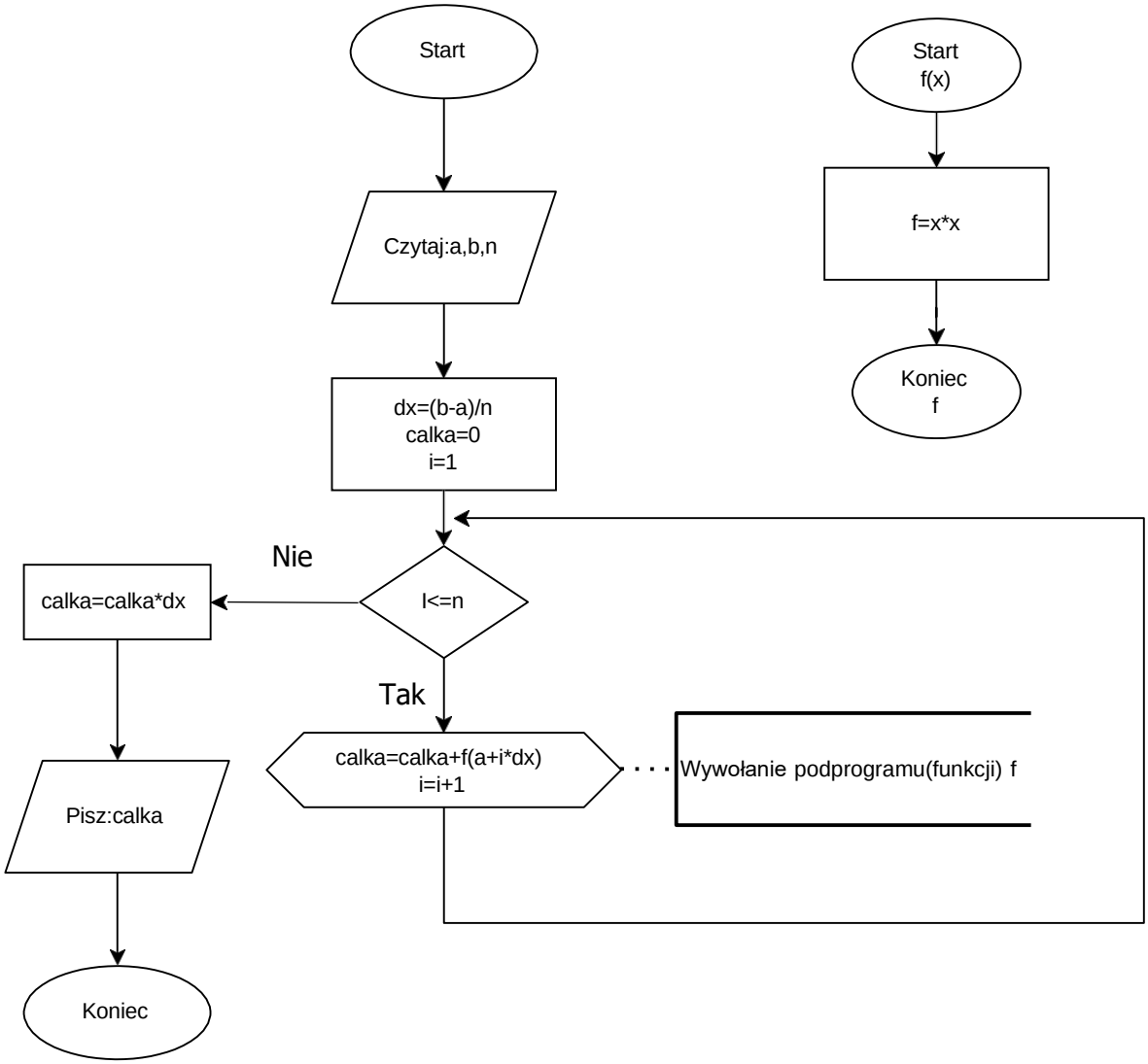
Przedmiot	Algorytmy i struktury danych				
Nazwa zadania	Ćwiczenie zaliczeniowe nr 1				
Nazwisko imię	Bagieński Kamil				
Nr albumu	155623	Grupa	D1	Rodzaj studiów (D,Z,W)	W
Rok akademicki	2022/2023			Semestr	I
Data wykonania	02.12.2022				

Zadanie 1

Schemat blokowy i lista korków do obliczania całki oznaczonej <a,b> z n ilością przedziałów

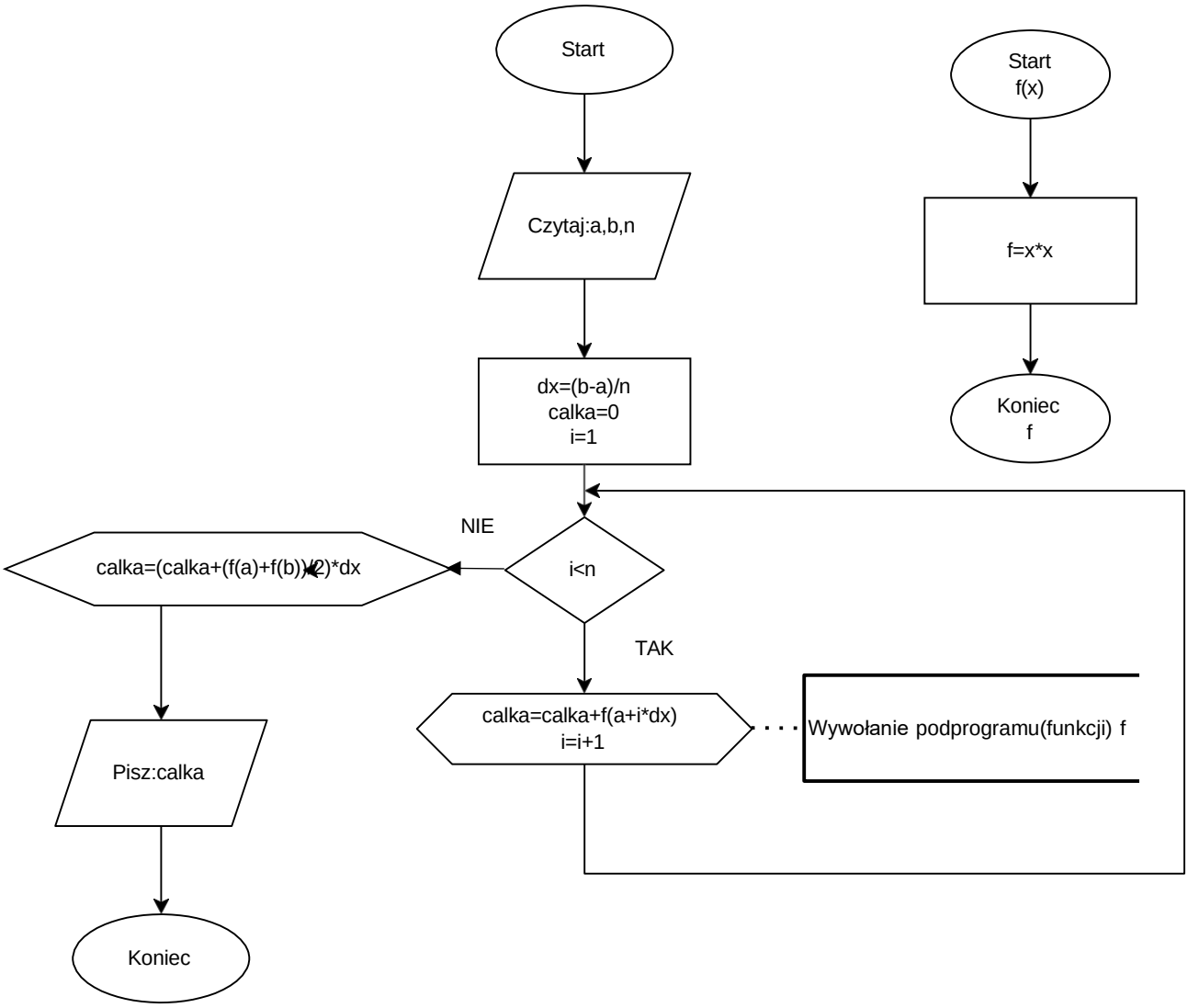
Schemat blokowy metoda prostokątów



Lista kroków metoda prostokątów

1. Krok 1- wprowadź początek przedziału a, koniec przedziału b oraz ilość podziałów n
2. Krok 2- oblicz dx=(b-a)/n, ustaw calka=0, ustaw i=1
3. Krok 3- Sprawdź czy i<=n jeśli nie to idź do Krok 5, jeśli tak idź do Krok 4
4. Krok 4- Wywołaj funkcje f i oblicz calka=calka+f(a+i*dx), ustaw i=i+1, idź do Krok 3
5. Krok 5- oblicz calka=calka*dx
6. Krok 6- wyświetl całka

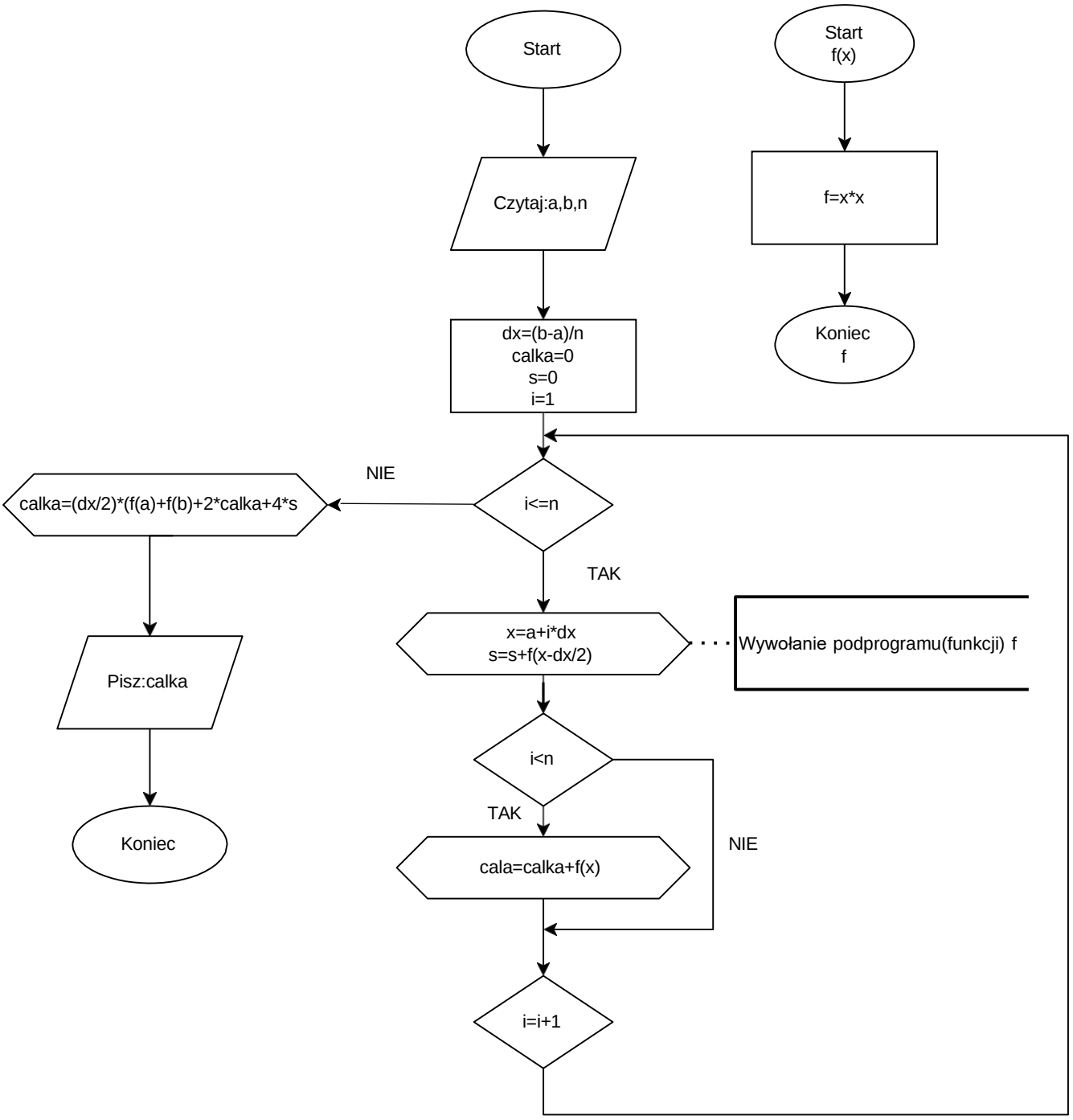
Schemat blokowy metoda trapezów



Lista kroków metoda trapezów

1. Krok 1- wprowadź początek przedziału a, koniec przedziału b oraz ilość podziałów n
2. Krok 2- oblicz dx=(b-a)/n, ustaw calka=0, ustaw i=1
3. Krok 3- Sprawdź czy i<n jeśli nie to idź do Krok 5, jeśli tak idź do Krok 4
4. Krok 4- Wywołaj funkcje f i oblicz (calka+(f(a)+f(b))/2)*dx, ustaw i=i+1, idź do Krok 3
5. Krok 5- oblicz calka=(calka+(f(a)+f(b))/2)*dx wywolujac funkcje f
6. Krok 6- wyświetl całka

Schemat blokowy metoda Simpsona



Lista kroków metoda Simpsona

1. Krok 1- wprowadź początek przedziału a, koniec przedziału b oraz ilość podziałów n
2. Krok 2- oblicz dx=(b-a)/n, ustaw calka=0, ustaw i=1, ustaw=0
3. Krok 3- Sprawdź czy i<=n jeśli nie to idź do Krok 8, jeśli tak idź do Krok 4
4. Krok 4- Wywołaj funkcje f i oblicz x=a+i*dx oraz s=s+f(x-dx/2)
5. Krok 5- sprawdz czy i<n jeśli nie to idz do kroku 7, jeśli tak to idz do kroku 6
6. Krok 6- oblicz cala=calka+f(x)
7. Krok 7- oblicz i=i+1, idz do kroku 3
8. Krok 8- oblicz calka=calka*dx
9. Krok 9- wyświetl całka

Zadanie 2
Tabelaryczne zestawienie wyników dla całki oznaczonej w przedziale <1,2> z N=5,10,100,1000

Liczba przedziałów:	5	10	100	1000
Metoda prostokątów	2.640000	2.485000	2.348350	2.334834
Metoda trapezów	2.340000	2.335000	2.333350	2.333334
Metoda Simpsona	2.333333	2.333333	2.333333	2.333333

Metoda Simpsona dawała najlepsze wyniki przy najmniejszej ilości iteracji(ilości podziałów N). Zarówno metoda prostokątów i trapezów nie była tak dokładna. Wynika z faktu, że przy liczeniu metodą prostokątów czy trapezów mamy kąty które wychodzą albo zostawiają miejsce poza polem funkcji. Zanim dojdzie do skorygowania błędu wynikającego z braku dokładności w wyliczeniach potrzeba dużo więcej iteracji. Metoda Simpsona liczy pole powierzchni funkcji niezależnie od jej kształtu, gdyż dostosowuje się do niej.

Aby sprawdzić poprawność wyników korzystałem z kalkulatorów całek na stronie
https://www.wolframalpha.com
oraz aplikacji
PhotoMath na Androida

Zadanie 3 i 4

Wartości które zakończyły się osiągnieciem założonej dokładności:

Wartości początkowe	Pierwiastek	Ilość iteracji/rekurencji	Wynik x^x	Różnica a-x^x
a=4, p=10, Eps=0.001, MaxI=5	2	5	4	0
a=9, p=10, Eps=0.001, MaxI=5	3	4	9	0
a=16, p=10, Eps=0.001, MaxI=5	4	4	16	0
a=25, p=10, Eps=0.001, MaxI=5	5	4	25	0
a=36, p=10, Eps=0.001, MaxI=5	6	3	36	0

Wartości które zakończyły się przez maksymalną ilość iteracji/rekurencji:

Wartości początkowe	Pierwiastek	Ilość iteracji/rekurencji	Wynik x^x	Różnica a-x^x
a=4, p=100, Eps=0.001, MaxI=5	3,54	5	12,53	-9
a=9, p=100, Eps=0.001, MaxI=5	4,03	5	16,24	-7
a=16, p=100, Eps=0.001, MaxI=5	4,67	5	21,8	-6
a=25, p=100, Eps=0.001, MaxI=5	5,42	5	29,42	-4
a=36, p=100, Eps=0.001, MaxI=5	6,26	5	39,22	-3

Przy założonej ilości iteracji oraz rekurencji nie można dowolnie zmieniać parametru pierwszego przybliżenia oraz dokładności. Program wypisałby takie same wyniki gdyby mógł przekroczyć MaxI. Jednakże dzięki wprowadzeniu maksymalnej ilości iteracji/rekurencji można zabezpieczyć program przed przeciążeniem stosu. Bo gdy użytkownik końcowy poda

jakieś bardzo duże liczby, program mógłby liczyć je w nieskończoność aż do momentu przeciążenia stosu i zajęcia całej pamięci obliczeniowej. Granica iteracji/rekurencji pozwala na ustalenie bezpiecznej ilości działań aplikacji. Dzięki czemu użytkownik końcowy nie przeciąży stosu wpisując bardzo skrajne wartości.