Defenzivno programiranje:

- uporabnik lahko da malicious input
- lahko gre kaj narobe s filesystemom, kakšen file vmes izgine (sprememba na datotečnem sistemu)
- lahko se vmes unmounta disk, ga nekdo ven potegne
- lahko vmes izgubiš network access (kabel se prereže, switch crkne) - včasih so problemi bolj na hardware nivoju

def_0:

- pot gre lahko kamorkoli - če napišeš relativno pot s `../`, lahko prideš tudi v direktorije, ki so na deny seznamu:
  - popravi na os.path.realpath()
  - imamo tudi TOC/TOU problem, ampak bomo to spustili sedaj
- če folder ne obstaja, samo izpišemo napako in gremo normalno naprej, namesto da bi končali:
  - exit(1)
- pri `open` ne preverimo, če se je file uspešno odprl - kaj če nimamo write pravic:
- eval lahko izvede karkoli:
  - namesto `eval` damo `int` in preverimo, če smo res dobili številko
- na koncu se lahko zgodi, da je `lines` enako 0 in imamo zero division
- datoteke nikoli ne zapremo:
  - tu sicer ni problem, ker itak želimo zapreti datoteko samo na koncu programa
  - pri kakšen web serverju bi bilo to pomembno, da datoteke ne ostanejo za vedno odprte
- ne preverimo, da je URL valid:
  - ne smemo catchati vseh exceptionov, ker bo to catchalo tudi 404
- med `start` in `end` časom se lahko ura premakne in bomo dobili napačno meritev

01.c:

- pri malloc bi rabili še sizeof in castati v char pointer
- scanf ne preverja dolžine inputa
- ne preverimo, če se je file odprl:
  - dobro je povedati še errno, če je bila napaka pri odpiranju datoteke
- fscanf lahko ne dobi številke
- malloc lahko faila

- ko vpisujemo `entry`, damo lahko preveliko cifro in gre out of bounds
- `10*1024*1024*1024` overflowa integer - uporabi long oz. `size_t`
- `argv[1]` ni nujno, da obstaja:
  - argc mora biti enak 2, ker je prvi argument ime programa
- fscanf mora prebrati točno toliko številk, kot nas zanima (1)
- `sum` lahko overflowa - zares bi rabili neskončno velika števila, kot pri pythonu
- tudi pri vsakem printf bi rabili preveriti, če se je uspešno končal
- da ne rabimo vedno pisati vseh free, jih lahko samo spodaj napišemo in uporabimo `goto`, čeprav naj tega ne bi delali