



Błędy obliczeń numerycznych – zadania maturalne

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Błędy obliczeń numerycznych – zadania maturalne

Źródło: Nick Hillier, domena publiczna.

W e-materiale [Błędy obliczeń numerycznych](#) poznaliśmy najważniejsze informacje dotyczące błędów obliczeń.

W tym e-materiale zajmiemy się przykładowym zadaniem maturalnym dotyczącym błędów w obliczeniach komputerowych.

Implementacje algorytmu w poszczególnych językach programowania zostały omówione w e-materiałach:

- [Błędy obliczeń numerycznych w języku C++](#),
- [Błędy obliczeń numerycznych w języku Java](#),
- [Błędy obliczeń numerycznych w języku Python](#).

Twoje cele

- Rozwiążesz przykładowe zadania maturalne z zakresu błędów obliczeń numerycznych.
- Przeanalizujesz zapisane za pomocą pseudokodu rozwiązania zadań maturalnych sprawdzających wiedzę dotyczącą błędów obliczeń numerycznych.
- Napiszesz program związany z błędami w obliczeniach numerycznych (w wybranym języku).

Przeczytaj

Zadanie 1

Grupa fizyków ATOM regularnie przeprowadza pomiary wybranych zjawisk przyrodniczych oraz analizę wyników związanych z nimi eksperymentów. Używane przez nich narzędzia pomiarowe cechują się bardzo wysoką dokładnością. Część obliczeń komputerowych przeprowadzanych na zmierzonych wartościach nie musi być jednak aż tak precyzyjna. Fizycy, chcąc skrócić czas operacji numerycznych, dokonują licznych zaokrągleń i [przybliżeń](#). Niezbędny jest system, który będzie pomagał w ich wyznaczaniu oraz w weryfikacji błędów. Naukowcy mają kilka koncepcji, jednak potrzebują pomocy w ich realizacji.

Zadanie 1.1

Pomóż fizykom wykonać kilka szybkich obliczeń na podstawie pomiarów zebranych podczas badania. Oblicz ich zaokrąglenie do dwóch cyfr po przecinku oraz wartość błędu bezwzględnego, wynikającego z przeprowadzonej operacji.

Oto wzór na błąd bezwzględny:

$$\Delta x = |x_0 - x|$$

gdzie:

- Δx – błąd bezwzględny,
- x_0 – wartość przed zaokrągleniem,
- x – zaokrąglona wartość.

Polecenie 1

Uzupełnij tabelę:

Wynik pomiaru	Zaokrąglenie	Wartość błędu bezwzględnego
0,8643		
0,672		
0,07911		
1,2189		

Rozwiązanie

Jeśli chcesz przypomnieć sobie, w jaki sposób zaokrąglamy liczby, informacje na ten temat znajdziesz w e-materiale [Przybliżenia i zaokrąglania](#).

Prawidłowe obliczenia, które ma wykonać uczeń, to:

- zaokrąglenie liczby będącej wynikiem pomiaru (zgodnie z zasadami znanymi z lekcji matematyki);
- obliczenie wartości bezwzględnej różnicy uzyskanego wyniku i jego zaokrąglenia.

Dla wyniku 0,8643 wykonujemy następujące operacje:

Zaokrąglamy liczbę do dwóch miejsc po przecinku. Zgodnie z zasadami zaokrąglania przyjrzyjmy się pierwszej odrzuconej cyfrze. Na trzecim miejscu po przecinku znajduje się cyfra 4. Zatem przepisujemy pozostawione cyfry bez zmian:

$$0,8643 \approx 0,86$$

Następnie od wyniku pomiaru odejmujemy przybliżenie. Pamiętajmy, że różnica ma być wartością bezwzględną:

$$|0,8643 - 0,86| = 0,0043$$

Analogicznie postępujemy w przypadku drugiego wyniku, jakim jest 0,672:

$$0,672 \approx 0,67$$

$$|0,672 - 0,67| = 0,002$$

W przypadku trzeciego wyniku - czyli 0,07911 - sytuacja wygląda nieco inaczej. Pierwsza odrzucona cyfra to 9, zatem powinniśmy zwiększyć ostatnią pozostawioną cyfrę o 1:

$$0,07911 \approx 0,08$$

Obliczamy różnicę. W tym przypadku widać, dlaczego istotne jest to, by była to wartość bezwzględna; gdyby było inaczej, błąd byłby wartością ujemną.

$$|0,07911 - 0,08| = 0,00089$$

Analogicznie postępujemy z czwartym pomiarem, czyli liczbą 1,2189:

$$1,2189 \approx 1,22$$

$$|1,2189 - 1,22| = 0,0011$$

Uzupełniona tabela wygląda następująco:

Wynik pomiaru	Zaokrąglenie	Wartość błędu bezwzględnego
0,8643	0,86	0,0043
0,672	0,67	0,002
0,07911	0,08	0,00089
1,2189	1,22	0,0011

Zadanie 1.2

Wytyczne fizyków: W pliku pp . txt zapisano 12 liczb rzeczywistych, każdą w oddzielnym wierszu. Pierwsze 10 liczb to liczby zmiennoprzecinkowe, oznaczające przybliżenia oryginalnego pomiaru. Następną liczbą to wartość oryginalnego pomiaru. Ostatnią liczbą jest wartość krytyczna.

Plik pp . txt do pobrania:

Plik o rozmiarze 94.00 B w języku polskim

Napisz program, który określi, czy dla każdego przybliżenia wartość jego błędu względnego przekracza wartość krytyczną. Program powinien w kolejnych wierszach wypisywać komunikat TAK, jeżeli wartość błędu względnego dla danego przybliżenia przekracza wartość krytyczną, lub NIE w przeciwnym wypadku.

Do oceny oddajesz:

- plik wynik.txt zawierający odpowiedź do zadania (10 wierszy, w każdym wierszu komunikat TAK, jeżeli wartość błędu względnego przekracza wartość krytyczną, lub NIE w przeciwnym wypadku),
- plik(i) z komputerową realizacją zadania (kodem programu).

Przykład 1

Dla przykładowego pliku:

```
1 0.088
2 0.083
3 0.0751
4 0.0853153
5 0.086
```

poprawna odpowiedź wynosi:

```
1 NIE
2 NIE
3 TAK
```

Przedstaw rozwiązanie zadania w postaci programu napisanego w wybranym języku (C++, Java lub Python). Zadbaj o prawidłowe wczytanie danych z pliku tekstowego do programu. Odpowiedź znajdziesz pod omówieniem rozwiązania.

Rozwiązanie

Pierwszym krokiem algorytmu jest wprowadzenie danych wejściowych, zapis przybliżeń z pliku do tablicy oraz pobranie błędu krytycznego z klawiatury. Po wykonaniu tych operacji należy dla każdego przybliżonego pomiaru obliczyć wartość błędu bezwzględnego według następującego wzoru:

$$bz = |op - pp|$$

Następnie obliczamy wartość błędu względnego:

$$bw = \frac{bz}{|op|}$$

Aby wykonać operację dla każdego z przybliżonych pomiarów, wykorzystamy pętlę dla. Następnie, za pomocą instrukcji warunkowej jeżeli, umieścimy w niej dodatkowo porównanie otrzymanych wyników z błędem krytycznym. W zależności od rezultatu wypisywany będzie stosowny komunikat.

Oto przykładowe rozwiązanie:

```

1 pp ← wczytaj 10 pierwszych wierszy z pliku pp.txt
2 op ← wczytaj 11 wiersz z pliku pp.txt
3 bk ← wczytaj 12 wiersz z pliku pp.txt
4
5 dla i = 0, 1, 2, ..., n - 1 wykonuj:
6
7     bz[i] ← wartoscBezwzgledna(op - pp[i])
8     bw[i] ← bz[i]/wartoscBezwzgledna(op)
9
10    jeżeli bw[i] > bk wykonaj:
11        wpisz "TAK" do pliku wynik.txt
12    w przeciwnym razie wykonaj:
13        wpisz "NIE" do pliku wynik.txt

```

Odpowiedź

Odpowiedź do zadania znajduje się w pliku wynik.txt:

Plik o rozmiarze 48.00 B w języku polskim

Słownik

błąd bezwzględny

wyraża bezwzględną różnicę pomiędzy wartością zmierzoną a dokładną; obliczamy go zgodnie ze wzorem:

$$\Delta x = |x - x_0|$$

gdzie x to dokładna wartość, x_0 oznacza zmierzoną wartość, a Δx – błąd bezwzględny

błąd względny

informuje, o ile procent różni się wartość zmierzona od dokładnej; obliczamy go zgodnie ze wzorem:

$$\delta = \frac{|x - x_0|}{x} \cdot 100\% = \frac{\Delta x}{x} \cdot 100\%$$

gdzie x to dokładna wartość, x_0 oznacza zmierzoną wartość, Δx – błąd bezwzględny, a za pomocą δ określamy błąd względny

przybliżenie

operacja zmniejszająca dokładność, z jaką prezentujemy liczbę; odbywa się poprzez pozbycie się z wartości ustalonej liczby miejsc po przecinku, przy założeniu, że ostatnią cyfrę przybliżenia (ostatnie zachowane miejsce po przecinku) zwiększamy o 1, gdy następuje po niej cyfra większa lub równa 5

Prezentacja multimedialna

Zadanie 1.3

Wytyczne fizyków: W ramach współpracy z innymi placówkami badawczymi zebrane zostały wyniki tego samego eksperymentu, pochodzące z 10 różnych laboratoriów (znajdziesz je w pliku tekstowym `Pomiary.txt`). Wyniki są liczbami rzeczywistymi zaokrąglonymi do czterech miejsc po przecinku.

Niestety, sprzęt wykorzystywany do przeprowadzania pomiarów przez odrębne grupy naukowców, był różny. Na skutek tego otrzymane rezultaty obarczone zostały błędami danych wejściowych o różnych wartościach (błędy wynikają z określonej dokładności przyrządów). Reprezentują one maksymalną różnicę pomiędzy uzyskanym wynikiem a rzeczywistą wartością mierzonego parametru. W oznaczeniach matematycznych błąd ten poprzedza znak $\pm$ (plus minus).

W każdym wierszu pliku tekstowego `Pomiary.txt` znajduje się wynik pojedynczego pomiaru oraz odpowiadający mu błąd danych wejściowych. Dane w jednej linii oddzielone są pojedynczym znakiem spacji.

Przykład reprezentacji pomiarów w pliku tekstowym:

```
1 0.0214 0.0005
2 0.0219 0.0002
3 0.0226 0.0004
```

Przedstawiony zapis oznacza, że pomiar osiągnięty przez fizyków (linia nr 1) wynosił 0,0214, natomiast jego maksymalny błąd wynosi 0,0005. W drugiej linii podany jest pomiar innej grupy naukowców – ich pomiar wyniósł 0,0219, a błąd pomiaru: 0,0002. W następnych liniach pliku mamy wartości i błędy wyznaczone przez kolejne zespoły.

Pomiar znajdujący się w pierwszym wierszu pliku został uzyskany przez twoją grupę badawczą.

Napisz program, który wyznaczy różnicę pomiędzy każdym z pozostałych wyników, a rezultatem uzyskanym przez twoją grupę badawczą oraz wyliczy błędy, jakimi obarczone są otrzymane różnice. Program powinien również obliczyć, jaką maksymalną i minimalną rzeczywistą wartość mógł mieć badany w eksperymentach parametr.

Rezultaty pracy zapisz w pliku `Wyniki.txt`. Wyniki kolejnych obliczonych różnic oraz maksymalnych błędów (oddzielone znakiem spacji) wypisz w osobnych wierszach.

W ostatnich dwóch liniach powinny się znaleźć odpowiednio: maksymalna i minimalna rzeczywista wartość parametru.

Przykładowe obliczanie różnicy, wynikowych błędów oraz potencjalnych minimalnych i maksymalnych rzeczywistych wartości pomiarów:

Dany jest pomiar wykonany przez nasz zespół – pomiar a :

$$a = 0,0214 \pm 0,0005$$

Dany jest również pomiar wykonany przez inny zespół b :

$$b = 0,0219 \pm 0,0002$$

Obliczamy wartość wynikowego błędu poprzez dodanie do siebie błędów pomiaru a oraz b :

$$0,0005 + 0,0002 = 0,0007$$

Różnicę pomiarów (c) obliczymy, odejmując od wartości pomiaru a wartość pomiaru b ; różnica powinna uwzględniać obliczoną wartość wynikowego błędu:

$$c = a - b = -0,0005 \pm 0,0007$$

$$0,03 \leq c \leq 0,09$$

Aby obliczyć maksymalną wartość, jaką mógł mieć w rzeczywistości parametr a , dodajemy do wartości pomiaru wartość błędu:

$$a_{\max} = 0,0214 + 0,0005 = 0,0219$$

Maksymalna wartość, jaką mógł mieć w rzeczywistości parametr b :

$$b_{\max} = 0,0219 + 0,0002 = 0,0221$$

Aby obliczyć minimalną wartość, jaką mógł mieć w rzeczywistości parametr a , odejmujemy od wartości pomiaru wartość błędu:

$$a_{\min} = 0,0214 - 0,0005 = 0,0209$$

Minimalna wartość, jaką mógł mieć w rzeczywistości parametr b :

$$b_{\min} = 0,0219 - 0,0002 = 0,0217$$

Przykład 1

Dla przykładowego pliku:

```
1 0.0214 0.0005
2 0.0219 0.0002
3 0.0226 0.0004
```

wynik powinien wyglądać następująco:

```
1 -0.0005 0.0007
2 -0.0012 0.0009
3 0.023
4 0.0209
```

W pierwszej linii mamy zapisane wartości $-0,0005$ i $0,0007$. Wynik $-0,0005$ oznacza różnicę pomiędzy $0,0214$ (wynikiem uzyskanym przez twój zespół badawczy) oraz $0,0219$ (wynikiem uzyskanym przez naukowców z drugiego zespołu, czyli wynikiem umieszczonym w drugiej linii). Wynik $0,0007$ natomiast jest wynikiem sumy błędów tych obliczeń ($0,0005 + 0,0002$).

W kolejnej linii proces został powtórzony – tym razem dla porównania wyników uzyskanych przez twój zespół badawczy oraz przez naukowców, których wyniki zapisano w trzeciej linii.

W trzeciej linii uzyskanych wyników mamy maksymalną wartość, jaką mógł osiągnąć pomiar. W tym przypadku jest to $0,023$, co zostało uzyskane z wykonania operacji $0,0026 + 0,0004$, czyli sumy wartości pomiaru oraz błędu otrzymanego przez naukowców z trzeciego zespołu.

W czwartej linii pliku mamy natomiast minimalną wartość, jaką mógł osiągnąć ten pomiar. Wynosi ona $0,0209$, co jest wynikiem operacji $0,0214 - 0,0005$, czyli różnicy wartości pomiaru oraz błędu otrzymanego przez twój zespół badawczy.

Plik `Pomiary.txt`:

Plik o rozmiarze 148.00 B w języku polskim

Do oceny oddajesz:

- plik Wyniki.txt zawierający odpowiedź do zadania,
- plik(i) z komputerową realizacją zadania (kodem źródłowym).

Polecenie 1

Przedstaw rozwiązanie zadania w postaci programu, napisanego w wybranym języku (C++, Java lub Python). Zadbaj o prawidłowe wczytanie danych z pliku tekstowego do programu. Odpowiedź do zadania znajdziesz w osobnym pliku, umieszczonym pod omówieniem pseudokodu.

Rozwiązanie

Polecenie 2

Zapoznaj się z rozwiązaniem zadania przedstawionym w formie pseudokodu.



1

Waga należy do najpopularniejszych narzędzi służących do pomiaru. Dokładność współczesnych wag jest większa niż dokładność wagi zaprezentowanej na ilustracji. Zastanów się, z czego mogły wynikać błędy pomiaru wykonane za pomocą przedstawionej wagi.

Źródło: Poussin jean, Wikimedia Commons, commons.wikimedia.org, dostęp: 08.12.2022 r., CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pl4cIEZWK>

Zaczynamy od wczytania wartości pomiarów i błędów z pliku `Pomiary.txt` i zapisania ich w postaci dwóch tablic: `pomiary` oraz `bład`, zawierających odpowiednio zmierzone wartości i odpowiadające im błędy danych wejściowych. Tworzymy pętlę, w której będziemy wczytywać liczby do tablic:

```
1 dla i = 0, 1, ..., liczba  
  wierszy w Pomiary.txt - 1:
```

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PI4cIEZWK>

Następnie w każdej iteracji wczytujemy pierwszą oraz drugą liczbę z wiersza do tablicy `pomiary`. Po wykonaniu tej czynności przechodzimy do kolejnego wiersza:

```
1 dla i = 0, 1, ..., liczba  
  wierszy w Pomiary.txt - 1:  
2   pomiary <- wczytaj  
   pierwszą liczbę z obecnego  
   wiersza  
3   blad <- wczytaj drugą  
   liczbę z obecnego wiersza  
4   przejdź do następnego  
   wiersza
```

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PI4cIEZWK>

Deklarujemy jeszcze dwie zmienne: `minWartosc` oraz `maxWartosc` - i przypisujemy im początkowe wartości:

`pomiary[0]` - `blad[0]` oraz `pomiary[0] + blad[0]`. Będą one potrzebne, ponieważ po obliczeniu wszystkich potencjalnych maksymalnych i minimalnych wartości będziemy szukać ekstremów (maksimum i minimum) globalnych.

```
1 dla i = 0, 1, ..., liczba
  wierszy w Pomiary.txt - 1:
2   pomiary <- wczytaj
  pierwszą liczbę z obecnego
  wiersza
3   blad <- wczytaj drugą
  liczbę z obecnego wiersza
4   przejdź do następnego
  wiersza
5
6 minWartosc <- pomiary[0] -
  blad[0]
7 maxWartosc <- pomiary[0] +
  blad[0]
```

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PI4cIEZWK>

Po zainicjowaniu potrzebnych tablic i zadeklarowaniu niezbędnych zmiennych przechodzimy do obliczania różnic między poszczególnymi pomiarami oraz błędów wynikowych. Tworzymy pętlę dla (for), w której wykonamy zaplanowane operacje.

Podczas każdej iteracji (będzie ich w sumie 9 – czyli tyle, ile wyników odejmowania) do wierszy pierwszej kolumny tablicy `rozPomiary` zapisywać będziemy odpowiednie wartości. Wykorzystamy ją do przechowywania informacji na temat różnic między pierwszym pomiarem (wykonanym przez nasz zespół) a

pozostałymi pomiarami oraz wartości uzyskanego błędu.

Zapisujemy zatem kolejne różnice pomiędzy pierwszym pomiarem (wykonanym przez grupę fizyków ATOM) a pozostałymi pomiarami. W drugiej kolumnie wspomnianej tablicy umieszczamy wyliczone wyniki błędów.

```
1 dla i = 0, 1, ..., liczba
  wierszy w Pomiary.txt - 1:
2   pomiary ← wczytaj
  pierwszą liczbę z obecnego
  wiersza
3   blad ← wczytaj drugą
  liczbę z obecnego wiersza
4   przejdź do następnego
  wiersza
5
6 minWartosc ← pomiary[0] -
  blad[0]
7 maxWartosc ← pomiary[0] +
  blad[0]
8
9 dla i = 0, ..., 8 wykonuj:
10   rozPomiary[i][0] ←
  pomiary[0] - pomiar[i+1]
11   rozPomiary[i][1] ←
  blad[0] + blad[i+1]
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pl4cIEZWK>

5

Po obliczeniu różnic oraz błędów zajmujemy się odnalezieniem wartości minimalnych i maksymalnych, którymi mogły się charakteryzować mierzone parametry. Tworzymy kolejną pętlę dla, która zapewni powtórzenie wskazanych operacji 10 razy (tyle, ile wynosi całkowita liczba pomiarów).

Pierwszym działaniem umieszczonym w pętli jest wyznaczenie maksymalnej potencjalnej wartości parametru dla każdego z pomiarów i zapisanie rezultatów do pierwszej kolumny tablicy `rzWartosc`. W tym celu dodajemy do zmierzonej wartości błąd danych wejściowych.

```
1 dla i = 0, 1, ..., liczba
  wierszy w Pomiary.txt - 1:
2   pomiary <- wczytaj
  pierwszą liczbę z obecnego
  wiersza
3   blad <- wczytaj drugą
  liczbę z obecnego wiersza
4   przejdź do następnego
  wiersza
5
6 minWartosc <- pomiary[0] -
  blad[0]
7 maxWartosc <- pomiary[0] +
  blad[0]
8
9 dla i = 0, ..., 8 wykonuj:
10   rozPomiary[i][0] <-
  pomiary[0] - pomiar[i+1]
11   rozPomiary[i][1] <-
  blad[0] + blad[i+1]
12
13 dla j = 0, ..., 9 wykonuj:
14   rzWartosc[j][0] <-
  pomiary[j] + blad[j]
```

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PI4cIEZWK>

Analogicznie przeprowadzamy drugą operację zawartą w pętli. Wyznaczamy minimalne wartości poprzez odjęcie od pomiaru błędu danych wejściowych. Wyniki umieszczamy w drugiej kolumnie tablicy `rzWartosc`.

```

1 dla i = 0, 1, ..., liczba
  wierszy w Pomiary.txt - 1:
2   pomiary <- wczytaj
  pierwszą liczbę z obecnego
  wiersza
3   blad <- wczytaj drugą
  liczbę z obecnego wiersza
4   przejdź do następnego
  wiersza
5
6 minWartosc <- pomiary[0] -
  blad[0]
7 maxWartosc <- pomiary[0] +
  blad[0]
8
9 dla i = 0, ..., 8 wykonuj:
10   rozPomiary[i][0] <-
    pomiary[0] - pomiar[i+1]
11   rozPomiary[i][1] <-
    blad[0] + blad[i+1]
12
13 dla j = 0, ..., 9 wykonuj:
14   rzWartosc[j][0] <-
    pomiary[j] + blad[j]
15   rzWartosc[j][1] <-
    pomiary[j] - blad[j]

```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pl4clEZWK>

7

Równoległe z wyznaczaniem maksymalnych i minimalnych wartości dla każdego z pomiarów poszukujemy także maksimum i minimum globalnego. W tym celu w utworzonej pętli dla umieszczamy dwa warunki jeżeli, które będą każdorazowo sprawdzać, czy otrzymany wynik maksymalny jest większy od aktualnej wartości zmiennej `maxWartosc` oraz czy otrzymany wynik minimalny jest mniejszy od `minWartosc`. Jeżeli warunek jest spełniony,

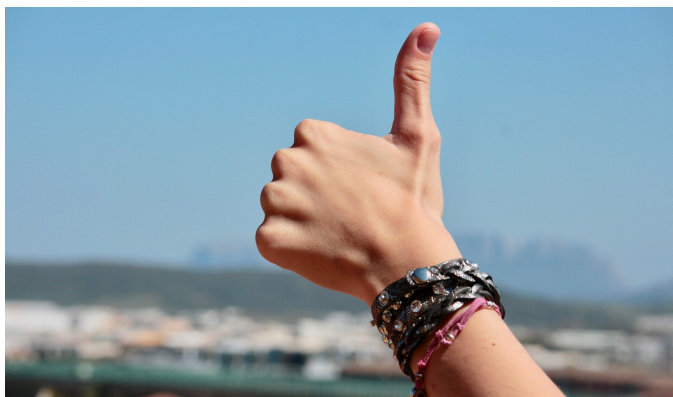
wówczas wartość zmiennej z warunku zostaje zastąpiona porównywaną wartością z tablicy rzWartosc.

```
1 dla i = 0, 1, ..., liczba
  wierszy w Pomiary.txt - 1:
2   pomiary <- wczytaj
  pierwszą liczbę z obecnego
  wiersza
3   blad <- wczytaj drugą
  liczbę z obecnego wiersza
4   przejdź do następnego
  wiersza
5
6 minWartosc <- pomiary[0] -
  blad[0]
7 maxWartosc <- pomiary[0] +
  blad[0]
8
9 dla i = 0, ..., 8 wykonuj:
10   rozPomiary[i][0] <-
    pomiary[0] - pomiar[i+1]
11   rozPomiary[i][1] <-
    blad[0] + blad[i+1]
12
13 dla j = 0, ..., 9 wykonuj:
14   rzWartosc[j][0] <-
    pomiary[j] + blad[j]
15   rzWartosc[j][1] <-
    pomiary[j] - blad[j]
16   jeżeli rzWartosc[j][0]
    > maxWartosc wykonaj:
17     maxWartosc <-
    rzWartosc[j][0]
18   jeżeli rzWartosc[j][1]
    < minWartosc wykonaj:
19     minWartosc <-
    rzWartosc[j][1]
```

Na koniec zapisujemy otrzymane wyniki do pliku tekstowego Wyniki.txt.

```
1 dla i = 0, 1, ..., liczba
  wierszy w Pomiary.txt - 1:
2   pomiary <- wczytaj
  pierwszą liczbę z obecnego
  wiersza
3   blad <- wczytaj drugą
  liczbę z obecnego wiersza
4   przejdź do następnego
  wiersza
5
6 minWartosc <- pomiary[0] -
  blad[0]
7 maxWartosc <- pomiary[0] +
  blad[0]
8
9 dla i = 0, ..., 8 wykonuj:
10   rozPomiary[i][0] <-
    pomiary[0] - pomiar[i+1]
11   rozPomiary[i][1] <-
    blad[0] + blad[i+1]
12
13 dla j = 0, ..., 9 wykonuj:
14   rzWartosc[j][0] <-
    pomiary[j] + blad[j]
15   rzWartosc[j][1] <-
    pomiary[j] - blad[j]
16   jeżeli rzWartosc[j][0]
    > maxWartosc wykonaj:
17     maxWartosc <-
    rzWartosc[j][0]
18   jeżeli rzWartosc[j][1]
    < minWartosc wykonaj:
19     minWartosc <-
    rzWartosc[j][1]
20
21 zapisz tablicę rozPomiary
  do pliku Wyniki.txt
22 zapisz maxWartosc do pliku
  Wyniki.txt
```

23 zapisz minWartosc do pliku
Wyniki.txt



9

Pierwsze narzędzia pomiaru były zainspirowane...
ludzkim ciałem. Pokłósiem tego są niektóre
jednostki, takie jak stopa. Czasem jednak wciąż
mówimy, że coś ma szerokość palca – zakładano,
że cal to szerokość kciuka.

Źródło: Pixabay.com, dostęp: 08.12.2022 r., CC 0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PI4cIEZWK>

Zapisałiśmy za pomocą pseudokodu program,
który może posłużyć do porównywania
prostych pomiarów. Zauważmy, że
przeprowadzając operacje na dwóch
wartościach obarczonych błędami danych
wejściowych, otrzymaliśmy wynikową wartość
obarczoną jeszcze większym błędem. Algorytmy,
w których dochodzi do sumowania się błędów
numerycznych, nazywamy algorytmami
niestabilnymi.

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Ważne!

W programie wykorzystaliśmy algorytm jednoczesnego wyznaczania minimalnej oraz
maksymalnej wartości. Opisany on został w e-materiale [Instrukcja warunkowa –
ćwiczenia](#).

Odpowiedź

Odpowiedź do zadania znajduje się w pliku Wyniki.txt:

Plik o rozmiarze 150.00 B w języku polskim

Sprawdź się

Zadanie 1.4

Wytyczne fizyków: W pliku tekstowym `Pomiary4.txt` zawarte zostały kolejne wyniki pomiarów dla jednego z eksperymentów (każdy w osobnym wierszu). Wyniki to 10 liczb rzeczywistych.

Plik o rozmiarze 202.00 B w języku polskim

Jesteś członkiem zespołu naukowców. Twoim zadaniem jest zaokrąglenie uzyskanych wyników do czterech miejsc po przecinku, zarówno z nadmiarem, jak i niedomiarem, a następnie ustalenie i zapisanie tej wartości, która jest obarczona mniejszym błędem względnym. W sytuacji, gdy błąd względny w obu przybliżeniach będzie identyczny, wybierz przybliżenie z nadmiarem. Przykładowo: dla 0,66665 należy wybrać 0,6667.

Przykład:

Założmy, że weryfikujemy następujący pomiar:

$$x = 0,0328532$$

Przybliżenie z nadmiarem:

$$x_{nad} = 0,0329$$

Przybliżenie z niedomiarem:

$$x_{nie} = 0,0328$$

Błąd względny dla przybliżenia z nadmiarem:

$$\delta_{nad} = \frac{|0,0328532 - 0,0329|}{0,0328532} \approx 0,0014$$

Błąd względny dla przybliżenia z niedomiarem:

$$\delta_{nie} = \frac{|0,0328532 - 0,0328|}{0,0328532} \approx 0,0016$$

Do pliku z wynikami zostanie zapisane przybliżenie z nadmiarem.

Przykład 1

Dla danych:

```
1 0.5971687771
2 0.3214921073
3 0.6773841485
4 0.0391127383
5 0.9181704912
```

poprawny wynik to:

```
1 0.5972
2 0.3215
3 0.6774
4 0.0391
5 0.9182
```

Napisz program, który dla podanych wartości wyznaczy i wypisze przybliżenie do czterech miejsc po przecinku, obarczone mniejszym błędem względnym. W sytuacji, gdy błąd względny w obu przybliżeniach (z nadmiarem i niedomiarem) będzie identyczny, program powinien wypisać przybliżenie z nadmiarem.

Do oceny oddajesz:

- plik `Wyniki4.txt` zawierający odpowiedź (dla każdego pomiaru wpisane przybliżenie, które jest dokładniejsze do czterech miejsc po przecinku, każde w osobnym wierszu),
- plik(i) z komputerową realizacją zadania (kodem źródłowym programu).

Praca domowa

Zadbaj o prawidłowe wczytanie danych z pliku tekstowego do programu. Przedstaw rozwiązanie zadania w postaci programu napisanego w wybranym języku (C++, Java lub Python).

Pokaż ćwiczenia:   

Przetestuj swój program dla następujących wyników pomiarów:

```
1 0.5971687771
2 0.3214921073
```

3	0.6773841485
4	0.0391127383
5	0.9181704912
6	0.3824046134
7	0.2733563614
8	0.2533776472
9	0.0624205043
10	0.4022714246
11	0.8589004220
12	0.2700638959
13	0.3056901354
14	0.5423353265
15	0.7115026918
16	0.8916964420
17	0.7021393621
18	0.9953887976
19	0.9248190635
20	0.2522954455
21	0.4428451020
22	0.8175704076
23	0.9546093979
24	0.7985598648
25	0.2145403166



C++

Twoje zadania

1. Program ma wypisać elementy tablicy wyniki, zawierającej przybliżone wartości przeprowadzonych pomiarów.





JAVA

Twoje zadania

1. Program ma wypisać elementy tablicy wyniki, zawierającej przybliżone wartości przeprowadzonych pomiarów.





PYTHON

Twoje zadania

1. Program ma wypisać elementy tablicy wyniki, zawierającej przybliżone wartości przeprowadzonych pomiarów.

Odpowiedź

Odpowiedź do zadania znajduje się w pliku:

Plik o rozmiarze 78.00 B w języku polskim

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Błędy obliczeń numerycznych – zadania maturalne

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

7) wyjaśnia, jakie może być źródło błędów pojawiających się w obliczeniach komputerowych: błąd zaokrąglenia, błąd przybliżenia;

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Rozwiążesz przykładowe zadania maturalne z zakresu błędów obliczeń numerycznych.
- Przeanalizujesz zapisane za pomocą pseudokodu rozwiązania zadań maturalnych sprawdzających wiedzę dotyczącą błędów obliczeń numerycznych.
- Napiszesz program związany z błędami w obliczeniach numerycznych (w wybranym języku).

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Błędy obliczeń numerycznych – zadania maturalne”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj” w kontekście programowania.

Faza wstępna:

1. Prowadzący wyświetla na tablicy interaktywnej zawartość sekcji „Wprowadzenie” i omawia cele do osiągnięcia w trakcie lekcji.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytania dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu i programowania, np. Chętni uczniowie udzielają na nie odpowiedzi.

Faza realizacyjna:

1. **Praca z tekstem.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj”. Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.
2. Uczniowie w parach zapoznają się z zadaniem nr 1.1 i jego rozwiązaniem. Następnie w parach zapoznają się z zadaniem nr 1.2 i jego rozwiązaniem. Nauczyciel odpowiada na ewentualne pytania.
3. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna”. Uczniowie wspólnie analizują rozwiązanie zadania 1.3.
4. **Ćwiczenia umiejętności.** Uczniowie wykonują zadanie 1.4 z sekcji „Sprawdź się”. Ćwiczenie to uczniowie realizują w jednym z dostępnych języków programowania.

Faza podsumowująca:

1. Nauczyciel wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji podsumowuje przebieg zajęć, a także wskazuje mocne i słabe strony pracy uczniów.
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie wykonują ćwiczenie z sekcji „Prezentacja multimedialna”. Ich zadaniem jest implementacja przedstawionego w tej sekcji zadania lokalnie na swoim komputerze w języku, w którym programują.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).

- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Prezentacja multimedialna”, „Sprawdź się” jako materiał do lekcji powtórkowej.