Witam uczniów klasy 3ti2.

Oceny: podobnie jak w kwietniu (ocena A1) również w maju zliczyłam ilość oddanych prac i wstawiłam z tych oddawanych w terminie ocenę A2, a następnie ocenę A uwzględniając wszystkie prace zdalne, również prace oddane po terminie. Oceny stawiane za niektóre zdalne też się liczą do średniej.

Ocena końcoworoczna uwzględnia oceny z **1 i 2 semestru** (jeżeli poprawiano klasówkę to zostawiałam ocenę lepszą).

Jeśli macie wątpliwości czy uwagi do propozycji ocen końcoworocznych, to indywidualnie prześlę wyliczenie. Niektóre oceny końcowe są wyraźnie lepsze od semestralnych, są też i gorsze – tu zadziałało zaangażowanie przy odsyłaniu prac.

Propozycje ocen wpisałam do dziennika, ostateczne oceny (po ewentualnych uwagach) wpiszę przed środą 17 czerwca.

Jak wspomniałam poprzednio, w materiałach na pozostałe tygodnie będą ćwiczenia i odnośniki do zadań maturalnych, które powinny rozwiązywać szczególnie osoby zainteresowane zdawaniem matury z informatyki. Oczywiście można pochwalić się rozwiązaniami czy zgłosić problem (mailem), nie będę już jednak sprawdzać czy oceniać zadań (oceny wystawione, przed Wami egzamin zawodowy).

I po raz kolejny - mam nadzieję, że rozumiecie działanie wysłanych do mnie programów i wszystkie stosowane w nich funkcje i konstrukcje. Cieszę się, że szukacie rozwiązań w Internecie i innych różnych źródłach. Warto jednak wykorzystywać pomysły najprostsze i przede wszystkim osobiście rozwiązywać problem. Na klasówkach i na egzaminie maturalnym nie ma Internetu.

Odpowiadałam na Wasze pytania i wątpliwości indywidualnie, jeżeli jednak coś mi umknęło, to proszę się przypomnieć.

Uwagi wysyłacie do mnie **dziennikiem** lub na adres: **aszczepaniak@zsp1piotrkow.edu.pl**.

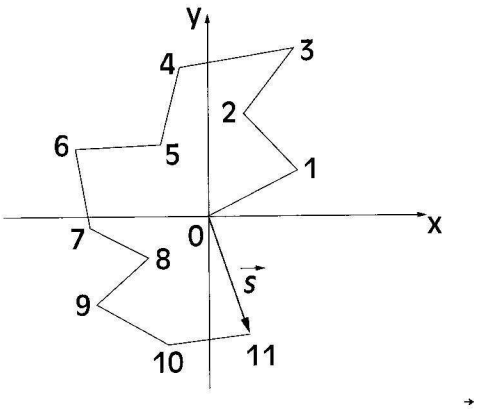
Kontynuujemy omawianie algorytmów klasycznych rozwiązywanych iteracyjnie (klasyczne2).

Omawiamy dwa przypadki zjawisk rozwiązywanych metodami Monte Carlo – obliczanie wartości liczby π (było w zeszłym tygodniu), obecnie zajmiemy się modelowaniem ruchów Browna. Problemy te (i sposób rozwiązywania) są zawarte w programie informatyki rozszerzonej i rzadko, ale zdarzają się w zadaniach maturalnych.

2. Modelowanie ruchów Browna.

W 1828r. szkocki botanik opublikował wyniki swoich badań. Zaobserwował, że różne cząsteczki (drobiny kurzu, sadzy, pyłki kwiatowe…) zawieszone w wodzie (lub innych substancjach) wykonują samoczynnie nieustanne i nieregularne ruchy podobne do błądzenia.

Spróbujemy za pomocą symulacji komputerowej zaprezentować przykładową trajektorię cząsteczki pływającej po powierzchni cieczy (czyli przestrzeń 2D) i obliczyć jej przemieszczenie w określonym czasie. Zakładamy, że startujemy z początku układu współrzędnych, że w każdym kroku cząsteczka przemieszcza się o wektor o stałej długości (np. o 1) w dowolnym (losowanym) kierunku. Wynikiem po n ruchach będzie odległość, na jaką przemieściła się cząsteczka od punktu 0,0 – długość wektora **s**.



- cząsteczka startuje z położenia 0, 0, współrzędne położenia po 1 ruchu będą x_1, y_1, po drugim x_2, y_2, … po n-tym x_n, y_n
- długość wektora policzymy wówczas jako: $|\vec{s}| = \sqrt{x_n^{\,2} + y_{n^2}}$
- kierunek „obrotu" cząstki przy każdym ruchu wylosujemy jako liczbę z przedziału <0, 2π>
- współrzędne położenia przy każdym ruchu obliczamy wg wzorów:

$$x_k = x_{k-1} + r \cdot \cos\varphi \qquad y_k = y_{k-1} + r \cdot \sin\varphi$$

gdzie: k = 1, 2, …, n
r – długość jednego kroku (przyjmujemy 1)
φ - kąt między kierunkiem ruchy cząsteczki a osią OX

Linie programu napisane na czerwono umożliwiają zapisanie wyników w pliku zewnętrznym i wykonanie np. wykresu trajektorii ruchu cząstki

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include <fstream>
using namespace std;
int main()
{
ofstream wyj("brown.xls");
const float pi=3.14159;
float x, y, s, fi;
long i, n;
cout << "ile chcesz ruchów? ";
cin>>n;
x=0; y=0;
wyj <<x << "\t" << y << endl;
srand(time(NULL));
for (i=0; i<n; i++)
{
  fi = (float) rand() /(RAND_MAX+1)*2*pi;
  x=x+cos(fi);
  y=y+sin(fi);
   wyj <<x << "\t" << y << endl;
}
s=sqrt(x*x+y*y);
cout << endl << "cząsteczka przemieściła się na odległość: "<< s;
wyj.close();
cin.ignore();  getchar();   return 0;
}
```

do obsługi plików

deklaracja i utworzenie pliku wyjściowego

deklaracja stałej pi

zapis początkowych współrzędnych do pliku

inicjacja generatora liczb

losowanie kierunku

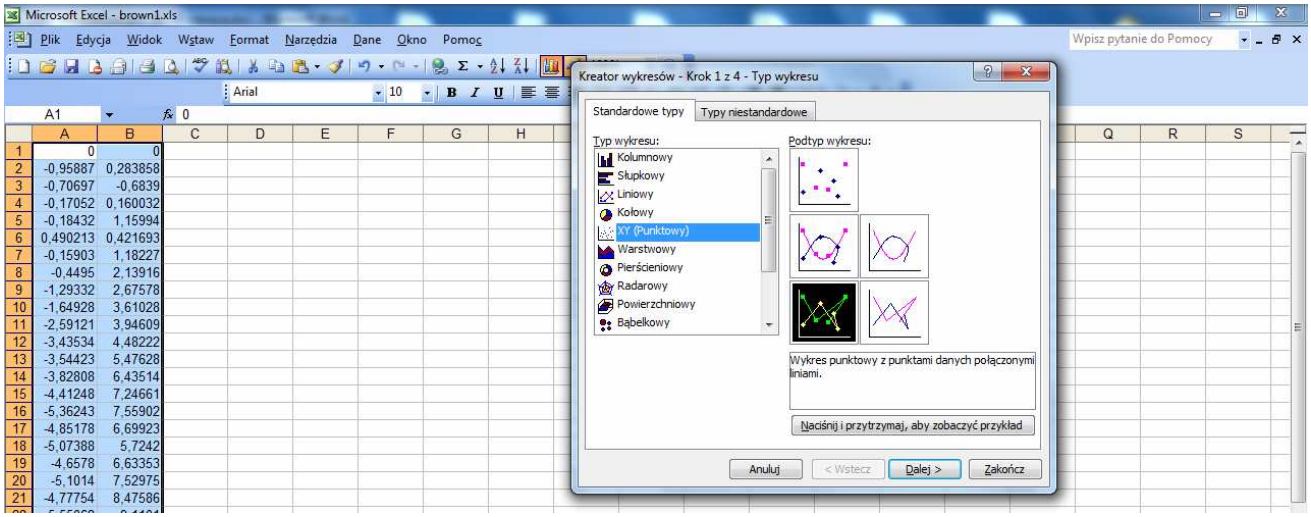
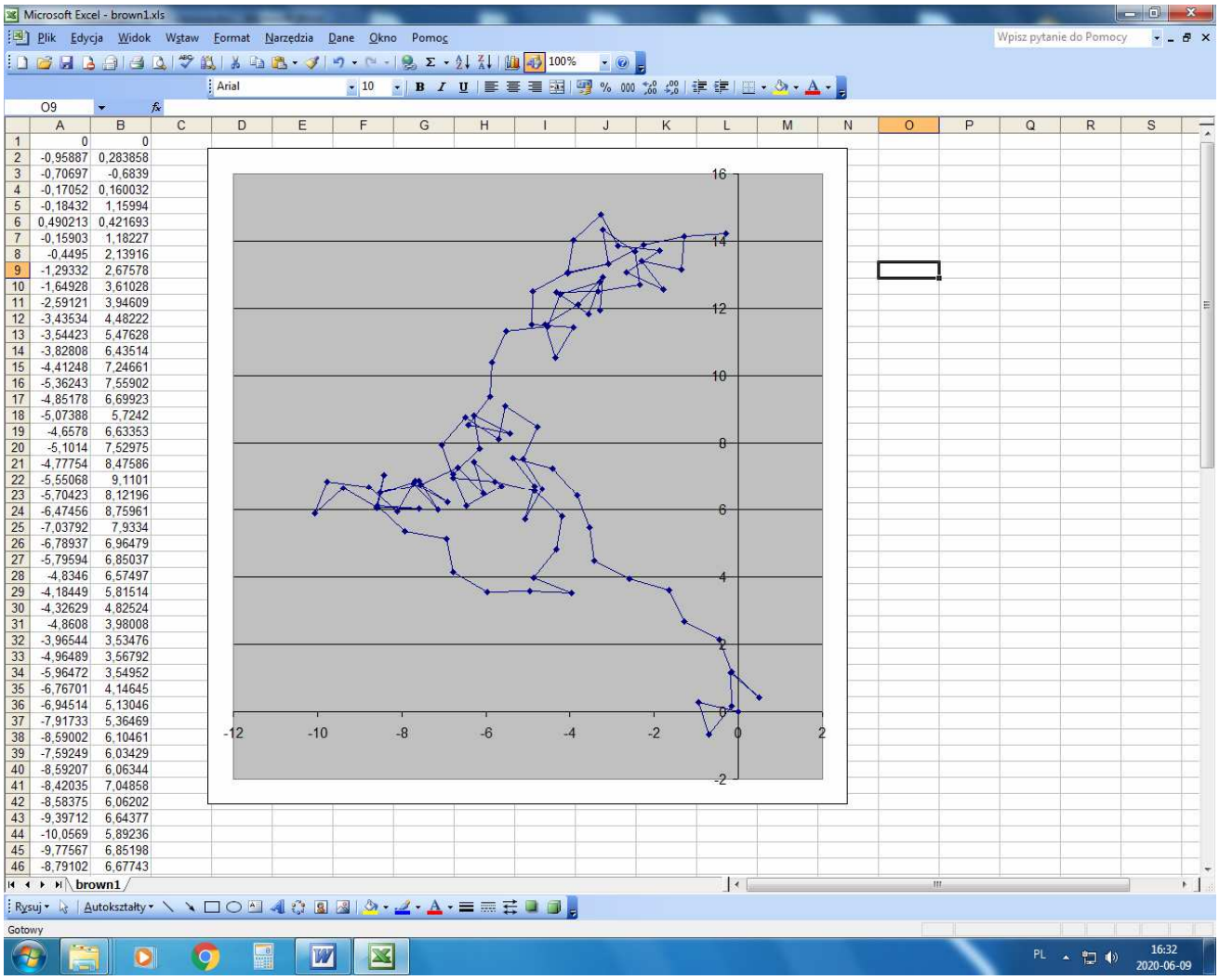
nowe współrzędne x i y

kolejne współrzędne zapisujemy do pliku

wynikowa długość przemieszczenia cząstki

zamykamy plik

Program oblicza i wypisuje wynikową długość wektora s.
Po uruchomieniu programu otrzymamy plik brown.xls, w którym będzie n par liczb.
Otwieramy plik w Excelu, zaznaczamy obie kolumny, wybieramy w menu Edycja/Zamień i zamieniamy kropki w liczbie rzeczywistej zapisanej w programie na przecinek (jeśli tego nie zrobimy, Excel widzi liczbę jak tekst).
Zaznaczamy obie kolumny i na pasku wybieramy Kreator wykresów/typ wykresu: x,y(punktowy)



W poniższym przykładzie n=100, punkty oczywiście przemieszczają się inaczej po każdym uruchomieniu programu.