życie" (C)

1. Każdy zrzut ekranu powinien być czytelny.
2. Każdy zrzut ekranu powinien być podpisany, abym nie miał wątpliwości, co Państwo przedstawiają (np.: „Reakcja programu na próbę otwarcia pliku, który nie istnieje.”).
3. Wszystkie operacje, pokazujące działanie programu, należy wykonać na serwerze wydziałowym (np. **jimp.iem.pw.edu.pl**).
4. Każda instrukcja (prezentująca działanie programu), którą Państwo wpisali (i która jest widoczna na zrzucie ekranu), powinna być **powtórzona w formie tekstowej**, abym mógł ją skopiować i wykonać po swojej stronie. Proszę pamiętać, że jeżeli użyją Państwo jakiegoś pliku z danymi wejściowymi, to ten plik powinien być również w repozytorium, abym mógł powtórzyć operację.
5. Jeżeli czegoś nie można pokazać poprzez uruchomienie programu, to należy to (krótko) opisać.
6. Aktualną wersję kodu (do końca zajęć) w repozytorium należy oznaczyć tagiem: **PoC_C_1**.

Działanie na zajęciach:

Dane wejściowe:

```
czechoa1@jimp: ~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod
File Edit View Search Terminal Help
Username for 'https://projektor.ee.pw.edu.pl': czechoa1
Password for 'https://czechoa1@projektor.ee.pw.edu.pl':
Already up to date.
czechoa1@jimp:~/2020L_JIMP2_git_proj_gr12$ ls
'PROJEKT GRA W ZYCIE'
czechoa1@jimp:~/2020L_JIMP2_git_proj_gr12$ cd PROJEKT\ GRA\ W\ ZYCIE/
czechoa1@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE$ ls
dokumentacja kod
czechoa1@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE$ cd kod
czechoa1@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  GameFunctions.c  Main.o
2DArrayFunctions.h  GameFunctions.h  makefile
2DArrayFunctions.o  GameFunctions.o  out
dane                Main.c           'Poprawne wyniki 3 generacji'
czechoa1@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ make
cc -o out Main.c 2DArrayFunctions.c GameFunctions.c
czechoa1@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ cat dane
4 4
0
1 1 0 0
1 0 0 0
0 0 1 0
0 0 1 1
czechoa1@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```

Makefile - kompilacja:

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ git checkout
Your branch is up to date with 'origin/master'.
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  2DArrayFunctions.o  GameFunctions.c  GameFunctions.o  Main.o      out
2DArrayFunctions.h  dane                GameFunctions.h  Main.c           makefile    'Poprawne wyniki 3 generacji'
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ make
cc -o out Main.c 2DArrayFunctions.c GameFunctions.c
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```

Poprawne uruchomienia:

./out -input dane -output outputfolderSBS -outgen 10 -SBS

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -output outputfolderSBS -outgen 10 -SBS
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory outputfoldersBS
Czytany argument -outgen
Number of generations is 10
STEB BY STEP option
Result of printOutGen array:
1
1
1
1
1
1
1
1
1
1
1
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```

Powstanie 10 generacji w wybranym katalogu "outputfolderSBS"

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  2DArrayFunctions.o  GameFunctions.c  GameFunctions.o  Main.o      out
2DArrayFunctions.h  dane                GameFunctions.h  Main.c           makefile    outputfolderSBS  'Poprawne wyniki 3 generacji'
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ cd outputfoldersBS/
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod/outputfoldersBS$ ls
gen1  gen10  gen2  gen3  gen4  gen5  gen6  gen7  gen8  gen9
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod/outputfoldersBS$
```

./out - output OneGen -input dane -outgen 6

Powstała ostatnia generacja w folderze OneGEN

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -output OneGen -input dane -outgen 6
Czytany argument -output
Created directory OneGen
Czytany argument -input
Opened file: dane
Czytany argument -outgen
Number of generations is 6
No SBS option and no X Y Z option, only last generation will be saved
Result of printOutGen array:
0
0
0
0
0
1
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  2DArrayFunctions.o  GameFunctions.c  GameFunctions.o  Main.o      OneGen  outputfolderOneGen  'Poprawne wyniki 3 generacji'
2DArrayFunctions.h  dane                GameFunctions.h  Main.c           makefile    out      outputfolderSBS
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ cd OneGen
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod/OneGen$ ls
gen6
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod/OneGen$ cat gen6
4 4
6
0 0 0 0
0 1 1 0
0 0 0 0
0 0 0 0
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod/OneGen$
```

./out -outgen 5 3 5 -input dane -output outputfolderXYZ

Powstanie w folderze outputfolderXYZ tylko 3 i 5 generacji. (zmiana z założeniem z -gen n x,z,y na -gen n x y z)

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -outgen 5 3 5 -input dane -output outputfolderXYZ
Czytany argument -outgen
Number of generations is 5
    atol = 4
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory outputfolderXYZ
Result of printOutGen array:
0
0
1
0
1
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  2DArrayFunctions.o  GameFunctions.c  GameFunctions.o  Main.o  makefile  OneGen  outputfolderOneGen  outputfolderXYZ
2DArrayFunctions.h  dane                GameFunctions.h  Main.c           out      outputfolderSBS  'Poprawne wyniki 3 generacji'
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ cd outputfolderXYZ/
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod/outputfolderXYZ$ ls
gen3  gen5
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod/outputfolderXYZ$
```

Nie trzeba podawać argumentu wywołania -output.

./out -input dane -outgen 3 -SBS

Powstanie trzech generacji (gen1, gen2, gen3) w obecnym folderze (kod).

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -outgen 3 -SBS
Czytany argument -input
Opened file: dane
Czytany argument -outgen
Number of generations is 3
STEB BY STEP option
Result of printOutGen array:
1
1
1
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  dane                GameFunctions.o  gen3  makefile  outputfolderOneGen  'Poprawne wyniki 3 generacji'
2DArrayFunctions.h  GameFunctions.c    gen1             outputfolderSBS
2DArrayFunctions.o  GameFunctions.h    gen2             Main.o  out      outputfolderXYZ
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```

Obsługa błędów:

Nie istniejący plik wejściowy.

./out -input dane1 -output outputfolderSBS -outgen 10 -SBS

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane1 -output outputfolderSBS -outgen 10 -SBS
Czytany argument -input
File dane1 doesn't exist!
```

Brak wpisania wymaganych argumentów.

./out -output outputfolderSBS -outgen 10 -SBS

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -output outputfolderSBS -outgen 10 -SBS
Czytany argument -output
Created directory outputfolderSBS
Czytany argument -outgen
Number of generations is 10
STEB BY STEP option
No required argument -input!
tu będzie help
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```

./out -input dane -output outputfolderSBS

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -output outputfolderSBS
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory outputfolderSBS
No required argument -outgen!
tu będzie help
```

Brak wpisania pliku wejściowego.

`./out -input -output outputfolderSBS -outgen 10 -SBS`

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input -output outputfolderSBS -outgen 10 -SBS
Czytany argument -input
No input file!
```

Nie podanie ilości generacji.

`./out -input dane -output outputfolderSBS -outgen`

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -output outputfolderSBS -outgen
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory outputfolderSBS
Czytany argument -outgen
No outgen!
```

`./out -input dane -outgen -output outputfolderSBS`

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -outgen -output outputfolderSBS
Czytany argument -input
Opened file: dane
Czytany argument -outgen
No generation number!
```

Podanie ujemnej liczby generacji.

`./out -input dane -outgen -1 -output outputfolderSBS`

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -outgen -1 -output outputfolderSBS
Czytany argument -input
Opened file: dane
Czytany argument -outgen
invalid generation number!
```

Podanie niecałkowitej liczby generacji.

`./out -input dane -output outputfolderSBS -outgen 1.6 -SBS`

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -output outputfolderSBS -outgen 1.6 -SBS
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory outputfolderSBS
Czytany argument -outgen
invalid generation number!
```

Mimo błędnych argumentów wywołania nadal tworzy się folder o podanej nazwie i jest pusty (do poprawy).

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -output folderBadOutgen -outgen 1.6 -SBS
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory folderBadOutgen
Czytany argument -outgen
invalid generation number!
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  dane  GameFunctions.h  gen2  Main.o  out  outputfolderXYZ
2DArrayFunctions.h  folderBadOutgen  GameFunctions.o  gen3  Makefile  outputfolderOneGen  'Poprawne wyniki 3 generacji'
2DArrayFunctions.o  GameFunctions.c  gen1  Main.c  OneGen  outputfolderSBS
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```


Bez argumentów wywołania:

./out

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out
No call arguments!
tu bedzie help
```

Mimo błędnych argumentów wywołania nadal tworzy się folder o podanej nazwie i jest pusty (do poprawy) cd.

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -output folderBadSBS -outgen -1 -SBS
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory folderBadSBS
Czytany argument -outgen
invalid generation number!
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  dane                folderBadXYZ        GameFunctions.o    gen3    makefile  outputfolderOneGen  'Poprawne wyniki 3 generacji'
2DArrayFunctions.h  folderBadOutgen    GameFunctions.c     gen1               Main.c  OneGen    outputfolderSBS
2DArrayFunctions.o  folderBadSBS       GameFunctions.h     gen2               Main.o  out       outputfolderXYZ
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -output folderBadXYZ -outgen 5 3 -5
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory folderBadXYZ
Czytany argument -outgen
Number of generations is 5
Czytany argument -5
Invalid argument: -5!
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  dane                GameFunctions.c     gen1               Main.c    OneGen    outputfolderSBS
2DArrayFunctions.h  folderBadOutgen    GameFunctions.h     gen2               Main.o    out       outputfolderXYZ
2DArrayFunctions.o  folderBadXYZ       GameFunctions.o     gen3               makefile  outputfolderOneGen  'Poprawne wyniki 3 generacji'
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```

Uruchomienie z nieznanym argumentem wywołania.

./out -input dane -output TEST -outgen 10 -SBS -nieznanyargument.

(:::::zlanazwa i --zlanazwa to wcześniejszy test czy folder może mieć taką nazwę (mogą, nie ma to większego znaczenia dlatego brak pokazania uruchomienia))

```
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ./out -input dane -output TEST -outgen 10 -SBS -nieznanyargument
Czytany argument -input
Opened file: dane
Czytany argument -output
Created directory TEST
Czytany argument -outgen
Number of generations is 10
STEB BY STEP option
Czytany argument -nieznanyargument
Invalid argument: -nieznanyargument!
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$ ls
2DArrayFunctions.c  folderBadOutgen    GameFunctions.o    gen3               OneGen    outputfolderXYZ  ::::::zlanazwa
2DArrayFunctions.h  folderBadSBS       GameFunctions.h    Main.c             out       'Poprawne wyniki 3 generacji'
2DArrayFunctions.o  folderBadXYZ       gen1               Main.o             outputfolderOneGen  TEST
dane                GameFunctions.c    gen2               makefile           outputfolderSBS    --zlanazwa
zakrzewb@jimp:~/2020L_JIMP2_git_proj_gr12/PROJEKT GRA W ZYCIE/kod$
```

Do zrobienia:

- dodanie funkcji sprawdzające poprawność formatu pliku wejściowego (return -4)
- **programu tworzącego generacje w formacie PNG**
- napisanie funkcji help
- Opisanie kodu - uporządkowanie i dodanie poprawnych komentarzy opisujących działanie kodu
- podzielenie main na dwa pliki (Main.c i MainFunction.c)

(Na testArgumentow_i_PNG już zaczęliśmy prace nad tworzeniem plików PNG oraz tworzenie MainFunction.c)

Program powinien zwracać kody błędów zgodnie z założeniem w Specyfikacji Funkcjonalnej.