

Podstawy programowania

Zadania ćwiczeniowe

Semestr I

dr inż. Sławomir Jeżewski

Wprowadzenie

Pisanie programów jest umiejętnością praktyczną. Jazda na rowerze też jest umiejętnością praktyczną.

Ostatnie osiągnięcia neuro-nauk wykazują, że w ludzkim mózgu istnieją osobne ośrodki do obsługi pamięci biernej i osobne ośrodki do obsługi umiejętności praktycznych. Rozdział ośrodków ma bardzo istotne konsekwencje. Inaczej trzeba się uczyć umiejętności teoretycznych (trenujemy wtedy pamięć bierną) a inaczej się uczymy umiejętności praktycznych.

Nikt nie uczy się jazdy na rowerze z podręczników. Trzeba samodzielnie próbować, upadać , jeszcze raz próbować aby osiągnąć sukces. W ten sposób trenowane są ośrodki umiejętności praktycznych zlokalizowane w mózdzku i korze nowej. Podobnie jest z programowaniem czytanie podręczników, które trenuje pamięć bierną (zapamiętywanie) nie jest wystarczające do tego by programować. Aby nauczyć się programować trzeba jak z jazdą na rowerze , próbować , upadać, frustrować się , jeszcze raz próbować.

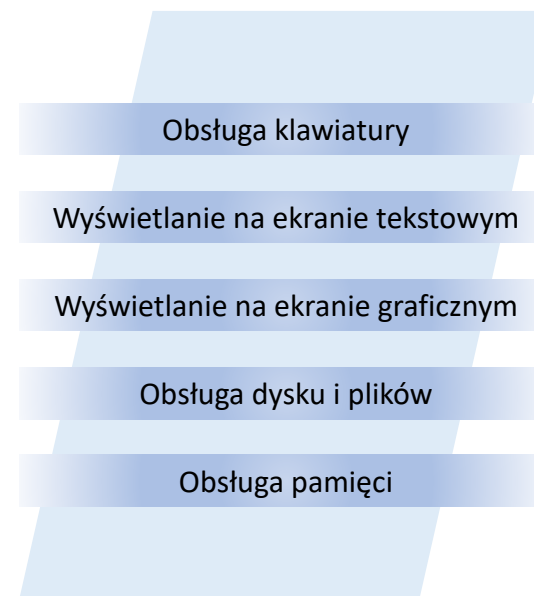
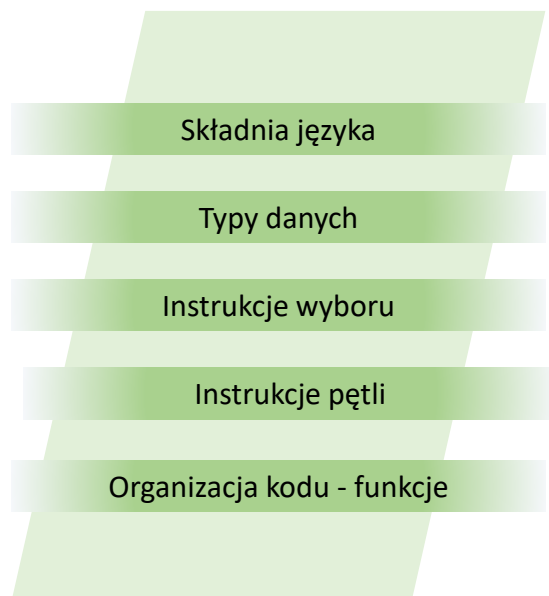
Wielu z nas swoją pierwszą samodzielną drogę na rowerze przebyło z górki, bo z górki jest łatwiej.

Niniejszy plik zawiera zestaw zadań programistycznych , które są taką górką. Umieszczone tu zadania mają ułatwić wejście w świat programowania. Ich trudność wzrasta wraz z kolejnym zdaniem. Wykonując je po kolei odkrywamy kolejne obszary sztuki programowania.



Zawartość kursu

Aby umieć programować na poziomie praktycznym czyli umieć zakodować dowolny algorytm, oraz umieć zaprojektować algorytm do rozwiązania przedstawionego problemu musimy poznać: konstrukcje języka programowania oraz poznać funkcje z bibliotek standardowych. Zadania w niniejszym kursie są tak zaprojektowane że stopniowo zapoznajemy się z obydwoma obszarami. Każde z zadań może należeć do jednej lub więcej kategorii



Praca z programami w języku Python

Język Python należy do języków interpretowanych.

Cykl prac z programem wygląda następująco:

- EDYCJA - Programista tworzy tekst programu w postaci kodu źródłowego (zrozumiałego dla człowieka)
- URUCHOMIENIE – programista uruchamia program, interpreter wykonuje program zgodnie z zapisem w kodzie źródłowym

Praca z systemem Python w Linuxie

Interpreter pythona można uruchomić w dwu trybach w trybie interaktywnym oraz w trybie wykonania skryptu

Aby uruchomić pythona w trybie wykonania skryptu należy podać jego nazwę jako element polecenia

```
python plik.py [ arg1, arg2 ,arg3 ]
```

gdzie:

python/python3 - nazwa programu który jest interpreterem języka python.

plik.py to nazwa pliku zawierających skrypt programu

arg1, ... argn - argumenty przekazywane do wykonywanego programu

[] nawiasy oznaczają że jakiś element wywołania jest opcjonalny

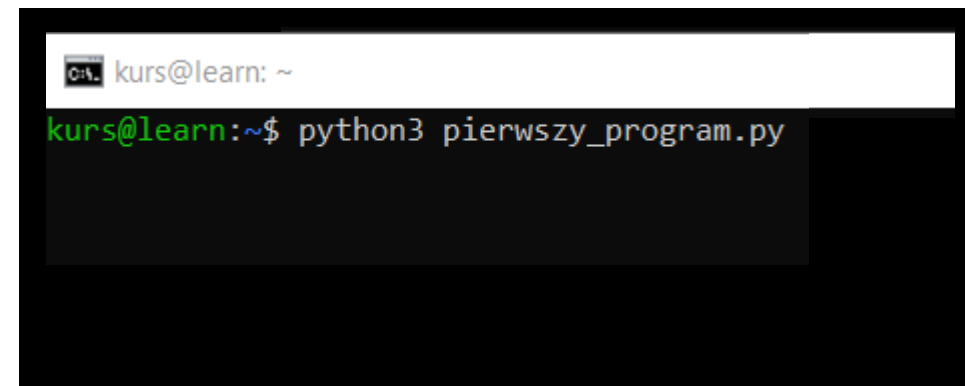
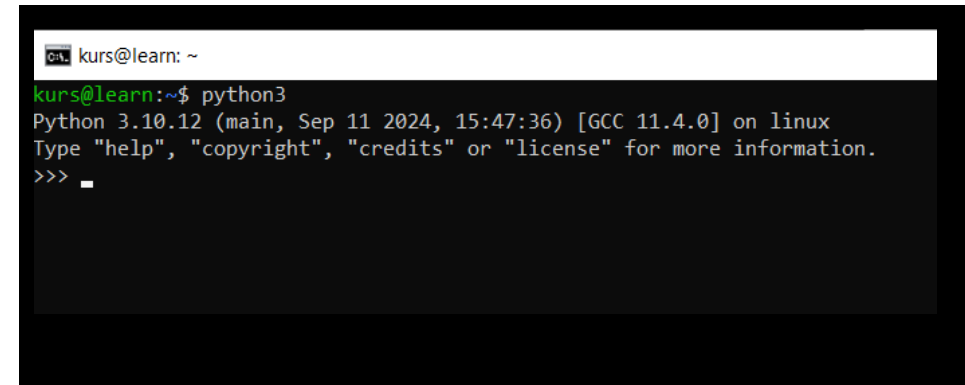
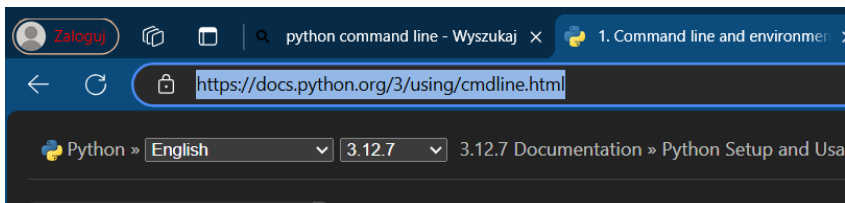
Aby uruchomić interpreter w trybie interaktywnym wystarczy napisać

```
python
```

Więcej informacji na temat przełączników interpretera pythona można znaleźć na stronie [docs.python.org](https://docs.python.org/3/using/cmdline.html)

[Command line and environment](https://docs.python.org/3/using/cmdline.html). Kolejne wersje pythona różnią się znacznie

w zakresie dopuszczalnych konstrukcji językowych , warto zwrócić uwagę , której wersji pythona dotyczy plik pomocy



Programy wykonywalne w Linux'ie

Program wykonywalny w systemie Linux to taki plik który ma atrybut wykonywalności

Atrybuty możesz sprawdzić poleceniem

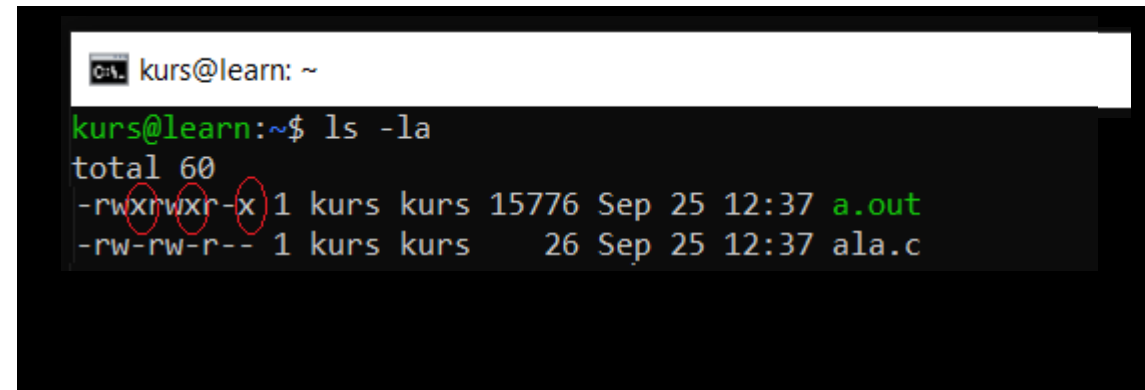
```
ls -la
```

służy ono do wyświetlania plików w danym katalogu

Atrybut wykonywalności zaznaczany jest literką x i jest osobno ustawiany dla właściciela pliku, grupy i wszystkich pozostałych użytkowników

Skrypt języka python można uczynić programem wykonywalnym , wystarczy nadać mu atrybut wykonalności poleceniem [chmod](#)

```
chmod 644 my_script.py
```



```
kurs@learn: ~  
kurs@learn:~$ ls -la  
total 60  
-rwxrwxr-x 1 kurs kurs 15776 Sep 25 12:37 a.out  
-rw-rw-r-- 1 kurs kurs 26 Sep 25 12:37 ala.c
```

Uruchamianie programów w systemie Linux

System linux ma standardowe miejsce przechowywania programów i poleceń

```
/bin
```

```
/usr/bin
```

```
/usr/local/bin
```

Jeśli uruchamiamy program i w poleceniu nie podamy całej ścieżki do programu , czyli

```
ls -la
```

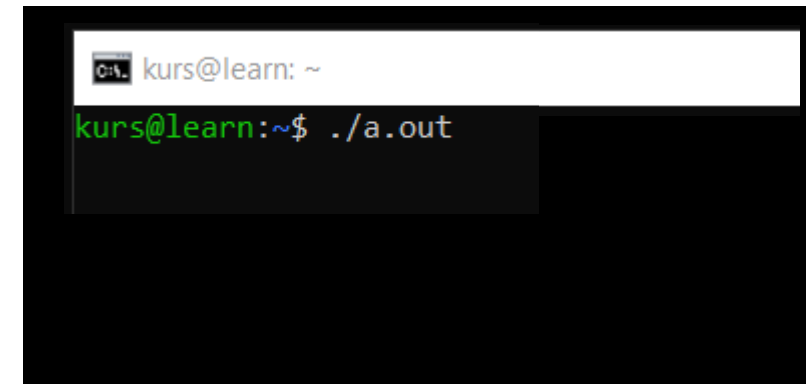
system linux będzie szukał polecenia `ls` w katalogach standardowych , jeśli go tam nie znajdzie wyświetli błąd

Aby uruchomić jakikolwiek program , który nie jest w nie jest w katalogach wskazywanych zmienną otoczenia path należy podać pełną ścieżkę do tego programu np.

```
/home/kurs/my_script.py
```

można skrócić ten zapis używając `./` na oznaczenie aktualnego katalogu

```
./my_script.py
```



```
kurs@learn: ~  
kurs@learn:~$ ./a.out
```

Zadanie

Polecenie

Zapoznaj się z plikami cw1.py ... cw5.py.

Cw1.py

```
# tak się oznacza komentarz w języku Python
# komentarz jest pomijany podczas kompilacji
# stanowi swoistą dokumentację i wyjaśnienia
# programisty dla programisty
```

```
"""
To jest przykład 'wielolinijkowego' komentarza
w Pythonie.
"""
```

```
# w komentarzach będą zawarte informacje i
wyjaśnienia dotyczące kodu programu
```

Cw2.py

```
# poniżej zdefiniowano najmniejszy , jaki można
# napisać program w języku python
# jest on pusty i nie zawiera żadnych poleceń
```

Cw3.py

```
#!/bin/python3
```

```
#Komentarz powyżej ma szczególne znaczenie w systemie Linux. Mówi on #
powłoce (sh) jakiego interpretera użyć do wykonania niniejszego pliku
```

```
Jeśli ten komentarz jest w pliku , oraz plik ma ustawiony atrybut
wykonywalności , to można ten plik uruchomić tak jak każdy inny program w
linuxie
./Cw3.py
```

Cw4.py

```
"""
aby skorzystać z zewnętrznego modułu i jego funkcji trzeba go zaimportować.
Poniżej zaimportowano moduł sys. Są tam funkcje do obsługi argumentów podanych z linii komend
"""
```

```
import sys
```

```
# funkcja print wyświetla na ekranie zawartość wszystkich argumentów podanych z linii komend
print(sys.argv)
```

```
# wyświetlenie na ekranie drugiego argumentu
print(sys.argv[1])
```

Cw5.py

```
import sys
```

```
# program który piszemy może być programem głównym albo być fragmentem modułu - zbioru
zmiennych funkcji użytkowych ( tak jak moduł sys ).
```

```
# Ta instrukcja warunkowa pozwoli dowiedzieć się czym jest nasz skrypt
if __name__ == '__main__':
    # Ten fragment kodu zostanie wykonany jeśli nasz moduł nie jest wykonywany jako fragment
    wyrażenia import
    # innymi słowami , ten fragment kodu jest wykonywany tylko wtedy gdy nasz skrypt jest
    skryptem głównym
    print(sys.argv, len(sys.argv))
    ...
```

```
// <-- tu nie zapisujemy instrukcji programu,
```

A.1 Uruchamianie programu

Dane wejściowe

-

Polecenie

Proszę napisać program, który wyświetli napis „Ala ma kota” i będzie miał nazwę cwA1.py

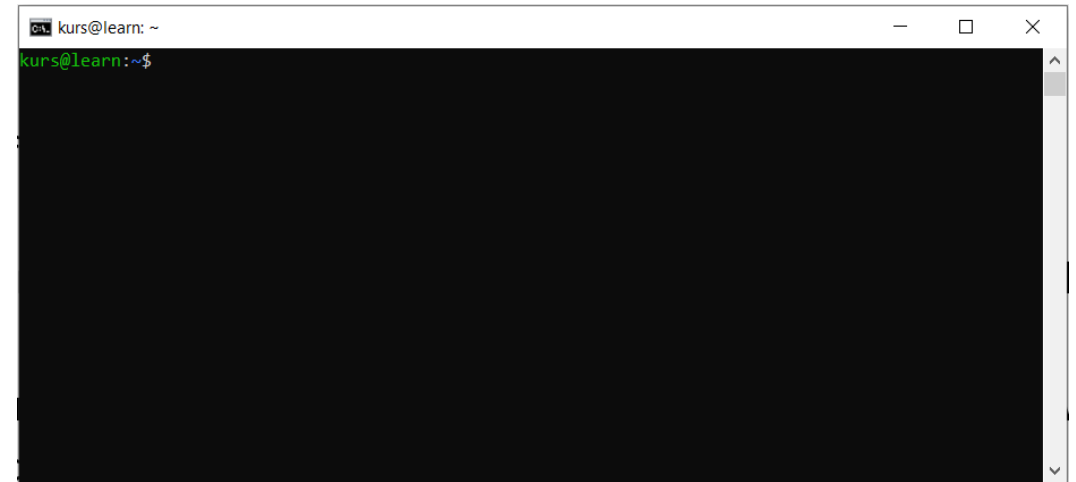
Oczekiwany wynik

Na dysku pojawi się program cwA1.py, który ma nadany atrybut wykonywalności

Po wydaniu polecenia ./cwA1.py w systemie Linux na ekranie pojawi się napis Ala ma kota

Uwagi

Do wyświetlenia na ekranie można użyć funkcji print. Więcej informacji na temat funkcji print znajdziesz tutaj [Python print\(\) Function \(w3schools.com\)](https://www.w3schools.com/python/python_print.asp)

A screenshot of a terminal window titled 'kurs@learn: ~'. The prompt 'kurs@learn:~\$' is visible on the first line, followed by a blank line. The terminal has a black background and a white scrollbar on the right side.

A.2 Wyświetlanie komunikatów

Dane wejściowe

-

Polecenie


Proszę napisać program, który wyświetli pierwszy dwuwiersz „Pana Tadeusza” w dwu kolejnych liniach. Do wyświetlenia danych proszę użyć funkcji print

Oczekiwany wynik

Na ekranie pojawią się 2 nowe wiersze tekstu

Uwagi

Do wyświetlenia na ekranie można użyć funkcji print. Więcej informacji na temat funkcji print znajdziesz tutaj [Python print\(\) Function \(w3schools.com\)](https://www.w3schools.com/python/python_print.asp)

A screenshot of a terminal window titled 'kurs@learn: ~'. The prompt 'kurs@learn:~\$' is visible on the first line, followed by a blank line. The terminal has a black background and a light gray border with standard window controls.

A.3 Formatowanie wyświetlania

Dane wejściowe

-

Polecenie

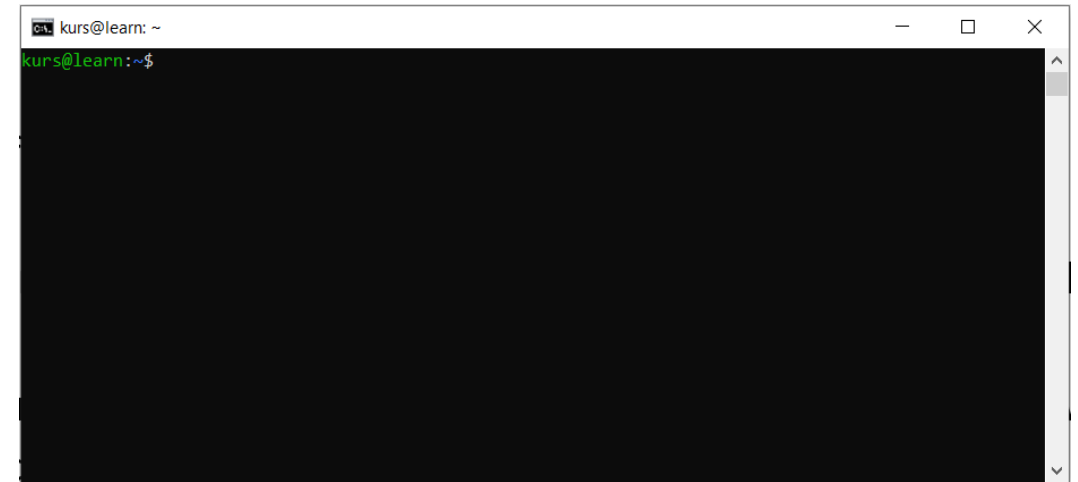
Proszę napisać program który wyświetli 15 początkowych liczb pierwszych , każda w osobnej linii, oddalona od krawędzi ekranu o jedną tabulację Do wyświetlania proszę użyć funkcji print

Oczekiwany wynik

Na ekranie pojawi się 15 nowych wierszy tekstu, będą to kolejne liczby pierwsze 2, 3, ...

Uwagi

- Do wyświetlenia na ekranie można użyć funkcji print. Więcej informacji na temat funkcji print znajdziesz tutaj [Python print\(\) Function \(w3schools.com\)](https://www.w3schools.com/python/python_print.asp)

A screenshot of a terminal window titled 'kurs@learn: ~'. The prompt 'kurs@learn:~\$' is visible on the first line, followed by a green cursor. The rest of the terminal area is black and empty.

A.4 Formatowanie wyjścia I

Dane wejściowe

Program cwA3.cpp

```
a = 15  
b = 17.5  
c = -1020
```

Polecenie

Napisz program w którym będą zainicjalizowane 3 zmienne a,b,c. Używając funkcji print i operatora % do interpolacji łańcucha znaków wyświetl:

- Zmienną a na 5 polach , brakujące pola mają być uzupełnione spacjami
- Zmienną a na 7 polach , brakujące pola mają być uzupełnione zerami
- Zmienną b na 7 polach z 3 liczbami po przecinku.
- Zmienną c w formatowaniu naukowym i w formatowaniu naturalnym.

Oczekiwany wynik

- Na ekranie pojawi się 5 liczb każda w osobnej linii , liczby wyświetlone są zgodnie z założeniami formatowania

Uwagi

- Do wyświetlenia na ekranie można użyć funkcji print. Więcej informacji na temat funkcji print znajdziesz tutaj [Python print\(\) Function \(w3schools.com\)](#)
- Do skonstruowania odpowiedniego łańcucha znaków możesz użyć operatora % lub funkcji format , lub formatted string literals (F-string)
 - [string — Common string operations — Python 3.12.7 documentation](#)
 - [Input and Output — Python 3.12.7 documentation](#)

cwA3.py

```
a = 15;  
b = 17.5;  
c = -1e20;
```

```
printf("");
```

A.5 Formatowanie wyjścia II

Dane wejściowe

Program cwA3.cpp

```
a = 15  
b = 17.5  
c = -1020
```

Polecenie

Napisz program w którym będą zainicjalizowane 3 zmienne a,b,c. Używając funkcji print i metody format z klasy string do interpolacji łańcucha znaków wyświetl:

- Zmienną a na 5 polach , brakujące pola mają być uzupełnione spacjami
- Zmienną a na 7 polach , brakujące pola mają być uzupełnione zerami
- Zmienną b na 7 polach z 3 liczbami po przecinku.
- Zmienną c w formatowaniu naukowym i w formatowaniu naturalnym.

Oczekiwany wynik

- Na ekranie pojawi się 5 liczb każda w osobnej linii , liczby wyświetlone są zgodnie z założeniami formatowania

Uwagi

- Do wyświetlenia na ekranie można użyć funkcji print. Więcej informacji na temat funkcji print znajdziesz tutaj [Python print\(\) Function \(w3schools.com\)](https://www.w3schools.com/python/python_print.asp)
- Do skonstruowania odpowiedniego łańcucha znaków możesz użyć operatora % lub funkcji format , lub formatted string literals (F-string)
 - [string — Common string operations — Python 3.12.7 documentation](https://docs.python.org/3.12/string.html)
 - [Input and Output — Python 3.12.7 documentation](https://docs.python.org/3.12/whatsnew/3.12.html#input-and-output)

cwA3.py

```
a = 15;  
b = 17.5;  
c = -1e20;
```

```
printf("");
```

A.6 Formatowanie wyjścia III

Dane wejściowe

Program cwA3.cpp

```
a = 15  
b = 17.5  
c = -1020
```

Polecenie

Napisz program w którym będą zainicjalizowane 3 zmienne a,b,c. Używając funkcji print i F-String do interpolacji łańcucha znaków wyświetl:

- Zmienną a na 5 polach , brakujące pola mają być uzupełnione spacjami
- Zmienną a na 7 polach , brakujące pola mają być uzupełnione zerami
- Zmienną b na 7 polach z 3 liczbami po przecinku.
- Zmienną c w formatowaniu naukowym i w formatowaniu naturalnym.

Oczekiwany wynik

- Na ekranie pojawi się 5 liczb każda w osobnej linii , liczby wyświetlone są zgodnie z założeniami formatowania

Uwagi

- Do wyświetlenia na ekranie można użyć funkcji print. Więcej informacji na temat funkcji print znajdziesz tutaj [Python print\(\) Function \(w3schools.com\)](#)
- Do skonstruowania odpowiedniego łańcucha znaków możesz użyć operatora % lub funkcji format , lub formatted string literals (F-string)
 - [string — Common string operations — Python 3.12.7 documentation](#)
 - [Input and Output — Python 3.12.7 documentation](#)

cwA3.py

```
a = 15;  
b = 17.5;  
c = -1e20;
```

```
printf("");
```

Dane wejściowe

a = 88
b = 25

Polecenie

W programie zadeklarowane są dwie zmienne a, b, które są typu int. Proszę napisać program który używając funkcji `print` wyświetli na ekranie:

- i. Zmienną a i zmienną b jako wartości hexadecymalne
- ii. Zmienną a i zmienną b jako wartości ósemkowe
- iii. Zmienną a i zmienną b jako wartości binarne

Oczekiwany wynik

- Na ekranie pojawi się 6 liczb, po dwie w jednej linii oddzielone spacją

Uwagi

- konwersji na wartość hexadecymalną można użyć funkcji `encode` z klasy `string`, lub funkcji `hex`, `oct`, `bin`
 - [string — Common string operations — Python 3.12.7 documentation](#)
 - [Built-in Functions — Python 3.12.7 documentation](#)
- lub jednej z metod formatowania wyjścia
 - [Input and Output — Python 3.12.7 documentation](#)

B.1 Wczytywanie danych z klawiatury

Dane wejściowe

łańcuch danych wpisany z klawiatury

Polecenie

- Napisz program, który wczyta z klawiatury łańcuch znaków a następnie 3 krotnie wyświetli go na ekranie Do wczytywania danych z klawiatury proszę użyć funkcji input

Oczekiwany wynik

- Na ekranie pojawią się trzy linijki tekstu

Uwagi

- Opis funkcji input można znaleźć tutaj [Python input\(\) Function \(w3schools.com\)](https://www.w3schools.com/python/python_input.asp)

B.2 Wczytywanie liczb z klawiatury la

Dane wejściowe

łańcuch znaków wpisany z klawiatury

Polecenie

Proszę napisać program który wczyta z klawiatury dwie liczby:

- i. liczbę całkowitą i wyświetli tą liczbę powiększoną o 15
- ii. liczbę zmiennoprzecinkową i wyświetli ją pomniejszoną o 7.2

Proszę zaobserwować co się stanie gdy wprowadzony łańcuch znaków nie jest prawidłową liczbą

Oczekiwany wynik

Na ekranie pojawią się 2 linie tekstu, wyświetlone liczby są zgodne z tymi wprowadzonymi

Uwagi

Opis funkcji scanf można znaleźć tutaj [scanf - cppreference.com](https://en.cppreference.com/w/cpp/string/basic_parse_int_results)

B.3 Wczytywanie liczb z klawiatury Ib

Dane wejściowe

łańcuch znaków wpisany z klawiatury

Polecenie

Proszę napisać program, używając w nim funkcji input, który wczyta z klawiatury trzy liczby:

- i. liczbę całkowitą typu i,
- ii. liczbę zmiennoprzecinkową
- iii. liczbę zmiennoprzecinkową .

Program ma za zadanie wyświetlić te liczby na ekranie oraz ich sumę. Każda z wyświetlonych liczb ma być w osobnej linii

Oczekiwany wynik

Na ekranie pojawią się 4 linie tekstu, wyświetlone liczby są zgodne z tymi wprowadzonymi, suma liczb jest poprawna. 4

Uwagi

B.4 Wczytywanie danych z predefiniowanym formatem

Dane wejściowe

łańcuch znaków wpisany z klawiatury o następującym formacie:

`<nazwa>, <nazwa>, <nazwa>`

gdzie `<nazwa>` to słowna nazwa koloru , nie zawierająca białych znaków

Polecenie

Proszę napisać program który wczyta nazwy 3 kolorów z klawiatury bądź standardowego wejścia. Przyjmujemy że nazwy kolorów są rozdzielone przecinkami np. Operacja wczytania danych powinna być wykonana jednym wywołaniem funkcji `input`

Oczekiwany wynik

Na ekranie pojawią się 3 linie tekstu,
Kolor1 : <kolor>,
Kolor2 : <kolor>,
Kolor3: <kolor>

Uwagi

Podzielenie łańcucha znaków na części można zrealizować funkcją [String.split\(\)](#)

B.5 Wczytywanie danych z predefiniowanym formatem la

Dane wejściowe

łańcuch znaków wpisany z klawiatury

Polecenie

Proszę napisać program, który w oparciu o funkcję input, wczyta dane z pojedynczego wiersza tekstu i wyświetli je na ekranie. Zakładamy że format wiersza tekstu jest następujący:

<zp>, _ <sp>: <sp> _ <sp> _ <zp> _ <zp>, <s>

gdzie: <zp> , <sp> to liczby zmiennoprzecinkowe i stałoprzecinkowe ,

<s> - łańcuch danych

_ - spacja

W łańcuchu znaków są następujące informacje, kolejno: grubość linii, numer warstwy, wierzchołki linii x0 , y0 , x1 , y1, nazwa linii.

Oczekiwany wynik

Na ekranie pojawią się napisy

Linia : <nazwa linii>

Grubość linii : <wartość>

Numer warstwy: <wartość>

P1 =(<wartość>, <wartość>) P2 = (<wartość>, <wartość>)

Uwagi

Podzielenie łańcucha znaków na części można zrealizować funkcją [String split\(\)](#)

B.6 Wczytywanie danych z predefiniowanym formatem lb

Dane wejściowe

łańcuch znaków wpisany z klawiatury

Polecenie

Proszę napisać program, który w oparciu o funkcje z modułu pyinputplus , wczyta dane z pojedynczego wiersza tekstu i wyświetli je na ekranie. Zakładamy że format wiersza tekstu jest następujący:

`<zp>, _ <sp>: <sp1> _ <sp2> _ <sp1> _ <sp2>, <s>`

gdzie: `<zp>` , `<spX>` to liczby zmiennoprzecinkowe i stałoprzecinkowe , `<sp1>` ma zakres od 0 do 800, `<sp2>` od 0 do 600

`<s>` - łańcuch danych może zawierać tylko znaki Alfanumeryczne i spację

`_` - spacja

W łańcuchu znaków są następujące informacje, kolejno: grubość linii, numer warstwy, wierzchołki linii x0 , y0 , x1 , y1, nazwa linii.

Oczekiwany wynik

Na ekranie pojawią się napisy

Linia : <nazwa linii>

Grubość linii : <wartość>

Numer warstwy: <wartość>

P1 =(<wartość>, <wartość>) P2 = (<wartość>, <wartość>)

W przypadku wprowadzenia nieprawidłowych danych , np. nie-liczby program poprosi o ponowne wprowadzenie danych

Uwagi

Aby zainstalować dodatkowy moduł pythona trzeba użyć polecenia pip

`pip install pyinputplus`

Jeśli moduł został zainstalowany można go użyć poleceniem `import pyinputplus`

Dodatkowe informacje na temat pyinputplus znajdziesz tutaj [PyInputPlus · PyPI](#)

B.7 Wczytywanie danych z predefiniowanym formatem lb

Dane wejściowe

łańcuch znaków wpisany z klawiatury

<nazwa_towaru>_<ilość_sztuk>

w którym nazwa towaru i ilość sztuk rozdzielona jest spacją. Nazwa towaru może zawierać spację w środku

Polecenie

Napisz program który wczyta z klawiatury nazwę towaru oraz ilość sztuk tego towaru a następnie wyświetli te informacje na ekranie. Do wczytania danych należy użyć jednej linii kodu i wywołania funkcji z modułu pyinputplus

Oczekiwany wynik

Na ekranie pojawią się napisy sformatowane w poniższy sposób

Towar1: <wczytana_nazwa_towaru> szt. <ilość_sztuk>

nazwa towaru ma być wyświetlana na 30 znakach, ilość sztuk ma być wyświetlana na 4 znaków

Uwagi

Dodatkowe informacje na temat pyinputplus znajdziesz tutaj [PyInputPlus · PyPI](#)

B.8 Wczytywanie danych z predefiniowanym formatem lb

Dane wejściowe

łańcuch znaków wpisany z klawiatury

Polecenie

Proszę napisać program który prosi użytkownika o wprowadzenie:

- i. Daty i czasu
- ii. Wartości stałoprzecinkowej z zakresu 1000, 1005
- iii. Wartości zmiennoprzecinkowej większej niż 4.2
- iv. Koloru , jako jednej z wartości na liście: red, green, blue
- v. Wartości True lub false

Oczekiwany wynik

Uwagi

Dodatkowe informacje na temat pyinputplus znajdziesz tutaj [PyInputPlus · PyPI](#)

C. Numeryczne typy danych w języku Python

Przy obliczeniach na liczbach stałoprzecinkowych python znacznie różni się od starszych języków programowania. Programista nie wybiera , jak przechowywać liczbę , ile bitów będzie miała liczba, nie dopasowuje jej do natywnych możliwości procesora. To wszystko automatycznie wykonuje interpreter pythona i podejmuje te decyzje za programistę. Począwszy od Pythona 2.5 liczby stałoprzecinkowe nie mają limitu górnego ponieważ python automatycznie przełącza się pomiędzy natywnymi typami procesora a liczbą bignum , która może mieć dowolną ilość bitów.

Nazwa typu	Ilość bitów	Najwyższa Liczba	Najniższa liczba
char	8	$2^7 - 1 = 127$	$2^7 = - 128$
short	16	$2^{15} - 1 = 32767$	$2^{15} = - 32768$
int	32	$2^{31} - 1 = 2147483647$	$2^{31} = - 2147483648$
long	64	$2^{63} - 1 = 2305843009213693951$	$2^{63} = - 2305843009213693952$
bignum	dowolna	dowolna	dowolna

Przy obliczeniach na liczbach zmiennoprzecinkowych python posługuje się natywnie zaimplementowaną w procesorze liczbą zmiennoprzecinką IEEE 745 double precision (double), która w nazewnictwie pythona nazywa się float. Obliczenia na tych liczbach są bardzo szybkie , jako że współczesny procesor posiada wiele równoległych potoków do obliczeń na nich. Z racji 64 bitowej reprezentacji liczby zmiennoprzecinkowe w pythonie mają ograniczoną dokładność. Jest to około 15-16 miejsc po przecinku

Nazwa typu natywnego	Ilość bitów	Największe liczby	Najmniejsze liczby x
IEEE 745 single precision	32	$\pm 3.402823466 * 10^{38}$	$\pm 3.402823466 * 10^{-38}$
IEEE 745 double precision	64	$\pm 1.7976931348623158 * 10^{308}$	$\pm 1.7976931348623158 * 10^{-30}$

Ograniczona dokładność reprezentacji liczb zmiennoprzecinkowych może produkować artefakty zaskakujące dla początkujących programistów
Liczby takie jak 1.1 i 2.2 nie mają swojej dokładnej reprezentacji. Ostatni znaczący bit mantysy jest zaokrągleniem w górę lub w dół przez to cała liczba jest zaokrągleniem w górę lub w dół.
Wynik dodawania tych dwu liczb nie jest równy dokładnie 3.3 ale jest równy 3.3000000000000003. Jest to efekt zaokrąglenia.

Inne obliczenie takie jak to 0.1 + 0.1 + 0.1 - 0.3 w matematyce jest równe dokładnie zero. W trakcie obliczeń na komputerze na liczbach float wynik jest następujący 5.5511151231257827e-017. Wartość jest bardzo bliska 0 ale nie jest to 0. Problemy pojawiają się podobnie jak w poprzednim wypadku w okolicy 16 miejsca po przecinku.

C. Dodatkowe typy danych do obliczeń zmiennoprzecinkowych

Aby rozwiązać problem niedokładnych obliczeń programiści pythona opracowali moduł decimal. Implementuje on dodatkowy typ danych [Decimal](#) , który pozwala on na operowanie na liczbach zmiennoprzecinkowych o powiększonej precyzji. Użytkownik w ramach konfiguracji może zdecydować jaka ma być precyzja obliczeń. Stosowanie liczb Decimal ma jednak swoje negatywne strony, wiąże się ze znacznie wolniejszymi obliczeniami. Typ Decimal trzeba stosować tylko tam gdzie jest to niezbędne

Obliczenia na liczbach Decimal stosowane są najczęściej przy obliczeniach finansowych, gdzie zaokrąglenia są niedopuszczalne

[illegible]

C. Deklarowanie zmiennych

Zmienna to taki składnik programu, który przechowuje dane niezbędne do działania programu. W trakcie wykonania programu zmienne zmieniają swoje wartości , zgodnie z zapisem w kodzie. Aby skorzystać ze zmiennej należy ją zainicjować czyli nadać jej wartość. Zmienna w języku Python ma swoją nazwę, ale nie ma sztywno przypisanego typu , może zmieniać swój typ każdorazowo, gdy wpisujemy do niej nową wartość, tzn. może być raz liczbą całkowitą, za chwilę ciągiem znaków, a jeszcze później liczbą zmiennoprzecinkową.

Python jest językiem obiekowym i w związku każda zmienna jest także obiektem. Jako obiekt posiada swoje składniki wewnętrzne , oraz metody (funkcje do wywołania). Aby dostać się do pól czy metod związanych ze zmienną trzeba użyć operatora . (kropka).

Skład pól i metod zależy od typu danych które są przechowywane w zmiennej np. : string , int ,float, tuple, list, dictionary

cwA6.cpp

```
# zmienna x jest skalar ( pojedynczą wartością )
x = 5

#zmienna y , jest łańcuchem znaków , a więc obiektem
y= "Ala ma asa"

# tak wywołujemy funkcję skojarzoną ze zmienną typu string
y.capitalize()
```

Zmienna może przechowywać pojedynczy element. To znaczy jest skalar

Może być listą wartości , lub listą par

cwA6.cpp

```
# ta zmienna jest listą wartości , inne nazwy to: wektor, tablica
# w każdej chwili można dodać do niej nowe wartości, usunąć bądź
# zmienić
x = [1,2,3,4,5]

# ta zmienna jest słownikiem ,przechowuje pary klucz wartość
# podobnie jak lista ta kolekcja jest „zmiennalna” ( mutable)
y= [ "Ala" : 4, "Ola" :5 ]

# Ta zmienna przechowuje krotkę ( tuple )
# krotki po utworzeniu nie można zmieniać
z = ( 2, 3,4,5 )
```

C. Rzutowanie

Gdy mówimy że język Python posiada zmienne które nie mają typu jest to pół prawdy. Wartość którą zmienna aktualnie przechowuje ma określony typ zarządzany przez interpreter Pythona , który determinuje w których obliczeniach zmienna może wziąć udział a w których nie. Bardzo często dochodzi do sytuacji w której chcemy zmienić „wewnętrzny” typ danych na inny. Taką operację nazywamy rzutowaniem

Następujące operacje rzutowania są w Pythonie

Operator konwersji	Typ źródłowy	Znaczenie
= int(...)	float, str	Konwersja z dowolnego typu na typ int
= float(...)	int, str	Konwersja z dowolnego typu na typ float
= str(...)	int, float	Konwersja do postaci łańcucha znaków

cwA6.cpp

```
# zmienna x jest skłarem ( pojedynczą wartością )
x = 5.0

#zmienna y jest łańcuchem znaków
y= str(x)

# zmienna z jest typu int
z = int(x)

print(type(x))
```

C. Moduł math

Znane z matematyki funkcje takie jak: sinus , cosinus , sinus hiperboliczny zostały zgromadzone w module math, Zapoznaj się z następującymi funkcjami [math \(Python 3.13.0\)](#)

Logarytmowanie

log	Logarytm przy podstawie e
log2	Logarytm przy podstawie 2
log1p	Logarytm przy podstawie e z 1+p
log10	Logarytm przy podstawie 10

Potęgowanie

pow	Potęga = x^y $x, y \in R$
sqrt	Pierwiastek \sqrt{x} $x \in R$

Funkcje trygonometryczne

sin	Sinus
cos	Cosinus
tan	Tangens
asin	Arcus sinus
acos	Arcus cosinus
atan	Arcus tangens
atan2	Arcus tangens dwuargumentowy

Funkcje hyperboliczne

sinh	Sinus hyperboliczny
cosh	Cosinus hyperboliczny
tanh	Tangens hyperboliczny
asinh	Arcus sinus hyperboliczny
acosh	Arcus cosinus hiperboliczny
atanh	Arcus tangens hyperboliczny

Dzielenie i Zaokrąglanie

ceil	Zaokrąglenie w górę $\lceil x \rceil$
floor	Zaokrąglenie w dół $\lfloor x \rfloor$
fmod	Reszta z dzielenia $\frac{x}{y}$ $x, y \in R$
remainder	Reszta z dzielenia $\frac{x}{y}$ $x, y \in R$

Funkcje informacyjne

isclose	Sprawdza czy dwie liczby są bliskie sobie
isfinite	Sprawdza czy liczba jest skończona
isnan	Sprawdza czy liczba jest NAN
nextafter	Następja większa liczba rzeczywista

Obliczenia na listach (wektorach)

sumprod	Iloczyn skalarny dwu wektorów $\sum_0^n x_i y_i$ $x, y - listy$
prod	Iloczyn wszystkich elementów wektora $\prod_0^N x_i$ $x - lista$

C.1 Obliczenia na zmiennych I

Dane wejściowe

- a, b, c – liczby wczytywane z klawiatury, bądź standardowego wejścia

Polecenie

Proszę napisać program który wczyta z klawiatury 3 liczby zmiennoprzecinkowe podwójnej precyzji: a , b , c , następnie wyświetli wartości:

- $a+b+b$
- $a^2+b^2+c^2$
- $a^3+b^3+c^3$

Oczekiwany wynik

- Na ekranie pojawiają się trzy liczby każda w osobnym wierszu

Uwagi

- Do potęgowania python posiada specjalny operator `**`

Dane wejściowe

liczby a, b wczytywane z klawiatury bądź standardowego wejścia

Polecenie

Proszę napisać program , który wczyta z klawiatury 2 liczby stałoprzecinkowe typu int. Ich wartości mają być umieszczone w zmiennych a , b. Zakładając że liczby a i b są różne od zera program ma wyświetlić wartość:

i. $\frac{a}{b}$

iii. $\frac{b}{a}$

ii. $\left\lfloor \frac{a}{b} \right\rfloor$ (dzielenie stałoprzecinkowe)

iv. $\left\lfloor \frac{b}{a} \right\rfloor$ (dzielenie stałoprzecinkowe)

Oczekiwany wynik

Na ekranie pojawiają się cztery liczby, każda w osobnym wierszu

Czy wyniki wyświetlone na ekranie są zgodne z oczekiwaniami ?

Uwagi

C.3 Obliczenia trygonometryczne I

Dane wejściowe

liczby a, b wczytywane z klawiatury bądź standardowego wejścia

Polecenie

Proszę napisać program który wczyta z klawiatury dwie liczby zmiennoprzecinkowe pojedynczej precyzji: a , b i wyświetli wartość

- i. $\sin(a)$
- ii. $\cos(b)$
- iii. $\cos(a) + \cos(b)$
- iv. przeliczy wartość a ze stopni na radiany
- v. przeliczy wartość b z radianów na stopnie

Oczekiwany wynik

Na ekranie pojawiają się pięć liczb, każda w osobnym wierszu

Uwagi

Opis funkcji trygonometrycznych można znaleźć tutaj [math — Mathematical functions](#)

C.4 Obliczenia trygonometryczne II

Dane wejściowe

liczby a , b , c wczytywane z klawiatury bądź standardowego wejścia będące bokami trójkąta

Polecenie

Proszę napisać program który wyliczy wszystkie 3 kąty trójkąta na podstawie długości ich boków
Program ma za zadanie wczytać z klawiatury długości 3 boków trójkąta i na ich podstawie wyliczyć
i wyświetlić wartości kątów trójkąta w radianach i w stopniach

Oczekiwany wynik

Na ekranie pojawiają się 3 wiersze, w każdym wierszu są dwie liczby: kąt w radianach i kąt w stopniach

Uwagi

Opis funkcji trygonometrycznych można znaleźć tutaj [math — Mathematical functions](#)

C.5 Obliczenia trygonometryczne III

Dane wejściowe

typ pomiaru: 1- wysokość górnej części ściany , 2 – wysokość dolnej części ściany, 3 – długość sufitu

liczby a, b: długości zmierzone dalmierzem

c : kąt odchylenia dalmierza od poziomu

Polecenie

Napisz program który wczyta dane wejściowe i wyliczy ich podstawie jedną z podanych niżej wielkości

- Wysokość górnej części ściany a (rys 1) lub
- Wysokość dolnej części ściany b (rys 1) lub
- Długość sufitu b (rys 2)

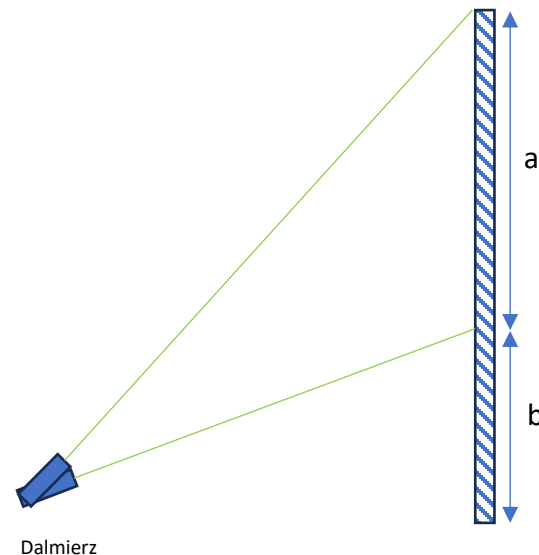
Oczekiwany wynik

Na ekranie pojawia się typ pomiaru i długość mierzonego odcinka

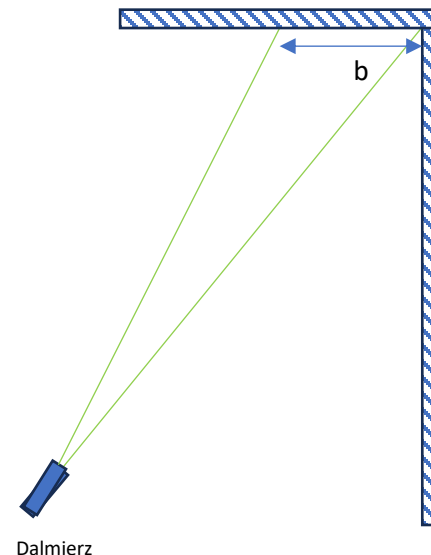
Uwagi

Opis funkcji trygonometrycznych można znaleźć tutaj [math — Mathematical functions](#)

Nowoczesne dalmierze laserowe wyposażone są w czujnik pochylenia , podaje on kąt pomiędzy wiązką promieniowania laserowego oraz poziomem. Dzięki temu nowoczesny dalmierz laserowy jest w stanie wyliczać nie tylko odległość pomiędzy dalmierzem a przeszkodą, ale również wymiary pośrednie np. wysokość ścian do których nie ma bezpośredniego dostępu.



Rys 1 Pomiar wysokości ściany



Rys 2. Pomiar długości sufitu

C.6 Zaokrąglanie

Dane wejściowe

liczba a wczytana z klawiatury bądź standardowego wejścia

Polecenie

Proszę napisać program , dla podanej z klawiatury liczby rzeczywistej a

- zaokrągli a w górę i wyświetli
- zaokrągli a w dół i wyświetli

Oczekiwany wynik

Na ekranie pojawią się dwa wiersze w każdym zaokrąglona liczba

Sprawdź jak zachowują się funkcje zaokrąglające dla liczb ujemnych

Uwagi

Dane wejściowe

liczby a , b wczytane z klawiatury bądź standardowego wejścia

Polecenie

Proszę napisać program który wczyta z klawiatury dwie liczby rzeczywiste: a , b . A następnie wyświetli

- | | |
|-------------------------------------|--|
| i. Logarytm z a przy podstawie 10 | v. $\sqrt[n]{b}$ |
| ii. Logarytm z a przy podstawie 2 | vi. Zaokrągli zmienną a do najbliższej całkowitej |
| iii. a^b | vii. Obetnie zmienną $(a+b)$ (usunie część ułamkową) |
| iv. e^b | viii. $ a $ |

Oczekiwany wynik

Na ekranie pojawi się osiem liczb , każda w osobnej linii

Uwagi

Dane wejściowe

typ figury - liczba 1 – trójkąt , 2 - trójkąt równoramienny , 3 - prostokąt , 4 - pięciokąt foremny

a,b,h,r, - wymiary figur wczytywane w zależności od typu figury

Polecenie

Proszę napisać program który poda pole powierzchni następujących figur na podstawie ich wymiarów.

- i. Trójkąta prostokątnego (a – podstawa , h - wysokość)
- ii. Trójkąta równoramiennego (b – długość ramienia , a – długość podstawy)
- iii. Prostokąta (a,b długości boków)
- iv. Pięciokąta foremnego (r – długość promienia okręgu opisanego na pięciokącie)

Oczekiwany wynik

Program zapyta o typ figury , jej wymiary i wyświetli na ekranie nazwę figury i jej pole powierzchni

Uwagi

C.9 Obliczenia geometryczne II

Dane wejściowe

n – ilość kątów n -kąta foremnego wczytywana z klawiatury lub standardowego wejścia

r - promień okręgu opisanego na n -kącie foremnym

Polecenie

Proszę napisać program który poda pole powierzchni n - kąta foremnego na podstawie :

- n - ilości wierzchołków w n -kącie
- r – promienia okręgu opisanego na n -kącie

Oczekiwany wynik

Na ekranie pojawi się liczba będąca polem powierzchni n -kąta foremnego

Uwagi

C.10 Obliczenia geometryczne III

Dane wejściowe

typ figury

a,h,r, - wymiary figur wczytywane w zależności od typu figury

Polecenie

Proszę napisać program który poda objętość następujących figur na podstawie ich wymiarów.

- stożka (r – promień podstawy , h - wysokość)
- ostrosłupa czworokątnego (a – długość boku , h – wysokość)
- graniastopuła o podstawie pięciokąta foremnego (a - boku podstawy , h – wysokość)
- ośmiościanu foremnego (r - promień okręgu opisanego na ośmiościanie)

Oczekiwany wynik

Program zapyta o typ figury , jej wymiary i wyświetli na ekranie nazwę figury i jej objętość.

Jeśli program będzie realizowany w trybie graficznym , należy wyrysować figurę i jej wymiary

Uwagi

D Operators

Języki Python posiada szereg instrukcji do przeprowadzania operacji obliczeniowych. Noszą one specjalistyczną nazwę "operatorów"

Każdy z tych operatorów symbolizuje konkretne działanie. Język programowania definiuje więcej operatorów niż podstawowa matematyka dlatego trzeba się z nimi zapoznać. Operatory które służą do obliczeń można podzielić na następujące grupy:

- | | |
|------------------------------------|-----------------|
| 1. Operatory arytmetyczne | + - / // * ** % |
| 2. Operatory logiczne | and or not |
| 3. Operatory porównania | < > <= >= == != |
| 4. Operatory bitowe | & ~ ^ |
| 5. Operatory przesunięcia bitowego | << >> |
| 6. Operatory identyczności | is is not |
| 7. Operatory przynależności | in not in |

Prócz nich język Python posiada operatory przesyłania danych

- | | |
|--|---|
| 1. Operator przypisania = | |
| 2. Operatory przypisania wraz z wykonaniem działania | += -= *= **= /= //= %= <<= >>= &= ^= = |

Operatory dostępu do składowej

1. Operator odwołania tablicowego []
2. Operator dostępu do składowych klasy .

Pozostałe operatory

1. operator wyliczeniowy ,

D Operatory arytmetyczne

Operatory arytmetyczne realizują

- | | |
|----------------------------------|----|
| 1. Dodawanie | + |
| 2. Odejmowanie | - |
| 3. Mnożenie | * |
| 4. Potęgowanie | * |
| 5. Dzielenie | / |
| 6. Dzielenie całkowitoliczbowe | // |
| 7. Obliczenie reszty z dzielenia | % |

i stosujemy je tak jak w matematyce podając dwa argumenty

```
arg1 operator arg2
```

W wyrażeniu złożonym z większej ilości operatorów. Istotna staje się kolejność wykonywania obliczeń

$$a * b + c$$

Kolejność wykonywania obliczeń w języku Python jest taka jak w matematyce

1. w pierwszej kolejności mnożenie i dzielenie
2. w następnej reszta z dzielenia
3. na końcu dodawanie i odejmowanie

Jeśli wyrażenie zawiera dwa lub więcej operatorów równorzędnych w pierwszej kolejności wykonywany jest ten po lewej stronie

Kolejność wykonania

2	+	12	*	8	+	7	%	12	+	4
3			1		4		2		5	

Dodawanie

```
5 + 5           // wynik 10
a + ( b + 15 )
a + b + 15
```

Mnożenie

```
2.5 * 5         // wynik 12,5
a * b * 15.0
a * a
```

Dzielenie

```
21 / 5          // wynik 4,02
a / b / 15.0
a * a
```

Reszta z dzielenia

```
12 % 5          // wynik 2
7 % 15 % 4      // wynik 3
```

Dzielenie całkowitoliczbowe

```
7 // 9          // wynik 0
7 // 9 // 4     // wynik 0
```

Potęgowanie

```
2 ** 8          // wynik 256
2 ** 3 ** 4     // wynik 2417851639229258349412352
```

D Operatory logiczne

Operatory logiczne reprezentują operacje znane z logiki matematycznej. Takie jak

- alternatywa (ang OR) `or`
- koniunkcja (ang AND) `and`
- negacja (ang NOT) `not`

Operatory logiczne `and` `or` są operatorami binarnymi potrzebują dwu argumentów.

Operator `not` jest operatorem unarnym potrzebuje tylko jednego argumentu

Argumentem operacji logicznych może być dowolna zmienna: czyli liczba, łańcuch znaków, tablica, lista krotka.

łańcuch znaków, tablica, lista lub krotka ma wartość Prawda jeśli nie jest pusta

liczba równa 0 jest traktowana jako FAŁSZ
liczba różna od zera jest traktowana jako PRAWDA

Wynikiem operacji logicznej jest zawsze liczba różna od zera (PRAWDA) lub (FAŁSZ)

Wynik operacji logicznej nie może być używany w obliczeniach arytmetycznych

cw_D1.py

```
#wartości poniżej obliczają się do wartości False
bool(False)
bool(None)
bool(0)

bool("") #pusty łańcuch znaków
bool(()) #krotka nie posiadająca elementów

bool([]) #pusta tablica
bool({}) #pusty słownik
```

cw_D.py

```
a = True
b = False

c = a and b; # iloczyn logiczny (koniunkcja)
c = c or b; # suma logiczna (alternatywa)

d = not c; # negacja logiczna

d = (a and b) + a; # ERROR , wynik operacji and nie musi być zerem lub jedynką
```

Koniunkcja (AND) &&

arg1 \ arg2	FAŁSZ	PRAWDA
FAŁSZ	FAŁSZ	FAŁSZ
PRAWDA	FAŁSZ	PRAWDA

Alternatywa (OR) ||

arg1 \ arg2	FAŁSZ	PRAWDA
FAŁSZ	FAŁSZ	PRAWDA
PRAWDA	PRAWDA	PRAWDA

D Operatory porównania

Operatory porównania służą do porównywania liczb

- Operator większości , mniejszości < >
- Operator równości ==
- Operator nierówności !=
- Operator większe równe >=
- Operator mniejsze równe <=

Wynikiem operacji porównania jest wartość true 1 lub false 0

Zero reprezentuje FAŁSZ

Jeden reprezentuje PRAWDĘ

Operatory porównania są operatorami binarnymi – wymagają dwu argumentów

`arg1 operator arg2`

Wynik działania operatora porównania jest zwyczajną liczbą i może brać udział w innych obliczeniach

`5 + (a > b)`

cw_C.py

```
a=5
b=2

c = a > b; # c ma wartość true
c = c == b; # c ma wartość false

d = ( a > b ) + a; # d ma wartość 6
```

D Operatory bitowe

Operatory bitowe to jedne z najpowszechniejszych operatorów w technice mikroprocesorowej niemalże każdy procesor ma rozkazy do ich wykonania. Operacje bitowe są bardzo proste do zaprojektowania na poziomie bramek i tranzystorów, są również bardzo szybkie w wykonaniu. Operatory bitowe pozwalają nam na modyfikowanie poszczególnych bitów danej zmiennej.

Mamy 4 operatory bitowe:

- Iloczyn bitowy (AND) &
- Alternatywa bitowa (OR) |
- Negacja bitowa (NEG) ~
- XOR bitowy ^

Można dokonać operacji bitowej jedynie na liczbach stałoprzecinkowych (int)

Operatory & | ^ są operatorami binarnymi potrzebują dwu argumentów
arg1 operator arg2

Operator negacji ~ jest operatorem unarnym
~a

Działanie operatorów bitowych zasadza się na założeniu że ta sama operacja wykonywana jest dla każdej pary bitów z obydwu operandów

Jeżeli mamy liczby a, b, c o reprezentacji bitowej:

a =	a _n	a _{n-1}	a _{n-2}	a _{n-3}	...	a ₃	a ₂	a ₁	a ₀
b =	b _n	b _{n-1}	b _{n-2}	b _{n-3}	...	b ₃	b ₂	b ₁	b ₀
c =	c _n	c _{n-1}	c _{n-2}	c _{n-3}	...	c ₃	c ₂	c ₁	c ₀

To w trakcie operacji bitowej zostanie wyznaczony wynik c którego bity c_n.. c₀ zostaną liczone jako

$$c_0 = a_0 \text{ operacja } b_0$$

$$c_1 = a_1 \text{ operacja } b_1$$

...

$$c_n = a_n \text{ operacja } b_n$$

Niektóre obliczenia matematyczne można przyspieszyć dzięki operatorom bitowym,
Na przykład:

reszta z dzielenia a % 2 to to samo co a & 2

wyznaczenie wartości bezwzględnej a & 0x80000000

cw_C.py

a=5

b=2

c = a & b; # c ma wartość 0

c = c | b; # c ma wartość 7

d = c ^ b; # c ma wartość 7

Iloczyn bitowy (AND) &

a _n \ b _n	0	1
FAŁSZ	0	0
PRAWDA	0	1

Alternatywa bitowa |

a _n \ b _n	0	1
0	0	1
1	1	1

Exclusive or (XOR) ^

a _n \ b _n	0	1
0	0	1
1	1	0

Negacja ~

a _n	0	1
c _n	1	0

D Operatory przesunięcia bitowego

Operatory te, podobnie jak operatory bitowe modyfikują indywidualne bity w liczbie. Przesuwają bity w liczbie o podaną ilość miejsc w prawo lub w lewo

Mamy dwa operatory przesuwania bitowego:

Operator przesunięcia w prawo >>

Operator przesunięcia w lewo <<

Operatory przesunięcia bitowego są operatorami binarnymi potrzebują dwu argumentów


$res = arg_1 \text{ operator } arg_2$

arg_1 - liczba której bity są przesuwane

arg_2 - ilość miejsc o które należy przesunąć liczbę

Przesunięcie bitowe o 2 pozycje na lewo działa w następujący sposób

arg1 =18	0	0	0	1	0	0	1	0	(18)
arg2 =2									(2)
res =80	0	1	0	0	1	0	0	0	(80)



matematycznie operacja ta jest równoznaczna z pomnożeniem liczby przez 2^x gdzie x jest liczbą przesuniętych bitów

cw_C.py

```
a=5
```

```
b=2
```

```
c = a << b; # c ma wartość 20
```

```
c = a >> b; # c ma wartość 1
```

D.1 Operacje bitowe

Dane wejściowe

kod źródłowy programu cw_C1.py

Polecenie

W programie przedstawionym obok zadeklarowane są dwie zmienne a, b i nadane są im wartości. Używając nie więcej niż 2 operatorów bitowych wylicz c tak by przyjęto wartość

- 2
- 12
- 1260

Oczekiwany wynik

kod źródłowy 3 programów, osobny program dla każdego z powyższych przykładów

Uwagi

- Opis operatorów bitowych można znaleźć na slajdach sekcji D lub tutaj [Python Operators \(w3schools.com\)](https://www.w3schools.com/python/operators/)

A screenshot of a code editor window titled "cw_C.py". The editor has a dark background and shows the following Python code:

```
a=5
b=27

c = ...
```

D.2 Ustawianie poszczególnych bitów

Dane wejściowe

- tekst programu cw_D2.py

Polecenie

W programie przedstawionym obok zadeklarowane jest zmienna `a` i nadana jest jej wartość.

- ustaw 5 bit zmiennej `a` na 1
- ustaw 2 bit zmiennej na 0
- zamień bity 0-7 na przeciwną wartość

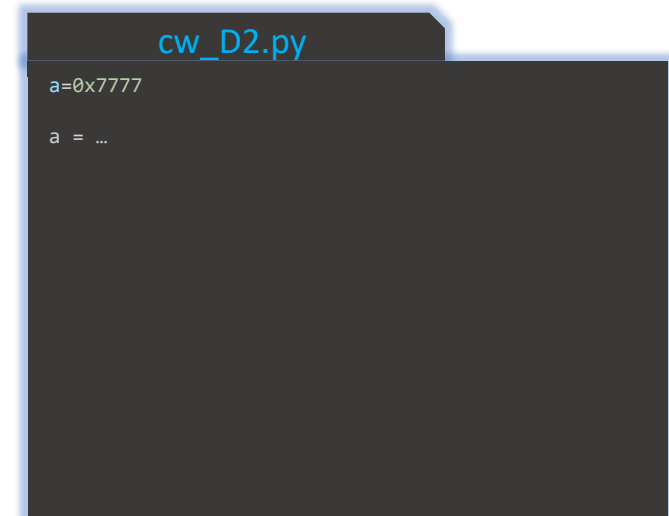
wolno użyć tylko jednego operatora bitowego

Oczekiwany wynik

- kod źródłowy 3 programów , osobny program dla każdego przypadku `a`, `b`, `c`

Uwagi

- Opis operatorów bitowych można znaleźć na slajdach sekcji D lub tutaj [Python Operators \(w3schools.com\)](https://www.w3schools.com/python/operators/)

A screenshot of a code editor window titled "cw_D2.py". The editor has a dark background and shows the following Python code:

```
a=0x7777  
  
a = ...
```

D.3 Ustawianie grup bitów

synchroniczna zmiana bitów jest kluczowa dla poprawnego działania urządzenia gdy oprogramujemy mikrokontroler, a zmieniana wartość oddziałuje na dołączone do systemu urządzenia elektroniczne i oznacza jego stan

Dane wejściowe

kod programu cw_D3.py

Polecenie

W programie przedstawionym obok zadeklarowana jest zmienna `a` i nadana jest jej wartość.

- a) zamień bity 0-7 na przeciwną wartość
- b) ustaw bity 3-5 na wartość 1 w jednej operacji
- c) ustaw bity 2-4 na wartość 0

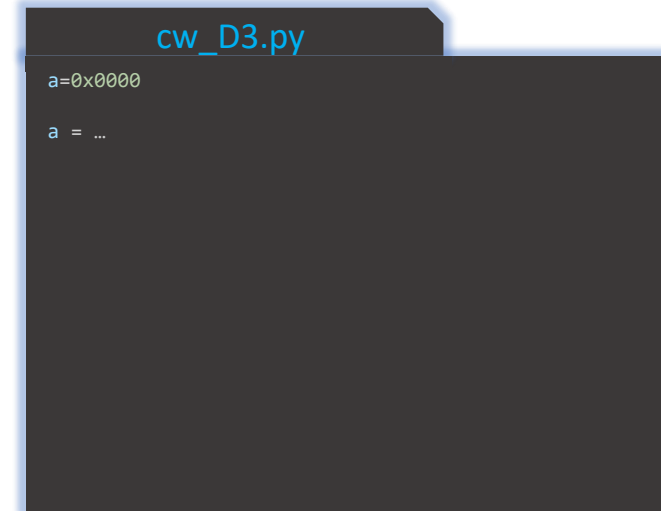
bity w zmiennej `a` muszą być zmienione w jednej operacji (synchronicznie)

Oczekiwany wynik

kod źródłowy 3 programów , osobny dla każdej opcji a, b, c

Uwagi

- Opis operatorów bitowych można znaleźć na slajdach sekcji D lub tutaj [Python Operators \(w3schools.com\)](https://www.w3schools.com/python/operators/)



```
cw_D3.py
a=0x0000
a = ...
```

D.4 Zadanie praktyczne I

Dane wejściowe

- kod źródłowy programu cw_D4.cpp
- statusDrukarki = 0xF423

Polecenie

Proszę napisać program który wyświetli w sposób czytelny dla człowieka stan drukarki

Oczekiwany wynik

Wszystkie informacje zawarte w bitach statusu drukarki muszą być wyświetlone w sposób zrozumiały dla użytkownika

Uwagi

- Opis operatorów bitowych można znaleźć na slajdach sekcji D lub tutaj [Python Operators \(w3schools.com\)](https://w3schools.com)

Elektronicy połączyli drukarkę specjalizowaną do systemu mikroprocesorowego. Drukarka widoczna jest w systemie jako 16 bitowa liczba statusDrukarki zlokalizowana w bloku pamięci printController. Odczytując poszczególne bity statusuDrukarki program może zorientować się jaki stan ma urządzenie elektroniczne. Znaczenie bitów statusu objaśnione jest poniżej

BIT	ZNACZENIE
0	1 - zasilanie włączone , 0 zasilanie wyłączone
1,2	Stan Drukarki - 00 – standby , 01 – loading data , 10 – printing , 11 – error
3,4,5	ilość zadań drukowania zawartych w pamięci drukarki
6-7	bity nieużywane , stan niekreślony
8-16	ilość pozostałych stron do wydrukowania w ramach aktualnie drukowanego zadania

cw_D4.py

```
# deklaracja zmiennej a i nadanie jej wartości 0xF423
statusDrukarki = 0xf423;
```

D.5 Zadanie praktyczne II

Dane wejściowe

-

Polecenie

Proszę zaprogramować robota by wykonał następującą pracę

1. Przesunął się do pozycji A z otwartym chwytakiem

2. Włączył laser

3. Przesunął się do pozycji B

4. Przesunął się do pozycji C

5. wyłączył laser

6. przesunął się do pozycji B

7. zamknął chwytak

8. przesunął się do pozycji A

9. otworzył chwytak

pomiędzy kolejnymi wydanyymi rozkazami należy odczekać co najmniej 1s

Oczekiwany wynik

kod źródłowy programu

Uwagi

Opis operatorów bitowych można znaleźć na slajdach sekcji D lub tutaj [Python Operators \(w3schools.com\)](https://w3schools.com)

Funkcja sleep zatrzymuje wykonanie programu na zadany czas [time — Time access and conversions — Python 3.13.0 documentation](https://docs.python.org/3.13.0/library/time.html#time.sleep)

Elektronicy podłączyli robota do komputera w ten sposób że jedna z komórek pamięci komputera połączona jest z wykonawczym układem elektronicznym robota. Jeśli zmieniamy wartość tej komórki pamięci , robot wykonuje działania zgodnie z wpisanymi danymi. Zmienna (komórka pamięci) nosi nazwę robotControl
Bity tej zmiennej mają następujące znaczenie

BIT	ZNACZENIE	Opis wartości
0	pozycja chwytaka	1 chwytak otwarty 0 - chwytak zamknięty
1,2	pozycja robota	00 -A, 01 - B, 10 - C, 11 - D
3	laser	1 – włączony 0 - wyłączony
4,5,6	bity nieużywane	, powinny być wyzerowane
7	zasilanie robota	1 - zasilanie robota włączone, 0 - robot wyłączony

E.1 Instrukcje wyboru - if

Dane wejściowe

pojedynczy znak wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury znak. Jeśli znak jest literą małą wypisze na ekranie napis „mała litera” , jeśli jest dużą literą wypisze na ekranie „duża litera” , jeśli nie jest literą wypisze , „znak specjalny”

Oczekiwany wynik

Na ekranie pojawia się napis „mała litera” lub „duża litera” lub „znak specjalny”

Uwagi

- Aby zrealizować to zadanie trzeba użyć instrukcji warunkowej [Control Flow Tools — Python 3.13.0 documentation](#)

cwA6.py

```
a = 4

if a > 0: # warunek, który oblicza się do wartości prawda , fałsz
    ...  # instrukcje do wykonania gdy warunek jest prawdziwy
else:
    ...  # instrukcje do wykonania gdy warunek jest fałszywy
```

E.2 Zadanie D.2 Instrukcja wielokrotnego wyboru match

Dane wejściowe

cyfra wczytana z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury cyfrę i wyświetli na ekranie słowną nazwę tej cyfry, np. 5 – „pięć”

Oczekiwany wynik

Na ekranie pojawia się słowna nazwa wprowadzonej cyfry

Uwagi

- Aby zrealizować to zadanie trzeba użyć instrukcji warunkowej match – case [4. More Control Flow Tools — Python 3.13.0 documentation](#)

cwA6.py

```
status = 202

match status:
    case 400:
        ... #instrukcje wykonywane gdy status jest równy 400
    case 404:
        ... #instrukcje wykonywane gdy status jest równy 404
    case 418:
        ... #instrukcje wykonywane gdy status jest równy 418
    case _:
        ... #instrukcje wykonywane we wszystkich pozostałych przypadkach
```

E.3 Definiowanie funkcji

Dane wejściowe

liczba zmiennoprzecinkowa `a` wczytana z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury liczbę i poda jej wartość bezwzględną. W trakcie realizacji programu nie wolno korzystać z funkcji bibliotecznych wyznaczających wartość bezwzględną.

Oczekiwany wynik

Kod źródłowy programu w którym zawarta jest funkcja `my_abs`

Uwagi

E.4 Instrukcja wyboru – zadanie praktyczne

Dane wejściowe

liczby a , b , c wczytywane z klawiatury bądź standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury 3 liczby zmiennoprzecinkowe a , b , c . Niech wczytane liczby będą współczynnikami równania kwadratowego

$$y = ax^2 + bx + c$$

Program ma wyświetlić na ekranie ilość miejsc zerowych oraz miejsca zerowe funkcji

Oczekiwany wynik

Na ekranie pojawia się ilość miejsc zerowych oraz miejsca zerowe funkcji kwadratowej, tyle ile funkcja ich ma lub pojawia się napis „brak miejsc zerowych”

Uwagi

Aby zrealizować to zadanie można użyć instrukcji warunkowej `if` lub instrukcji `match-case`

E.5 Instrukcja wyboru – wartość maksymalna

Dane wejściowe

- liczby a_1, a_2, a_3, a_4, a_5 wczytane z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program który pobiera z klawiatury 5 liczb stałoprzecinkowych i wyświetla największą z nich , taką która będzie parzysta , jeśli nie ma parzystej liczby wyświetla znak ‘-’

Oczekiwany wynik

Na ekranie pojawia się liczba lub znak „-”

Uwagi

- Aby zrealizować to zadanie można użyć instrukcji warunkowej [if](#) lub instrukcji switch – case

E.6 Złożony warunek logiczny

Dane wejściowe

liczba a wczytana z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program który pobiera z klawiatury liczbę , jeśli liczba jest większa od 1000 i jest podzielna przez 3 zmniejsza tę liczbę o połowę a następnie wyświetla liczbę

Oczekiwany wynik

Na ekranie pojawia się liczba o wartości zgodnej z poleceniem

Uwagi

F. Łańcuchy znaków

Język Python posiada odrębny typ danych przeznaczony do obsługi łańcuchów znaków – string. Jak w przypadku pozostałych typów danych Python zarządza nim w sposób automatyczny. Zmienna jest typu string jeśli taka dana została od niej przypisana.

łańcuch znaków w Pythonie jest niezmienny, nie wolno go modyfikować. Można go za to użyć do wytworzenia innego łańcucha znaków zgodnie z algorytmem.

łańcuchy znaków jest kolekcją pojedynczych znaków i możemy go przetwarzać tak jak pozostałe kolekcje (list, tuple). Mamy dostęp do każdego znaku z osobna poprzez operator []

```
lanc[<index>];
```

gdzie indeks to liczba całkowita z zakresu od 0 do długość łańcucha znaków

Python zapewnia wygodny dostęp zarówno do pojedynczego znaku w łańcuchy jak i zbioru znaków. Możemy użyć pętli for by przetworzyć każdy ze znaków łańcucha z osobna. Wyrażenie

```
lanc[a1:a2]
```

daje w wyniku łańcuch znaków składający się z liter o indeksach od a1 do a2-1

Jeśli a1 pominiemy przyjmowane jest 0 , jeśli a2 pominiemy przyjmowany jest koniec łańcucha znaków

cw_F1.py

```
# tak w języku Python deklarujemy łańcuch znaków, który jest niezmienny
# jego wartość nie zmienia i nie może się zmieniać w trakcie działania programu
x= "To jest łańcuch znaków";

# tak w języku Python deklarujemy wieloliniowy łańcuch znaków, który jest niezmienny
# jego wartość nie zmienia i nie może się zmieniać w trakcie działania programu

y = """To jest łańcuch znaków
składający się z wielu linii
"""

print(x[0]) #na ekranie zostanie wyświetlona litera T
```

cw_F2.py

```
lanc= "To jest łańcuch znaków";

a = lanc[3:5]          # a ma wartość "je"

b = lanc[3:]           # b ma wartość "jest łańcuch znaków"

c = lanc[:5]           # c ma wartość "To je"
```

F. Łańcuchy znaków w C++

Klasa string posiada całą różnorodność metod, użytecznych przy standardowych zadaniach dotyczących przetwarzania znaków . Każda z funkcji przetwarzających łańcuch znaków, zwraca nowy , przetworzony łańcuch jako wynik swojego działania

Funkcje informacyjne

Funkcja	Funkcja sprawdza łańcuch znaków i zwraca wartość True lub False
count	Ilość znaków w łańcuchu znaków
isalnum	True jeśli w łańcuchu są tylko znaki alfanumeryczne znaki które w zestawieni unicode zostały zakwalifikowane jako litery lub cyfry
isalpha	True jeśli w łańcuchu są tylko znaki alfanumeryczne znaki które w zestawieni unicode zostały zakwalifikowane jako litery
isascii	True jeśli w łańcuchu są tylko znaki alfanumeryczne znaki które są znakami ASCII (kod < 255)
isdecimal	True jeśli w łańcuchu są tylko znaki składające się na liczbę dziesiętną
isdigit	True jeśli w łańcuchu są tylko znaki składające się na liczbę dziesiętną
isidentifier	True jeśli w łańcuchu są tylko znaki składające się na identyfikator języka Python
islower	True jeśli w łańcuchu są tylko małe litery
isnumeric	True jeśli w łańcuchu są tylko cyfry
isprintable	True jeśli w łańcuchu są tylko znaki możliwe do wydrukowania
isspace	True jeśli w łańcuchu są tylko białe znaki
isupper	True jeśli w łańcuchu są tylko małe litery

Justyfikacja łańcucha znaków

Funkcja	Description
center	Wypełnia łańcuch znaków dodatkowymi znakami tak by go wycentrować
expandtabs	Specifies the amount of space to be substituted with the “\t” symbol in the string
ljust	Justyfikuje tekst do lewej na zadanej ilości znaków
lstrip	Usuwa spacje/znaki z początku łańcucha.
rjust	Justyfikuje tekst do prawej na zadanej ilości znaków
rstrip	Usuwa spacje/znaki z końca łańcucha znaków
strip	Usuwa spacje/znaki z początku i końca łańcucha znaków
zfill	Wypełnia łańcuch znaków zerami ,0’ od lewej strony aż osiągnie zadaną długość

Wyszukiwanie i zamiana łańcuchów znaków

Funkcja	Opis
casefold	Zwraca łańcuch znaków który jest przygotowany do przeszukiwania bez uwzględniania małych i dużych liter
endswith	Zwraca true gdy łańcuch znaków kończy się wyspecyfikowanym przyrostkiem
startswith	Zwraca true gdy łańcuch znaków kończy się wyspecyfikowanym przedrostkiem
find(sub)	Wyszukuje pierwszego wystąpienia łańcucha sub w łańcuchu i podaje gdzie się on zaczyna
index	Wyszukuje pierwszego wystąpienia łańcucha sub w łańcuchu i podaje gdzie się on zaczyna
replace(old,new)	Zamienia znalezione wystąpienia łańcucha old na łańcuch dest
rfind	Wyszukuje ostanie wystąpienie łańcucha sub w łańcuchu i podaje gdzie ono się zaczyna
rindex	Wyszukuje ostanie wystąpienie łańcucha sub w łańcuchu i podaje gdzie ono się zaczyna

Podział łańcucha na części

Function Name	Description
partition(sep)	Znajduje pierwsze wystąpienie separatora i dzieli w tym miejscu łańcuch znaków na łańcuch znaków przed separatorem, separator i po separatorze
rpartition(sep)	Znajduje ostatnie wystąpienie separatora i dzieli w tym miejscu łańcuch znaków na łańcuch znaków przed separatorem, separator i po separatorze
split(sep)	Dzieli łańcuch znaków na słowa . Słowa to znaki między separatorami
rsplit(sep)	Robi to samo co split ale zaczyna od prawej strony
splitlines	Dzieli łańcuch znaków na linie

Zamiana dużych i małych liter

Function Name	Description
capitalize	Zmienia pierwszą literę z łańcucha znaków na wielką literę
lower	Zmienia wszystkie litery z łańcucha znaków na małe litery
upper	Zmienia wszystkie litery z łańcucha znaków na wielkie litery
swapcase	Zamienia wszystkie litery z łańcucha znaków z wielkich na małe i z małych na wielkie
title	Zmienia wszystkie pierwsze litery
upper	Converts all lowercase characters in a string into uppercase

Inne

Function Name	Description
encode	Transluje łańcuch znaków pomiędzy standardami kodowania znaków narodowych (iso,utf, ascii)
format	Wytwarza łańcuch znaków na podstawie łańcucha formatującego formatu i danych
format_map	Formats specified values in a string using a dictionary
join	Łączy dwa łańcuchy znaków
translate	Zamienia znaki z łańucha na inne wynikające z tablicy translacji
maketrans	Ustawia zawartość tablicy translacji

F.1 Operator [] w stosunku do łańcucha znaków

Dane wejściowe

łańcuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program który wczyta z klawiatury łańcuch znaków i wyświetli na ekranie znaki:

- i. od pozycji 3 do pozycji 5
- ii. Od pozycji 3 do końca
- iii. Od początku do pozycji 7

Zakładamy że łańcuch znaków jest dłuższy niż 7 znaków

Oczekiwany wynik

Na ekranie pojawią się znaki w trzech kolejnych liniach

Uwagi

F.2 Konwersja łańcucha znaków

Dane wejściowe

łańcuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program , który wczyta z klawiatury łańcuch znaków, przekonwertuje go na wielkie litery i wyświetli na ekranie

Oczekiwany wynik

Na ekranie pojawi napis

Uwagi

F.3 Znajdź i zamień

Dane wejściowe

Łącuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program który wczyta z klawiatury łańcuch znaków, sprawdzi czy występuje w nim ciąg znaków „jest” i wyświetli informację na ekranie

Oczekiwany wynik

Na ekranie pojawi się napis „Znaleziono słowo jest” , lub „Nie znaleziono słowa jest”

Uwagi

F.4 Palindrom

Dane wejściowe

Łańcuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę wczytać z klawiatury łańcuch znaków i sprawdzić czy jest on palindromem. Palindrom czytany od przodu i od tyłu jest tym samym słowem np. TOWOT

Oczekiwany wynik

Na ekranie pojawi się napis „To jest palindrom” lub „To nie jest palindrom”

Uwagi

F.5 Konwersja liczb na tekst

Dane wejściowe

Łańcuch znaków który jest lub nie jest prawidłową liczbą

Polecenie

Proszę napisać program, który każdą z podanych poniżej zmiennych przekonwertuje na łańcuch znaków a następnie wyświetli powstałe łańcuchy znaków na ekranie jeden pod drugim.

- i. ala = 5.4;
- ii. ola = 2;
- iii. lista = [„Ala” , „ma” , 4, „koty”]

Oczekiwany wynik

Na ekranie powinny pojawić się 3 linie tekstu przedstawiające wyniki konwersji

Uwagi

Składnia języka

Wyświetlanie na ekranie tekstowym

F.6

Informacja o łańcuchach znaków

Dane wejściowe

łańcuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program który wczyta z klawiatury łańcuch znaków oraz wypisze na ekranie

- i. Czy znaki w łańcuchu należą do znaków ASCII
- ii. Czy znaku w łańcuchu są znakami alfanumerycznymi
- iii. Czy znaki w łańcuchu tworzą liczbę

Oczekiwany wynik

NA ekranie pojawią się linie tekstu. Każda linia zawiera słowo PRAWDA lub FAŁSZ zgodnie z algorytmem

Uwagi

AKSIM 2024

Podstawy programowania

F.7 Kodowanie znaków

Dane wejściowe

Liczba wczytana z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który pobierze z klawiatury liczbę , jeśli ta liczba jest mniejsza niż 255 potraktuje ją jako kod ASCII i wypisze: znak, odpowiadającą mu liczbę dziesiętnie i hexadecymalnie.

Jeśli liczba jest większa niż 255 program ma potraktować liczbę jako kod UNICODE i wyświetlić odpowiadający mu znak, kod dziesiętny i hexadecymalny.

Oczekiwany wynik

Na ekranie pojawi się linia składająca się ze znaku i dwu liczb , jednej dziesiętkowej , drugiej szesnastkowej

Uwagi

F.8 Przetwarzanie tekstu III

Dane wejściowe

Łącuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury łańcuch znaków i potraktuje to jako zdanie (wielu słów rozdzielonych spacjami). Proszę wyświetlić ten łańcuch znaków w taki sposób że każde słowo pisane będzie od końca, zaś kolejność słów nie ulega zmianie.

Oczekiwany wynik

Na ekranie pojawi się napis , będzie to zdanie w którym każde słowo będzie pisane od końca

„Ola ma psa„ -> „a1O am asp”

Uwagi

F.9 Przetwarzanie tekstu II

Dane wejściowe

Łącuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury łańcuch znaków tworzący zdanie. Proszę wyświetlić ten łańcuch znaków w taki sposób że odwrócona zostanie kolejność słów

Oczekiwany wynik

Na ekranie pojawi się napis zgodnie z algorytmem np. „asa ma Ala” dla wprowadzonego łańcucha znaków „Ala ma asa”

Uwagi

F.10 Przetwarzanie tekstu III

Dane wejściowe

Liczba całkowita wczytana z klawiatury lub standardowego wejścia

Polecenie

- Proszę napisać program który wczyta z klawiatury liczbę całkowitą i wyświetli ją w formie pisanej

Oczekiwany wynik

- Na ekranie pojawi się napis będący pisaną wersją wprowadzonej liczby np.
sto dwadzieścia trzy

Uwagi

F.9 Przetwarzanie tekstu IV

Dane wejściowe

Łącuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury łańcuch znaków, jest to liczba zapisana słownie w języku polskim, i przekonwertuje tą liczbę na formę binarną (int) i wyświetli ją powiększoną o 12

Oczekiwany wynik

Wejście : dwieście pięćdziesiąt dwa
Wyjście : 252

Uwagi

F.10 Przetwarzanie tekstu IV

Dane wejściowe

Łącuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury łańcuch znaków i zamieni wszystkie litery „r” w łańcuchu znaków na litery „w”. Wynikowy łańcuch znaków należy wyświetlić

Oczekiwany wynik

Na ekranie pojawi się łańcuch znaków , o zawartości zgodnej z poleceniem

Uwagi

F.11 Przetwarzanie tekstu V

Dane wejściowe

Łącuch znaków wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program który wczyta z klawiatury łańcuch znaków i wyliczy ile razy występuje w nim dwuznak „pa” i wyświetli

Oczekiwany wynik

Na ekranie pojawi się liczba

Uwagi

G.1 Zastosowanie pętli for

Dane wejściowe

-

Polecenie

Napisz program który , wyświetli na ekranie 100 kolejnych liczb nieparzystych

Oczekiwany wynik

Na ekranie pojawi się 100 wierszy tekstu , każdy wiersz zawiera liczbę nieparzystą większą niż poprzednia

Uwagi

- Aby zrealizować to zadanie trzeba użyć instrukcji pętli [Python For Loops](#)

G.2 Pętla while i do while

Dane wejściowe

znak wczytany z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program, który wczyta z klawiatury znak , jeśli znak jest małą literą ma ją wyświetlić na ekranie , jeśli jest innym znakiem , ma ponownie poprosić o wczytanie znaku. Czynność ma być powtarzana tak długo aż użytkownik wpisze małą literę

Zadanie proszę wykonać na z pętlą while

Oczekiwany wynik

Po wprowadzeniu małej litery program wyświetli ją na ekranie i zakończy działanie

Uwagi

Aby zrealizować to zadanie trzeba użyć instrukcji pętli [Python While Loops](#)

G.3 Ciąg Fibonacciego

Dane wejściowe

N- ilość liczb do wygenerowania , wartość pobrana z klawiatury lub ze standardowego wejścia

Polecenie

Proszę napisać program , który wylicza i wyświetla kolejne liczby ciągu Fibonacciego. Program na starcie powinien pobrać ilość liczb do wyświetlenia. Proszę Napisać program w dwu wersjach: użyciem rekurencji i bez użycia rekurencji

Proszę wyznaczyć czas działania obydwu programów i dowiedzieć się który z nich jest szybszy. I o ile procent

Oczekiwany wynik

- Na ekranie pojawiają się liczby z ciągu Fibonacciego , rozdzielone spacją

Uwagi

G.4 Liczba pierwsza

Dane wejściowe

N – liczba całkowita wczytana z klawiatury lub ze standardowego wejścia

Polecenie

Proszę napisać program, który sprawdzi czy podana z klawiatury liczba jest liczbą pierwszą.

Oczekiwany wynik

Na ekranie pojawi się napis „Liczba jest/nie jest liczbą pierwszą”

Uwagi

G.5 Suma cyfr

Dane wejściowe

Liczba całkowita wczytana z klawiatury lub ze standardowego wejścia

Polecenie

Proszę napisać program , który wczyta z klawiatury liczbę całkowitą następnie wyznaczy i wypisze na ekran sumę jej cyfr.

Oczekiwany wynik

Na ekranie pojawi się jedna liczba zgodna z poleceniem

Uwagi

Dane wejściowe

- Liczba całkowita wczytana z klawiatury lub standardowego wejścia

Polecenie

Proszę napisać program , który na podstawie podanej na wejściu liczby całkowitej wyznaczy liczbę lustrzaną dziesiętną i binarną. Liczba lustrzana dziesiętna jest reprezentowana ciągiem cyfr, który jest lustrzanym odbiciem oryginału. Liczba lustrzana binarna jest reprezentowana ciągiem zer i jedynek który jest lustrzanym odbiciem liczby wejściowej

Oczekiwany wynik

Na ekranie pojawią się dwa wiersze tekstu ,

- w pierwszym wierszu będzie wczytana liczba i jej liczba lustrzana obydwie w zapisie dziesiętnym
- w drugim wierszu będzie wczytana liczba i jej liczba lustrzana obydwie w zapisie hexadecymalnym

Uwagi

G.7 Zadanie E.7 Elementy ciągu

Dane wejściowe

- $n_{\text{start}}, n_{\text{end}}$ - liczby całkowite wczytywane z klawiatury lub ze standardowego wejścia

Polecenie

Proszę napisać program który wczyta z klawiatury wyznaczy sumę elementów ciągu

$$F(n) = n^3 + 3n^2 + 5n - 45 \quad \text{dla } n \in \mathbb{C}$$

w zakresie $\langle n_{\text{start}}, n_{\text{end}} \rangle$

Oczekiwany wynik

Na ekranie pojawi się liczba

Uwagi

Dane wejściowe

Polecenie

Proszę napisać program, który poprawnie wyznaczy silnię 10002. Jak wiele bitów musi mieć liczba by poprawnie i bez żadnych zaokrągleń wyliczyć wynik

Oczekiwany wynik

- Ciąg cyfr będący silnią liczby 10002
- Ilość bitów wyniku

Uwagi

G.8 Obliczenia na wielu wątkach

Dane wejściowe

Kod programu cw_G8 przedstawiający sposób tworzenia nowych wątków w pythonie

Polecenie

Proszę napisać program, który poprawnie bez zaokrążeń wyznaczy silnię liczby 10002. Opracowany program ma być wielowątkowy i wykorzystywać około 100% mocy obliczeniowej procesora. Klasy i funkcje dotyczące wielowątkowości zawarte są w module threading

Oczekiwany wynik

Ciąg liczb stanowiący silnię liczby 10002

Uwagi

Wątki nie są zsynchronizowane ze sobą w żaden sposób, czas wykonywania się wątku należy przyjmować jako nieznany. Programista nie powinien czynić założeń że jeden wątek skończy się wcześniej niż drugi , bo ma mniej zadań do wykonania.

Przy pisaniu programów wielowątkowych należy zwrócić uwagę by wątki nie zapisywały do tej samej zmiennej. Nie wiadomo który wątek zapisze dane jako pierwszy a który jako drugi.

Niebezpieczne jest również to gdy jeden wątek czyta dane ze zmiennej a drugi do niej zapisuje. Może się zdarzyć taka sytuacja że zmienna będzie zaktualizowana częściowo gdy wątek czytający do niej sięgnie.

cw_G8.py

```
import threading

# funkcja która ma być uruchomiona jako nowy wątek
def calculate_c1(num, num1):
    pass

# funkcja która ma być uruchomiona jako nowy wątek
def calculate_c2(num, num1):
    pass

t1 = threading.Thread(target=calculate_c1, args=(10,100))
t2 = threading.Thread(target=calculate_c2, args=(100,200))

# uruchomienie wątków t1 i t2 , funkcje calculate_c1 calculate_c2 zaczynają się wykonywać
t1.start()
t2.start()

# teraz nasz program ma 3 wątki , wątek główny , t1 ,t2

# czekamy na zakończenie się funkcji calculate_c1
t1.join()

# czekamy na zakończenie się funkcji calculate_c2
t2.join()

print("Done!")
```

H.1 Odczyt plików tekstowych

Dane wejściowe

Plik danych ludzie.txt



ludzie.txt
(kliknij na ikonę by otworzyć)

Polecenie

Proszę wczytać dane z pliku ludzie.txt i wyświetlić je na ekranie. Plik zawiera imię , nazwisko , wzrost , wagę i datę urodzenia człowieka.

Oczekiwany wynik

Na ekranie pojawią się 3 wiersze o następującej strukturze

1 : <imie> , 2: <nazwisko>, 3: <data urodzenia>, 4: <waga>

Uwagi

Przy realizacji zadania proszę zapoznać się z funkcjami [open](#) , [read](#) , [write](#)

H.2 Zapis do plików tekstowych

Dane wejściowe

-

Polecenie

Proszę wyliczyć 100 kolejnych elementów ciągu fibonnaciego i zapisać wyniki do pliku fibbonacci.txt

Oczekiwany wynik

Plik fibonnaci.txt będzie zawierał 100 linii tekstu , każda linia zawiera kolejną liczbę fibonacciego

Uwagi

Przy realizacji zadania proszę zapoznać się z funkcjami [open](#) , [read](#) , [write](#)

H.3 Odczyt pliku od zadanej pozycji

Dane wejściowe

- plik tekstowy PanTadeusz.txt
- N – pozycja początkowa w pliku, liczba wczytywana z klawiatury
- M- ilość znaków do wyświetlenia, liczba wczytywana z klawiatury



PanTadeusz.txt

kliknij by zapisać plik na dysku

Polecenie

Proszę napisać program , który wyświetla na ekranie fragment pliku PanTadeusz.txt. Fragment obejmuje N liter , począwszy od pozycji M w pliku.

Oczekiwany wynik

- Na ekranie pojawia się tekst składający się z M znaków, będący fragmentem Pana Tadeusza

Uwagi

H.4 Modyfikacja pliku tekstowego

Dane wejściowe

plik tekstowy ludzie.txt



ludzie.txt

kliknij by zapisać plik na dysku

Polecenie

Proszę napisać program który zmodyfikuje zawartość pliku ludzie.txt. Imię Pana Nowickiego powinno być zamienione na Abelard. Przed modyfikacją pliku program powinien utworzyć plik ludzie.bak, który jest kopią zapasową oryginalnego pliku ludzie.txt (sprzed zmiany nazwy)

Oczekiwany wynik

- Na dysku powstanie pliku ludzie.bak
- Plik ludzie.txt zmieni swoją zawartość

Uwagi

H.5 Zapis do pliku binarnego

Dane wejściowe

- -

Polecenie

Proszę stablicować funkcję sinus , to jest zapisać do pliku sinus.bin wartości funkcji $\sin(\alpha)$.
Zapisane wartości funkcji sinus muszą być z przedziału $\alpha \in (0, \pi)$ z krokiem 0.001 radiana.
($\sin(0)$, $\sin(0.001)$, $\sin(0.002)$ $\sin(2\pi)$). Zapis powinien być wykonany w trybie binarnym.

Oczekiwany wynik

Powstanie binarny plik sinus.bin w którym będą zapisane wartości funkcji sinus

Uwagi

Przy realizacji zadania proszę zapoznać z trybem binarnym otwarcia pliku

H.6 Odczyt z pliku binarnego

Dane wejściowe

- plik binarny data.pmi



data.pmi

kliknij by zapisać plik na dysku

Polecenie

- Proszę napisać program , który wczyta z plik data.pmi i wypisze na ekranie jego zawartość w sposób czytelny dla człowieka
- to jest wypisze datę utworzenia , wersję oraz listę figur wraz z położeniem ich wierzchołków

Oczekiwany wynik

Uwagi

Plik binarny data.pmi jest plikiem graficznym. Jest on listą linii, kwadratów i prostokątów. Dane w pliku zostały zapisane w konkretnym formacie, oto on:

Bajty początkowy	Ilość Bajtów	Opis
0	3	(identyfikator pliku , zawsze to powinny być litery PMI)
3	8	(data utworzenia pliku - epoch time)
11	1	(wersja pliku)

począwszy od bajtu 12 zaczyna się lista figur.

Definicja figury rozpoczyna się od typu figury , po której występuje lista wierzchołków. W przypadku linii są to dwa wierzchołki , w przypadku trójkąta są trzy wierzchołki , prostokąta cztery wierzchołki. Bezpośrednio po wierzchołkach rozpoczyna się definicja następnej figury

Bajty początkowy	Ilość Bajtów	Opis
n	1	0 - linia, 1 – trójkąt , 2 – prostokąt , inne wartości zabronione
n+1	2	współrzędna X pierwszego wierzchołka
n+3	2	współrzędna Y pierwszego wierzchołka
...		
n+2k+1	2	współrzędna X k-tego wierzchołka
n +2k+3	2	współrzędna Y k-tego wierzchołka

Dane wejściowe

-

Polecenie

Proszę napisać program który będzie utrzymywał bazę danych zwierząt w zoo. Baza danych ma być zlokalizowana w pliku baza.bin i być w formacie binarnym. Baza danych ma zawierać spis wszystkich gatunków zwierząt w zoo wraz z ich ilościami. Np. flamingi – 15 szt. Prócz ilości zwierząt ma baza ma zawierać średnią wagę pożywienia spożywanego przez dany gatunek zwierząt, typ pożywienia np. ziarno, mięso oraz koszt utrzymania w złotychkach. Napisany program powinien mieć następujące możliwości .

- i. Dodanie nowego gatunku zwierząt
- ii. Usunięcie gatunku zwierząt
- iii. Modyfikacja ilości sztuk
- iv. Modyfikacja typu pożywienia ,
- v. Modyfikacja średniej wagi pożywienia
- vi. Modyfikacja kosztu utrzymania
- vii. Wyświetlanie zawartości bazy danych (wszystkie gatunki)
- viii. Wyświetlanie danych wybranego gatunku

Oczekiwany wynik

Na dysku powstanie binarny plik baza.bin. , który będzie modyfikowany zgodnie z poleceniami i – vi,

Na żądanie użytkownika (vii , viii)dane z bazy danych zostaną zaprezentowane na ekranie

Uwagi

H.8 Odczyt z pliku o powszechnie znanym formacie

Dane wejściowe

dowolny plik w formacie BMP (bitmapa)

Polecenie

Proszę napisać program, który odczyta nagłówki pliku bmp i wyświetli zawarte tam informacje w formie tekstowej czytelnej dla człowieka. Plik BMP zawiera 2 nagłówki BITMAPFILEHEADER, BITMAPINFOHEADER.

Oczekiwany wynik

Na ekranie pojawi się zawartość nagłówka BITMAPFILEHEADER wypisana w sposób czytelny dla człowieka pod spodem będzie wypisana zawartość BITMAPINFOHEADER

Uwagi

Plik BMP ma dwa nagłówki na początku. Tu znajdziesz ich opis

- [BITMAPINFOHEADER](#)
- [BITMAPFILEHEADER](#)

I.1 Rysowanie figur geometrycznych

Dane wejściowe

-

Polecenie

- Proszę napisać program, który narysuje trójkąt t na ekranie graficznym używając biblioteki sGraphics

Oczekiwany wynik

Na ekranie pojawi się okno graficzne, w oknie graficznym pojawi się trójkąt.

Uwagi

- Proszę zapoznać się z biblioteką sGraphics

I.2 Rysowanie figur geometrycznych

Dane wejściowe

-

Polecenie

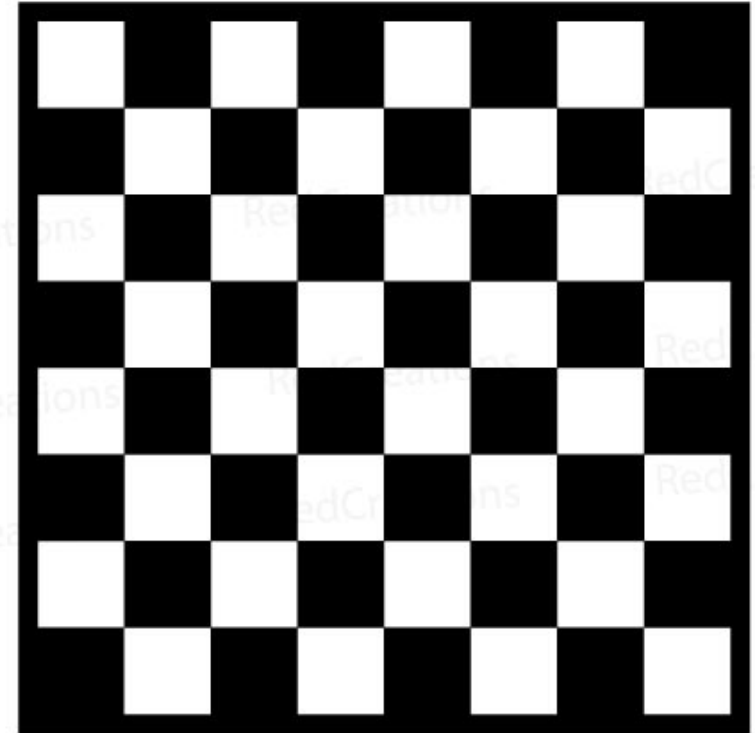
- Proszę napisać program, który narysuje na ekranie szachownicę do gry w szachy (czarno , białą)

Oczekiwany wynik

NA ekranie graficznym pojawi się szachownica , szachownica obejmuje cały ekran

Uwagi

Proszę zapoznać się z API biblioteki sGraphics



I.3 Rysowanie figur geometrycznych

Dane wejściowe

-

Polecenie

Proszę napisać program, który narysuje na ekranie 15 koncentrycznych kół każde w innym kolorze

Oczekiwany wynik

Na ekranie graficznym pojawi się na środku ekranu 14 koncentrycznych pierścieni i , kółko w środku

Uwagi

-

I.4 Rysowanie figur geometrycznych

Dane wejściowe

N – liczba wierzchołków n-kąta, liczba wczytana z klawiatury

r – promień koła wpisanego do wielokąta

Polecenie

Proszę napisać program, który narysuje na ekranie n-kąt foremny

- trójkąt równoramienny (3 kąty),
- kwadrat(4),
- pięciokąt foremny,
- sześciokąt foremny ,.... .

W zależności od parametrów n i r wczytywanych z klawiatury

Oczekiwany wynik

- W centralnej części ekranu powstanie figura geometryczna zgodna z wprowadzonymi danymi

Uwagi

I.5 Tworzenie prostych ikon

Dane wejściowe

Polecenie

Proszę napisać program który narysuje na ekranie ikonę, a następnie zapisze ją do pliku png



Oczekiwany wynik

Na ekranie graficznym powstanie ikona jak na obrazku obok.

Na dysku pojawi się plik ico.png , w którym będzie ikona

Uwagi

I.6 Tworzenie prostych ikon II

Dane wejściowe

Polecenie

Proszę napisać program, który narysuje na ekranie ikonę a następnie zapisze ją na dysk w postaci pliku bmp

Oczekiwany wynik

Na ekranie pojawi się ikona taka jak na obrazie obok

Na dysku pojawi się plik ico.bmp zawierający ikonę.



Uwagi

Dane wejściowe

Polecenie

- Proszę napisać program, który odczytuje plik w formacie HPGL (HP Graphics Language) i rysuje jego zawartość na ekranie. Program powinien implementować podzbiór rozkazów HPGL, obowiązkowo musi rozpoznawać komendy PA, PR, PD, PU , SP , CI

Oczekiwany wynik

Uwagi

- Plik HPGL można wygenerować za pomocą programu Corel Draw (opcja export) lub za pomocą programu ImageMagic

Dane wejściowe

Polecenie

- Proszę napisać program który wygeneruje zbiór mandelbrota na ekranie komputera (fraktal mandelbrota)

Oczekiwany wynik

Uwagi

I.9 Kalejdoskop

Dane wejściowe

- a – bok trójkąta , który jest polem operacyjnym kalejdoskopu

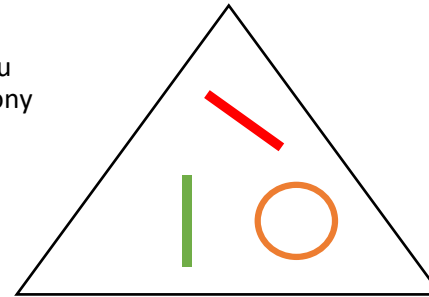
Polecenie

Proszę napisać program który wytworzy obraz tak sam jak kalejdoskop , dla 3 elementów w polu operacyjnym kalejdoskopu. Polem operacyjnym kalejdoskopu jest trójkąt równoboczny położony centralnie w środku obrazu . Elementami są :

- 2 prostokąty o długości r i szerokości $0.1r$, każdy w innym kolorze
- kolorowy pierścień o promieniach r , $1.1r$

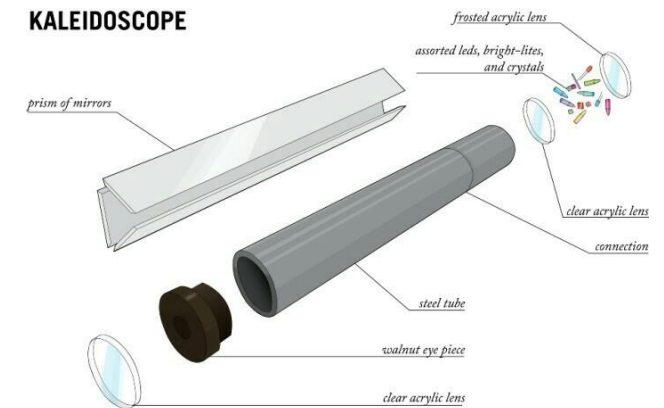
Położenie figur powinno być losowane na starcie programu

Oczekiwany wynik



W latach dziewięćdziesiątych popularnością cieszyła się zabawka dla dzieci zwana kalejdoskopem . Miała ona kształt walca, którego jedno dno jest półprzepuszczalne a drugie zawiera małą dziurkę obserwacyjną. Wewnątrz tuby są 4 lustra ustawione tak by tworzyły trójkąt równoboczny , lustra rozciągają się na całą wysokość tuby. Formalnie tworzą one graniastosłup o podstawie trójkąta równobocznego. Lustra odbijają światło od siebie. Na dnie kalejdoskopu znajdują się drobne elementy z kolorowego plastiku, takie jak patyczek, pierścień, kawałek łuku. Po potrząśnięciu zabawki elementy ustawiają się losowo na dnie kalejdoskopu. Światło od odbite od tych elementów i odbite wielokrotnie od lusterek tworzy przepiękny, symetryczny wzór.

KALEIDOSCOPE



źródło: [Explain the working of a kaleidoscope? - Brainly.in](https://www.brainly.in/explain-the-working-of-a-kaleidoscope/)

Uwagi

J.1 Wypełnianie tablicy

Dane wejściowe

A_0, A_1 liczby początkowe ciągu

Polecenie

Napisz program , który wypełnia tablicę 100 liczb zmiennoprzecinkowych kolejno liczbami według wzoru. $A_0 = 1$, $A_{n+1} = 2 * A_n$.

W programie zawrzyj funkcję która wykonuje zadanie wypełniania tablicy. Adres do tablicy ma być argumentem funkcji.

Oczekiwany wynik

Kod źródłowy programu zawierający funkcję fill_table

Uwagi

J.2 Tablice dwuwymiarowe

Dane wejściowe

rozmiar tablicy A = 40x40

Polecenie

Napisz program, który wypełnia dwuwymiarową tablicę liczb stałoprzecinkowych kolejno liczbami według wzoru. $A_{i,j} = i+j$, gdzie A jest nazwą tablicy zaś i, j są indeksami

Oczekiwany wynik

- Tablica wypełniona jest jak na rysunku obok

0	1	2	...	37	38	39
1	2	3		38	39	40
2	3	4		39	40	41
...						
37	38	39	
38	39	40	
39	40	41		78

Uwagi

J.3 Tablice dwuwymiarowe

Dane wejściowe

rozmiar tablicy A = 40x40

Polecenie

Napisz program, który wypełnia dwuwymiarową tablicę liczb stałoprzecinkowych kolejno liczbami według wzoru. $A_{i,j} = i+j$, gdzie A jest nazwą tablicy zaś i, j są indeksami

Zadanie wypełniania tabeli realizuje funkcja `table_init` , której argumentem jest wskaźnik do tablicy

Oczekiwany wynik

- Tablica wypełniona jest jak na rysunku obok
- Kod źródłowy zawiera prawidłowo zadeklarowaną funkcję `table_init`

0	1	2	...	37	38	39
1	2	3		38	39	40
2	3	4		39	40	41
...						
37	38	39	
38	39	40	
39	40	41		78

Uwagi

J.4 Operacje na tablicach dwuwymiarowych

Dane wejściowe

- tablica o wymiarach $N \times M$ wczytywana z klawiatury lub z pliku
- N, M rozmiary tablicy również są definiowane przez użytkownika

Polecenie

Proszę napisać program który wczytuje z klawiatury lub z pliku tablicę dwuwymiarową , a następnie zmienia kolejność wierszy w tablicy w taki sposób, że wiersz pierwszy ma się znaleźć na miejscu drugiego, wiersz drugi ma się znaleźć na miejscu trzeciego itd., natomiast ostatni wiersz ma się znaleźć na miejscu pierwszego . Zmienioną tablicę należy wypisać na ekran lub zapisać do pliku

Oczekiwany wynik

- Na ekranie (w pliku) pojawi się zmodyfikowana tablica

Uwagi

J.5 Tablice 3 wymiarowe

Dane wejściowe

a – wymiar tablicy

b – rozmiar boku kwadratu

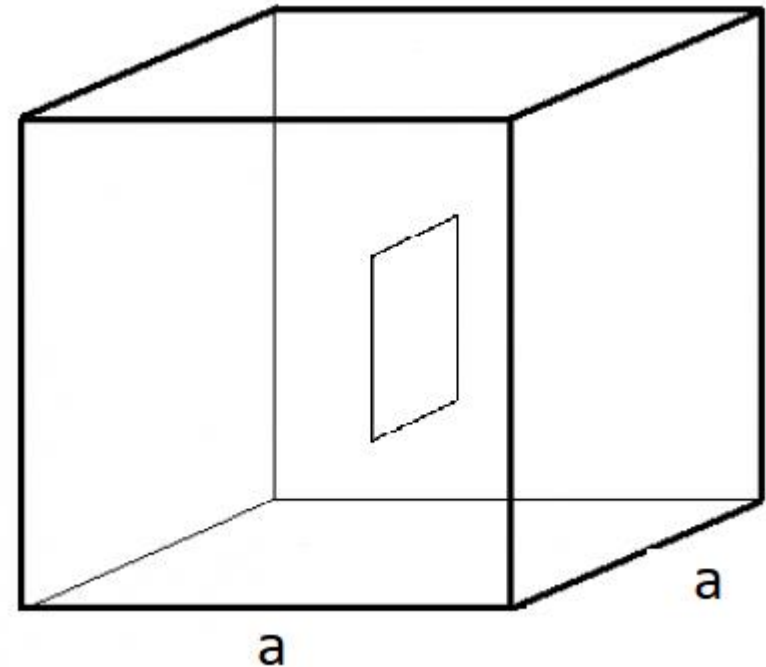
Polecenie

Proszę napisać program, który wypełni zadeklarowaną w nim 3 wymiarową tablicę liczb int o wymiarach 50x50x50 w opisany dalej sposób. Elementy tablicy mają być wypełnione wartościami 0 albo 1 tak że wartości 1 tworzą wewnątrz tablicy kwadrat o boku b i grubości 1, przy czym a musi być podany jest z klawiatury. Kwadrat powinien być położony wzdłuż osi X.

Oczekiwany wynik

- Zawartość tablicy będzie analogiczna do rysunku obok

Uwagi



J.6 Tablice 3 wymiarowe

Dane wejściowe

- r –promień kuli wczytywany z klawiatury

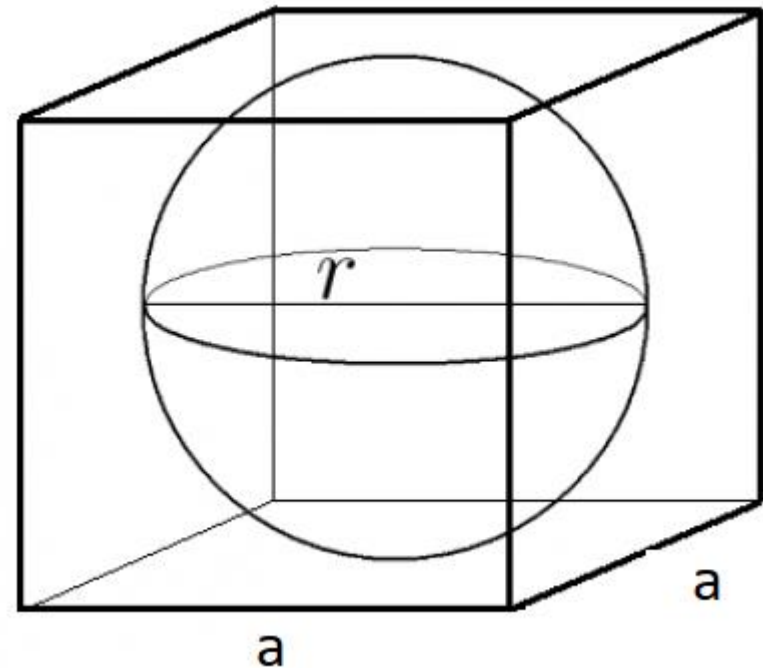
Polecenie

Proszę napisać program, który wypełni zadeklarowaną w nim 3 wymiarową tablicę liczb int o wymiarach 50x50x50 w opisany dalej sposób. Elementy tablicy mają być wypełnione wartościami 0 albo 1 tak że wartości 1 tworzą wewnątrz tablicy kulę o promieniu r , przy czym r musi być podany jest z klawiatury. Kwadrat powinien być położony wzdłuż osi X.

Oczekiwany wynik

Zawartość tablicy będzie analogiczna do rysunku obok

Uwagi



J.7 Operacje na wektorach

Dane wejściowe

- a,b wektory liczb zmiennoprzecinkowych , wczytywane z klawiatury lub z pliku

Polecenie

Proszę napisać program który będzie kalkulatorem wektorowym i będzie realizował operacje na dwu wektorach a, b

- | | |
|-------------------------|-------------|
| • Dodawanie | $a = a+b$ |
| • Odejmowanie | $a = a-b$ |
| • Mnożenie przez liczbę | $a = r * a$ |
| • Mnożenie skalarne | $a = a * b$ |

Oczekiwany wynik

- Program działa tak jak prosty kalkulator 4 działaniowy , wyświetla wynik ostatniego działania i oczekuje na wprowadzenie nowego działania i operandu by wykonać działanie i znowu przejść do stanu oczekiwania

Uwagi

J.7 Operacje na macierzach

Dane wejściowe

- a,b macierze liczb zmiennoprzecinkowych , wczytywane z klawiatury lub z pliku

Polecenie

Proszę napisać program, który będzie prostym kalkulatorem macierzowym i będzie realizował operacje na dwu macierzach a, b

- | | |
|--------------------------|-----------------|
| • Dodawanie | $a = a + b$ |
| • Odejmowanie | $a = a - b$ |
| • Mnożenie przez liczbę | $a = r \cdot a$ |
| • Mnożenie przez macierz | $a = a \circ b$ |

Oczekiwany wynik

- Program działa tak jak prosty kalkulator 4 działaniowy , wyświetla wynik ostatniego działania i oczekuje na wprowadzenie nowego działania i operandu by wykonać działanie i przejść do stanu oczekiwania

Uwagi