

2.

```
from sys import stdin

if __name__ == '__main__':
    file = ''
    line_idx = 0
    char_idx = 0
    start = 0
    last_temp = ''

    for line in stdin:
        temp = line.strip()
        temp = temp + '\n' if temp else line

        if line_idx > 0 and line_idx < 10 and not start:
            if temp == last_temp == '\n':
                start = char_idx

        file += temp
        last_temp = temp

        line_idx += 1
        char_idx += len(temp)

    while start < len(file) and file[start] == '\n': start += 1
    stop = file.find('-----')
    print(file[start:stop])
```

```
sto 35_cau_02.py 42.py
zaczeka: /root/.python-3.7/3.7.2-64bit/bin/python3 22.py
Pewna dobra kobieta bardzo pragnela miec malenkie dziecko, ale nie wiedziala, skad by je wziac. Poszla wloc do czarownicy i rzekla:
- Tak bym chciala miec malutkie dziecko. Powiedz mi, co tu robic, zebym je miala?
- O! To nieludno - odpowiedziala czarownica - moze ci doskonale poradzic. Masz tu ziarnko jecznienia - ale to nie jest takie zwyczajne ziarnko, ktore staja w polu albo sypla kuron na pokarm; zasadz je st
arannie w doniczce do kwiatow, a zobaczysz, co z tego bedzie.
- Dziekuje - rzekla pocziwa kobieta i zaplacila iedz dziesiec groszy. bo tyle to ziarnko kosztowalo.
```

```
import sys

def count_paragraphs() -> int:
    line_with_chars = False
    empty_line = False
    count = 0
    try:
        for line in sys.stdin:
            if all_white_chars(line):
                empty_line = True
                if line_with_chars and empty_line:
                    count+=1
                line_with_chars = False
            else:
                line_with_chars = True
                empty_line = False
    except Exception as e:
        print(f"Błąd podczas przetwarzania wejścia: {e}")
    if line_with_chars:
        return count+1
    else:
        return count

def all_white_chars(line):
    for char in line:
        if char!=' ' and char!='\n' and char!='\t':
            return False
    return True

if __name__ == '__main__':
    print(count_paragraphs())
```

3.a

```
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$ cat txt/calineczka.txt | python3 z3/count_paragraphs.py
138
```

3.b

```
1 import sys
2
3 def count_non_whitespace_chars() -> int:
4     count = 0
5     try:
6         while True:
7             char = sys.stdin.read(1)
8             if not char:
9                 break
10            if not white_char(char):
11                count+=1
12    except Exception as e:
13        print(f"Błąd podczas przetwarzania wejścia: {e}")
14    return count
15
16 def not_white_char(char):
17     if char!=' ' and char!='\n' and char!='\t':
18         return True
19     return False
20
21 if __name__ == '__main__':
22     print(count_non_whitespace_chars())
```

```
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$ cat txt/calineczka.txt | python3 z3/count_non_whitespace.py
20052
```

3.c

```
1 import sys
2
3 def percentage_sentences_with_proper_names():
4     sentences = 0
5     sentences_with_proper_names = 0
6     try:
7         for line in sys.stdin:
8             sentence_started = False
9             sentence_has_proper_name = False
10            for char in line:
11                if char.isupper() and sentence_started:
12                    sentence_has_proper_name = True
13
14                if char == '.' or char == '!' or char == '?':
15                    if sentence_started:
16                        sentences += 1
17                        if sentence_has_proper_name:
18                            sentences_with_proper_names += 1
19                        sentence_started = False
20                        sentence_has_proper_name = False
21                elif char.isupper() and not sentence_started:
22                    sentence_started = True
23    except Exception as e:
24        print(f"Błąd podczas przetwarzania wejścia: {e}")
25    if sentences == 0:
26        raise Exception("Brak zdań w tekście")
27    return sentences_with_proper_names/sentences *100.0
28
29 if __name__ == '__main__':
30     try:
31         print(percentage_sentences_with_proper_names())
32     except Exception as e:
33         print(f"Błąd podczas przetwarzania wejścia: {e}")
```

```
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$ cat txt/calineczka.txt | python3 z3/percentage_sentences_with_proper_names.py
12.937062937062937
```

3.d

```
from sys import stdin

def filter(check_function) -> str:
    sentence = ''
    result = ''
    for line in stdin:
        sentence = ''
        for i,v in enumerate(line):
            if v == '.' or v == '!' or v == '?' or v == '...':
                if len(sentence) == 0:
                    sentence = ''
                    continue

                if i+1 < len(line):
                    if line[i+1] == '!' or line[i+1] == '?':
                        sentence += v
                        continue

                ### wykrywanie dialogu
                temp_sentence = sentence.strip()
                hyphen = 2 if temp_sentence[:2] == '- ' else 0
                temp_sentence = temp_sentence[hyphen:]
                temp_sentence += v
                ###

                ### uznajemy ze fraza z jednym slowem nie moze byc zdaniem
                # print( [temp_sentence], hyphen)
                word_count = temp_sentence.count(' ') + 1
                if word_count >= 2:
                    result += f'{temp_sentence}\n' * check_function(temp_sentence)

                sentence = ''
            else:
                sentence += v

    return result
```

```
longest_sentence.py > ...
from lib import filter

ANS = ''
def get_longer(new):
    global ANS

    if len(new) > len(ANS):
```

```
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/longest_sentence.py
Pani Gratchitt zaprzęta przy koniku zawczasu przygotowaną polewą do pieczystego; pan Piotr przysmażał kartofle z niepodobnym do opisanie zapachem; panna Belinda studiła kompot z jabłek; Marta ścierała prochy z ograniczonych talerzy; Bob wziął na ręce Matego Tima i posadził go przy stole obok siebie w najwygodniejszym miejscu; dwójce małych Gratchittów przystawiało krzesła do stołu, przy czym żadne z nich nie zapomniało o sobie i zajmwszy przeznaczone im miejsca, powsadzały w buzie łyżki, aby nie napierać się gęsi, zanim przyjdzie na nie kolej.
```

```
def check(sentence):
    get_longer(sentence)
    return True

if __name__ == '__main__':
    filter(check)
    print(ANS)
```

3e.

```
import sys

def longest_sentence_diff_char():
    longest_sentence_length = 0
    longest_sentence = ''
    try:
        for line in sys.stdin:
            sentence = ''
            sentence_size = 0
            sentence_started = False
            last_char = ''
            can_be_longest = True
            for char in line:
                if sentence_started:
                    if sentence[-1] == ' ':
                        if last_char.lower() == char.lower():
                            can_be_longest = False
                        if char != ' ':
                            last_char = char
                    sentence_size += 1
                    sentence += char

                if char == '.' or char == '!' or char == '?':
                    if sentence_started:
                        if sentence_size > longest_sentence_length and can_be_longest:
                            longest_sentence_length = sentence_size
                            longest_sentence = sentence
                        sentence_started = False
                        sentence = ''
                        sentence_size = 0
                        last_char = ''
                        can_be_longest = True
                    elif char.isupper() and not sentence_started:
                        sentence_started = True
                        sentence_size = 1
                        sentence += char
                        last_char = char
            except Exception as e:
                print(f"Błąd podczas przetwarzania wejścia: {e}")
                return longest_sentence

if __name__ == '__main__':
    print(longest_sentence_diff_char())
```

```
kac@kla:~/studia/python_syf/języki-skryptowe/12$ cat txt/opowiesc_wigilijna.txt | python3 z3/longest_sentence_diff_char.py
Tu można było zobaczyć przygotowania do Włoczek, bogato nakryte stoły, całe stoły talerzy ogrzanych na gazie, kosztowne opony i meble; tam dzieci wbiegały na pokrytą śniegiem ulicę, aby powitać jak na jarmarku przybywające zamek stolicy i braci, kuzynów, wujów, ciotki; owdzie na oknach przesuwali się cienie zgromadzonych gości; tam znów dziewczęta, przybrane w futrzane kołnierze i także bućki, szczerząc się jedna przez drugą, lekko krokem spieszyły w sąsiedztwo.
```

3f.

```
import sys

def first_sentence_with_subordinate_clauses(min_number_of_commas=3):
    try:
        for line in sys.stdin:
            sentence = ''
            sentence_started = False
            number_of_commas = 0
            for char in line:
                if sentence_started:
                    if char == ',':
                        number_of_commas += 1
                    sentence += char

                if char == '.' or char == '!' or char == '?':
                    if sentence_started:
                        if number_of_commas >= min_number_of_commas:
                            return sentence
                        sentence_started = False
                        sentence = ''
                        number_of_commas = 0
                    elif char.isupper() and not sentence_started:
                        sentence_started = True
                        sentence += char
            except Exception as e:
                print(f"Błąd podczas przetwarzania wejścia: {e}")
                return ''

if __name__ == '__main__':
    print(first_sentence_with_subordinate_clauses())
```

```
kac@kls:~/studia/python_syf/języki-skryptowe/1$ cat txt/opowiesc_wigilijna.txt | python3 z3/first_sentence_with_subordinate_clauses.py
Scrooge przy tym był jedynym egzekutorem, jedynym spadkobiercą, jedynym administratorem, jedynym przyjacielem Marleya i jedynym człowiekiem, który szedł za jego trumną.
```

3g.

```
from lib import filter

def check(sentence):
    word_count = sentence.count(' ') + 1
    return word_count >= 2 and word_count <= 4

if __name__ == '__main__':
    result = filter(check)
    print(result, end='')
```

```
Tu mała uwaga.
Oczywiście, że wiedział.
Jakżeby mogło być inaczej?
On tego właśnie pragnął.
Wesołych świąt, wuju!
zawołał młody, sympatyczny głos.
Cóż znowu za głupstwa?!
mruknął Scrooge.
Święta nazywasz głupstwem, wuju?
zawołał siostrzeniec.
I mam zupełną słuszność.
Wesołych świąt!
Ty, biedak?
Ty, bogacz?
rzekł siostrzeniec.
Wesołych świąt!
Znać ich nie chcę!
mitygował go łagodnie siostrzeniec.
przetrwał wuj surowo.
powtórzył siostrzeniec.
sarknął Scrooge.
zgromił go ostro Scrooge.
Nie gniewaj się, wuju!
A dlaczego ożeniłeś się?
```

3i.

```
First_20.py > ...
from lib import filter

def count(c=[0]):
    # korzystamy z faktu, że listy są mutowalne
    # i osiągamy "statyczną" zmienną
    c[0] += 1
    return c[0]

def check(_):
    return count() <= 20

if __name__ == '__main__':
    result = filter(check)
    print(result)
```

```
kac@kla:~/studia/python_syf/języki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/first_20.py
```

Zaczął się od tego, że Marley umarł.  
 To żadnej nie ulega wątpliwości.  
 Protokół pogrzebu podpisali: ksiądz, urzędnik, grabarz i właściciel zakładu pogrzebowego.  
 Scrooge także go podpisał, a nazwisko Scrooge'a miało poważny walor na giełdzie zawsze, cokolwiek by podpisał.  
 Stary Marley był więc martwy jak ćwiek we drzwiach.  
 Tu mała uwaga.  
 Nie chcę wcale przez to powiedzieć, że wiem z własnego doświadczenia, jakoby na świecie nie istniało nic bardziej martwego nad ćwiek w drzwiach.  
 Co do mego osobistego poglądu, sądziłbym raczej, że ćwiek w trumnie jest najbardziej martwym artykułem w handlu żelaznym.  
 Lecz mądrość naszych przodków kryje się w tym porównaniu i moje niepoświęcone ręce nie będą zmieniały tradycji narodowej, bo kraj mógłby przez to ucierpieć.  
 Niech mi więc będzie wolno powtórzyć z naciskiem, że Marley był martwy jak ćwiek w drzwiach.  
 Czy Scrooge wiedział o jego śmierci?  
 Oczywiście, że wiedział.  
 Jakżeby mogło być inaczej?  
 Scrooge i on byli współnikami, nie wiem przez ile lat, w każdym razie bardzo długo.  
 Scrooge przy tym był jedynym egzekutorem, jedynym spadkobiercą, jedynym administratorem, jedynym przyjacielem Marleya i jedynym człowiekiem, który siedział za

3.j

```
from lib import filter

def check(sentence):
    ans = min(1, sentence.count(' ale ')) + \
          min(1, sentence.count(' oraz ')) + \
          min(1, sentence.count(' że ')) + \
          min(1, sentence.count(' lub ')) + \
          min(1, sentence.count(' i '))

    return ans >=2

if __name__ == '__main__':
    print(filter(check))
```

```
kac@kla:~/studia/python_syf/języki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/spojniki.py
```

każdy bowiem to przyzna, że gdybyśmy nie byli najmocniej przekonani, iż ojciec Hamleta umarł, zanim przedstawił tragedię, to jego błędzenie nocą, podczas wschodniego wiatru i to po swoich własnych watach fortecznych, w celu wstrząśnięcia umysłu swego syna, umysłu i bez tego słabego, nie miałoby w sobie nic bardziej osobliwego, jak spacer pierwszego lepszego podstarzałego jegomościa o zmierzchu i wieczornym, na przykład po cmentarzu świętego Pawła.  
 Niepogoda nie wiedziała, z której strony zaatakować go, największy deszcz, śnieg, grad albo zawierucha miały nad nim wyższość pod jednym tylko względem: one często okazywały się szczodre, spadały na świat i ludzi hojnie, Scrooge zaś nie rozumiał wcale pojęć szczodrości lub hojności.  
 Mgła wdierała się do mieszkań przez każdą szparę i dziurkę od klucza, a tak była gęsta, że domy na przeciwnej stronie wąskiej ulicy trudno było dojrzeć.  
 Ziębnięty pracownik nie mógł jednak podsycać ognia, ponieważ Scrooge trzymał skrzynię z węglami w swoim pokoju, a ile razy pomocnik wszedł z łopatką, pryncypał zawsze mu groził, że będą musieli się rozstać, wskutek czego błędny pomocnik obwinął sztygę białym szalikiem i usiłował ogrzać się przy świecy; ponieważ atoli nie posiadał dość bujnej inaginacji, usiłowania były bezskuteczne.  
 Był to głos siostrzeńca Scrooge'a, który wszedł tak nagle i szybko, że Scrooge wtedy dopiero zauważył jego obecność, kiedy siostrzeniec do niego przewodził; inaczej byłby go z pewnością nie wpuścił.  
 Siostrzeniec Scrooge'a tak się rozgrzał szybko chodząc wśród mgły i mrozu, że cały był rozpromieniony; piękna twarz jego jaśniała, oczy błyszczały, a płuca wyrzucały chmury pary.  
 W ten dzień przeprowadzasz bilans twoich ksiąg handlowych i przekonujesz się, że po dwunastu miesiącach galerniczej pracy nie masz ani odrobiny zysku.  
 Mimo to zawsze myślę o nich nie tylko jako o świętej i czcigodnej pamiętce dla całego chrześcijańskiego świata, ale także jako o chwilach bardzo upragnionych.  
 Dlatego, wuju, choć święta ta nigdy nie przyczyniła mi w kieszeni ani ziarenka srebra lub złota, wierzę, że przyniosła mi niejedno dobro i że przyniosła mi jeszcze wiele radości.  
 Radość zdanie i to, co pan na wstępie powiedziałeś, przyprowadzi mnie o obawę, że coś wstrzymało ich użyteczną działalność.  
 Ponieważ wiadomo powszechnie, że instytucje wspomniane, pomimo ustnych zabiegów, nie są w stanie po chrześcijańsku zaspokoić potrzeb ani duchowych, ani cielesnych wielkich mas nędzarzy, przychodzący im z pomocą, zbierając ofiary na dostarczenie choć najbiedniejszym jadła, napojów i środków do ogrzania złodowaczących ich mieszkań.  
 Nauczni, widząc, że nic tu nie wskorają, wzruszyli tylko ramionami i wyszli.  
 Tymczasem mgła i ciemność tak zgęstniały, że na ulicach zjawili się ludzie i biegali z pozapalonymi latarniami, ofiarując się iść przed konie ciągniętymi powozy i prowadzić je po właściwej drodze.  
 Starożytna wieża kościoła, z której gotyckiego okna ciekawie spoglądał wciąż na Scrooge'a stary rozbity dzwon, zniknęła i ów dzwon wybił godzinę oraz kwadrans wśród mgły z tak silnymi wibracjami, jak gdyby mu żęby szarpały z zima.  
 Kłady drobitu i składowe porządkowały tak imponujące widokiska, że zdawało się, iż pozioma mgła kupa - sprządały i targu nie może nieć z nimi nic wspólnego.

3h

```
from lib import filter

def check(sentence):
    return sentence[-1] == '!' or sentence[-1] == '?'

if __name__ == '__main__':
    print(filter(check))
```



```
kac@kla:~/studia/python_syf/języki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/question_or_exclamation.py
Czy Scrooge wiedział o jego śmierci?
Jakżeby mogło być inaczej?
Nikt nigdy nie zatrzymał go na ulicy z uśmiechem przyjaznym, aby powiedzieć: „Mój kochany Scrooge, jak się masz?
Kiedy też raczysz mnie odwiedzić?
Brak oka jest lepszy od złego oka, mój niewidomy panie!
Ależ co to obchodziło Scrooge'a?
Wesołych świąt, wuju!
Niech cię Bóg ma w swej opiece!
Cóż znowu za głupstwa?!
Święta nazywasz głupstwem, wuju?
Jestem pewny, że nie myślisz tego!
Wesołych świąt!
Jakież ty masz powód być wesołym?
Jakież ty możesz mieć powód do wesołości?
Ory, biedak?
A jakież ty masz powód być ponurym, wuju?
Jakież ty możesz mieć rację do smutku?
Ory, bogacz?
Nie bądź taki kwaśny, wuju!
```

3k.

```
from lib import filter

TAB = list()
def get_lengths(sentence):
    global TAB

    TAB.append(sentence)
    return True

if __name__ == '__main__':
    filter(get_lengths)

    lengths = sorted([len(i) for i in TAB])
    for i in TAB:
        if len(i) >= lengths[len(lengths)* 3 // 4]:
            print(i)
```

```
kac@kla:~/studia/python_syf/języki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/4th_quartile.py
Nie chcę wcale przez to powiedzieć, że wiem z własnego doświadczenia, jakoby na świecie nie istniało nic bardziej martwego nad dźwięk w drzwiach.
Co do mego osobistego poglądu, sądziłbym raczej, że dźwięk w trumnie jest najbardziej martwym artykułem w handlu żelaznym.
Lecz mądrość naszych przodków kryje się w tym porównaniu i moje niepoświęcone ręce nie będą zniechęcały tradycji narodowej, bo kraj mógłby przez to ucierpieć.
Scrooge przy tym był jedynym egzektorem, jedynym spadkobiercą, jedynym administratorem, jedynym przyjacielem Marleya i jedynym człowiekiem, który siedzi za jego trumną.
A nawet i Scrooge nie był znowu tak bardzo dotknięty tym wypadkiem, bo okazał się doskonałym biznesmenem w dniu pogrzebu i uczcił go korzystnym interesem.
Każdy bowiem to przyzna, że gdybyśmy nie byli najmocniej przekonani, iż ojciec Hamleta umarł, zanim przedstawienie tragedii się zaczyna, to jego błądzenie nocą, podczas wschodniego wiatru i to po swoich własnych watach fortecznych, w celu wstrząśnięcia umysłu swego syna, umysłu i bez tego słabego, nie miałoby w sobie nic bardziej osobliwego, jak spacer pierwszego lepszego podstarzałego jegomościa o zmierzchu w wieczornym, na przykład po cmentarzu świętego Pawła.
Czasami nowicjusze w interesach handlowych nazywali Scrooge'a Scrooge, czasami Marley, on zaś odpowiadał na obydwie nazwiska.
Chwytała ona, ścisnęła, dusiła, ze skóry oddzierając swoją ofiarę i nie wypuszczała jej, aż po doszczętnym wyzyskaniu.
Twardy i ostry jak krzenie, z którego
Zimno wewnętrzne mroziło jego starcze
głosie.
Nosili ustawicznie ze sobą swą lodowatą
Nie było wiatru bardziej niż on gryzący
Niepogoda nie wiedziała, z której strony
i ludzi hojnie, Scrooge zaś nie rozumiał
" - Żebracy nie prosili go o jałmużnę.
Iść samotnie przez życie, odpychać od
amotny jak ostrzyga.
aczerwieniało mu oczy, zabarwiało na niebiesko usta i brzmiało gniewnie w jego ostrym
en stopień nawet podczas wigilii Bożego Narodzenia.
mniej dostępnego błaganiem i prośbom.
ższość pod jednym tylko względem: one często okazywały się szczodre, spadały na świat
tał Scrooge'a o drogę.
ej drogi się trzymał niewzruszenie.
```

3l.

```
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/in_order.py
mruknął Scrooge.
Wesołych świąt!
rzekł siostrzeniec.
Wesołych świąt!
powtórzył siostrzeniec.
sarknął Scrooge.
Dlaczego odmawiasz, wuju?
A dlaczego ożeniłeś się?
Bądź zdrów!
powtórzył Scrooge.
Bądź zdrów!
mruknął Scrooge.
Bądź zdrów!
mruczał Scrooge.
Co to znaczy?
Bardzo wiele!
odparło widmo.
Czy możesz
```

4. Wszystkie skrypty są wywoływane przy pomocy konstrukcji `if __name__ == '__main__':`.

5.

```
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/in_order.py | python3 z3/question_or_exclamation.py
Wesołych świąt!
Wesołych świąt!
Dlaczego odmawiasz, wuju?
A dlaczego ożeniłeś się?
Bądź zdrów!
Bądź zdrów!
Bądź zdrów!
Co to znaczy?
Bardzo wiele!
Głupstwo, powtarzam!
Bardzo dawnej przeszłości?
Czy ją poznaję?!
Biedny chłopiec!
A to Walentyn!
Czy go nie widzisz?
Dobrze mu tak!
Biedny, biedny chłopiec!
Czy go poznaję?
Hej tam!
Biedny Dick!
Jakaż tu zasługa?
Czy nieprawda?
Kogóż to?
Cóż znowu!
Pana Scrooge'a?
```

```
broken pipe: [errno 32] broken pipe
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/in_order.py | python3 z3/longest_sentence.py
Niech pan pozwoli, zaprowadzę.
```

```
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$ cat txt/opowiesc_wigilijna.txt | python3 z3/in_order.py | python3 z3/longest_sentence_diff_char.py
Dlatego musimy się śpieszyć.
kac@kla:~/studia/python_syf/jezyki-skryptowe/l2$
```



