

ZADANIE 1:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]  
$ touch lab03_plik.txt
```

W tym pliku umieścę kilka zdań na temat algorytmu Hornera

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]  
$ base64 lab03_plik.txt > lab03_plik_zaszyfrowany.txt
```

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]  
$ cat lab03_plik.txt
```

Algorytm Hornera (Horner's scheme, Horner's method) to sposób obliczania wartości wielomianu dla danej wartości argumentu wykorzystujący minimalną liczbę mnożeń. Jest to również algorytm dzielenia wielomianu $W(x)$ przez dwumian $(x-c)$. Schemat ten wiązany jest z nazwiskiem Hornera, był jednak już znany Newtonowi, Ruffiniemu i matematykom chińskim w XII wieku.

Dla wielomianu stopnia n należy wykonać n mnożeń i n dodawań. Schemat Hornera jest optymalny pod względem liczby wykonywanych działań (1971 Borodin).

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]  
$ cat lab03_plik_zaszyfrowany.txt  
QWxnb3J5dG0gSG9ybmlVYSAoSG9ybmlVYjMgc2NoZW1lLCBib3JuZXIncylBtZXRob2QpIHRvIHVwYmV3PDs2IGb2JsaWN6YW5pYSB3YXJ0b8WbY2kgd2llbG9taWFudSBkbGEgYmFkZWogd2FydG/Fm2NpIGFyZ3VtZW50dSB3eWtvncp5c3R1asSFY3kgbWluaW1hbG7EhSBsaWN6YsSZIG1ub8W8ZcwELiBKZXN0IHRvIHLDs3duaWxFvCBhbGdvcnl0bSBkemllbGVuaWEgd2llbG9taWFudSBXKHgpIHBiemV6IGR3dW1pYW4gKHgtYykuIFNjaGVtYXQgdGVuIHdpXIV6YW55IGplc3QgeiBuYXp3aXNraWVtIEhvcm5lcmEsIGJ5xYIgamVkbmFrIGp1xbwgem5hbnkgTmV3dG9ub3dpLCBSdWZmaW5pZW11IGkgbWFOZW1hdHlrb20gY2hpxYRza2ltIHcgWEIJIHdpZWt1LgoKRGxhIHdpZWxvbWlhbUgc3RvcG5pYSBuIG5hbGxFvHkgd3lrb25hxcgbiBtbm/FvGXFhCBpIG4gZG9kYXdhxYQuIFNjaGVtYXQgSG9ybmlVYYSBqZXN0IG9wdHltYWxueSBwb2Qgd3pnbMSZZGVtIGxpY3pieSB3eWtvbnl3YW55Y2ggZHppYcWcYcWEICgxOTcxIEJvcmlkaW4pLgo=
```

Tak wygląda jego zaszyfrowana wersja

Dekodowanie lab03_plik_zaszyfrowany.txt:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ base64 --decode lab03_plik_zaszyfrowany.txt
Alorytm Hornera (Horner's scheme, Horner's method) to sposób obliczania wartości
wielomianu dla nadej wartości argumentu wykorzystujący minimalną liczbę mnożeń.
Jest to również algorytm dzielenia wielomianu  $W(x)$  przez dwumian  $(x-c)$ . Schemat t
en wiązany jest z nazwiskiem Hornera, był jednak już znany Newtonowi, Ruffiniemu
i matematykom chińskim w XII wieku.

Dla wielomianu stopnia  $n$  należy wykonać  $n$  mnożeń i  $n$  dodawań. Schemat Hornera jes
t optymalny pod względem liczby wykonywanych działań (1971 Borodin).
```

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ file lab03_plik.txt
lab03_plik.txt: Unicode text, UTF-8 text, with very long lines (359)

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ file lab03_plik_zaszyfrowany.txt
lab03_plik_zaszyfrowany.txt: ASCII text
```

Plik z oryginalną wiadomością jest kodowany za pomocą UTF-8

Plik zaszyfrowany jest kodowany za pomocą ASCII

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ strings lab03_plik.txt
Alorytm Hornera (Horner's scheme, Horner's method) to spos
b obliczania warto
ci wielomianu dla nadej warto
ci argumentu wykorzystuj
cy minimaln
cy liczb
mno
. Jest to r
wnie
algorytm dzielenia wielomianu  $W(x)$  przez dwumian  $(x-c)$ . Schemat ten wi
zany jest z nazwiskiem Hornera, by
jednak ju
znany Newtonowi, Ruffiniemu i matematykom chi
skim w XII wieku.
Dla wielomianu stopnia  $n$  nale
y wykona
n mno
i n dodawa
. Schemat Hornera jest optymalny pod wzgl
dem liczby wykonywanych dzia
(1971 Borodin).
```

W pliku pomijane są polskie znaki, a dodatkowo uwzględniane są tabulacje

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ strings lab03_plik_zaszyfrowany.txt
QWxbn3J5dG0gSG9ybmVyYSAoSG9ybmVyJ3Mgc2NoZW11LCBIb3JuZXIncyBtZXRob2QpIHRvIHNo
b3PDs2Igb2JsaWN6YW5pYSB3YXJ0b8WbY2kgd2llbG9taWFudSBkbGEgYmFkZWogd2FydG/Fm2Np
IGFyZ3VtZW50dSB3eWtvcnp5c3R1asSFY3kgbWluaW1hbG7EhSBsaWN6YsSZIG1ub8W8ZcWELiBK
ZXN0IHRvIHLDs3duaWxFvCBhbGdvcnl0bSBkemllbGVuaWEgd2llbG9taWFudSBKXGpIHByemV6
IGR3dW1pYW4gKHgtYykuIFNjaGVtYXQgdGVuIHdpXIV6YW55IGplc3QgeiBuYXp3aXNraWVtIEhv
cm5lcmEsIGJ5xYIgamVkbmFrIGp1xbwgem5hbnkgTmV3dG9ub3dpLCBSdWZmaW5pZW11IGkgbWF0
ZW1hdHlrb20gY2hpYXRza2ltIHcgWELJIHdpZWt1LGoKRGxhIHdpZWxvbWlhbUgc3RvcG5pYSBu
IG5hbGXFvHkgd3lrb25hXicgbiBtbm/FvGXFHCBpIG4gZG9kYXdhxYQuIFNjaGVtYXQgSG9ybmVy
YSBqZXN0IG9wdHltYWxueSBwb2Qgd3pnbMSZGvtIGxpY3pieSB3eWtvcnl3YW55Y2ggZHppYcWC
YcWEICgxOTcxIEJvcn9kaW4pLgo=
```

W pliku zaszyfrowanym nie ma takiego problemu

ZADANIE 2:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ sudo fdisk -l
Device Boot Start End Sectors Size Id Type
/dev/sda1 778135908 1919645538 1141509631 544.3G 72 unknown
/dev/sda2 168689522 2104717761 1936028240 923.2G 65 Novell Netwa
/dev/sda3 1869881465 3805909656 1936028192 923.2G 79 unknown
/dev/sda4 2885681152 2885736650 55499 27.1M d unknown

Partition table entries are not in disk order.

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ sudo mount /dev/sda ~/InformatykaSledcza/InfSl_LAB03

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ cd InformatykaSledcza/InfSl_LAB03

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ ls
AGH USB_4GB_Kingston.E01
File.zip Conclusion reszta
```

następnie przekopiowałem plik File.zip i wymontowałem swój pendrive

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ ls
File.zip lab03_plik.txt lab03_plik_zaszyfrowany.txt
```

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ unzip File.zip
Archive:  File.zip
  inflating: D19910350Lj.pdf
  inflating: D2020000211201.pdf
  extracting: Text
```

File:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ ls
D19910350Lj.pdf  D2020000211201.pdf  File.zip  Text  lab03_plik.txt  lab03_plik_zaszyfrowany.txt

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ file File.zip
File.zip: Zip archive data, at least v2.0 to extract, compression method=deflate

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ file D19910350Lj.pdf
D19910350Lj.pdf: PDF document, version 1.5, 351 pages

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ file D2020000211201.pdf
D2020000211201.pdf: PDF document, version 1.5, 18 pages

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ file Text
Text: ASCII text, with no line terminators

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$
```

Pdftinfo:

1. D19910350Lj.pdf:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ pdftinfo D19910350Lj.pdf
Title:                Akt prawny
Author:               Władysław Baksza
Creator:              Microsoft® Word 2013
Producer:             Microsoft® Word 2013
CreationDate:         Tue Oct 12 13:08:08 2021 CEST
ModDate:              Tue Oct 12 13:08:08 2021 CEST
Custom Metadata:      no
Metadata Stream:      no
Tagged:               yes
UserProperties:        no
Suspects:              no
Form:                 none
JavaScript:            no
Pages:                351
Encrypted:             no
Page size:            595.32 x 841.92 pts (A4)
Page rot:              0
File size:            2601654 bytes
Optimized:            no
PDF version:          1.5
```

Odpowiadając na pytanie:

Tytuł dokumentu: "Akt prawny"

Data wytworzenia pliku: Tue Oct 12 13:08:08 2021

Liczba stron: 351

Wielkość stron: 595.32 x 841.92 pts (A4)

Czy plik zawiera JavaScript: nie

Autor: Władysław Baksza

Oprogramowanie użyte: Microsoft® Word 2013

2. D2020000211201.pdf:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ pdftinfo D2020000211201.pdf
Title:      Ustawa z dnia 28 października 2020 r. o zmianie niektórych ustaw w związku z przeciwdzia-
aniem sytuacjom kryzysowym związanym z wystąpieniem COVID-19
Author:     RCL
Creator:    Microsoft® Word 2010
Producer:   Microsoft® Word 2010; modified using iTXt 2.1.7 by 1T3XT
CreationDate: Sat Nov 28 18:39:52 2020 CET
ModDate:    Sat Nov 28 18:40:01 2020 CET
Custom Metadata: no
Metadata Stream: yes
Tagged:     yes
UserProperties: no
Suspects:   no
Form:       AcroForm
JavaScript:  no
Pages:      18
Encrypted:   no
Page size:  595.32 x 841.92 pts (A4)
Page rot:   0
File size:  407654 bytes
Optimized:  no
PDF version: 1.5
```

Odpowiadając na pytanie:

Tytuł dokumentu: "Ustawa z dnia 28 października 2020 r. o zmianie niektórych ustaw w związku z przeciwdziałaniem sytuacjom kryzysowym związanym z wystąpieniem COVID-19"

Data wytworzenia pliku: Sat Nov 28 18:39:52 2020

Liczba stron: 18

Wielkość stron: 595.32 x 841.92 pts (A4)

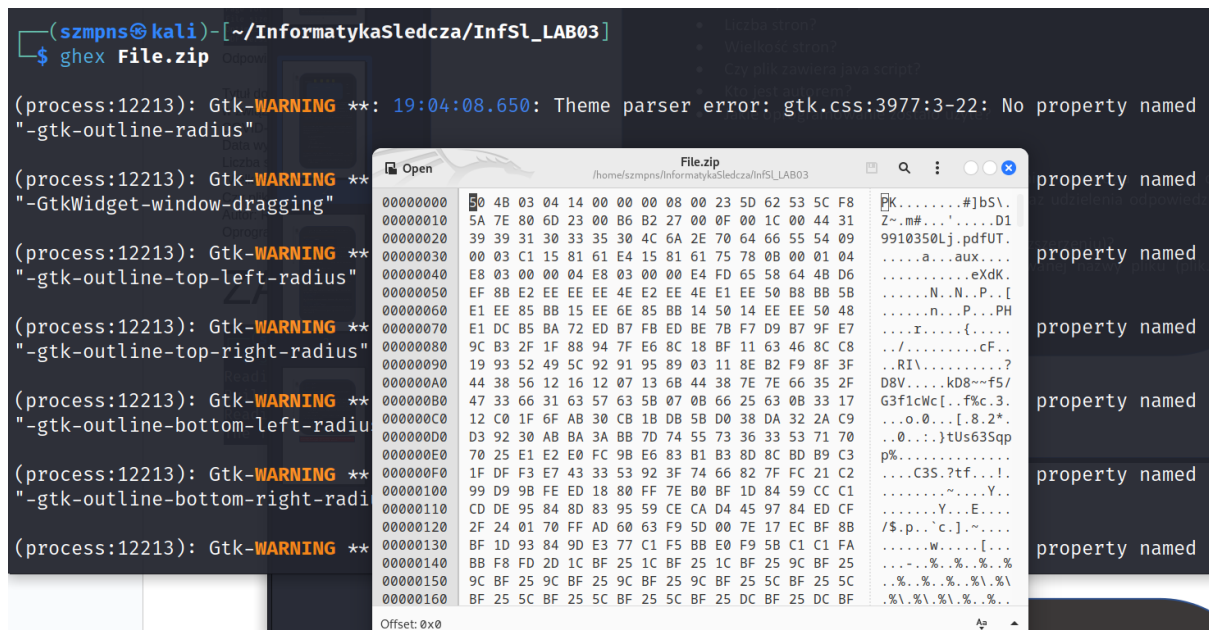
Czy plik zawiera JavaScript: nie

Autor: RCL

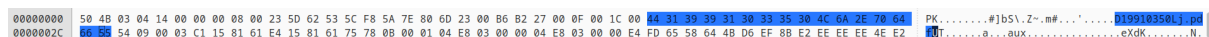
Oprogramowanie użyte: Microsoft® Word 2010

ZADANIE 3:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ sudo apt-get install ghex
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and
```

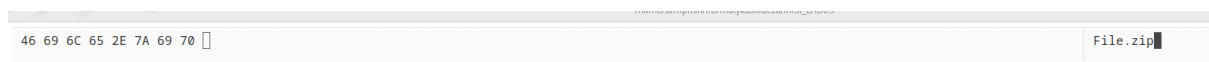
Widoczna nazwa pliku zawartego w archiwum:



Pytanie 1:

Informacja dotycząca rozszerzenia pliku znajduje się w pierwszym wierszu każdego skompresowanego pliku.

Pytanie 2:

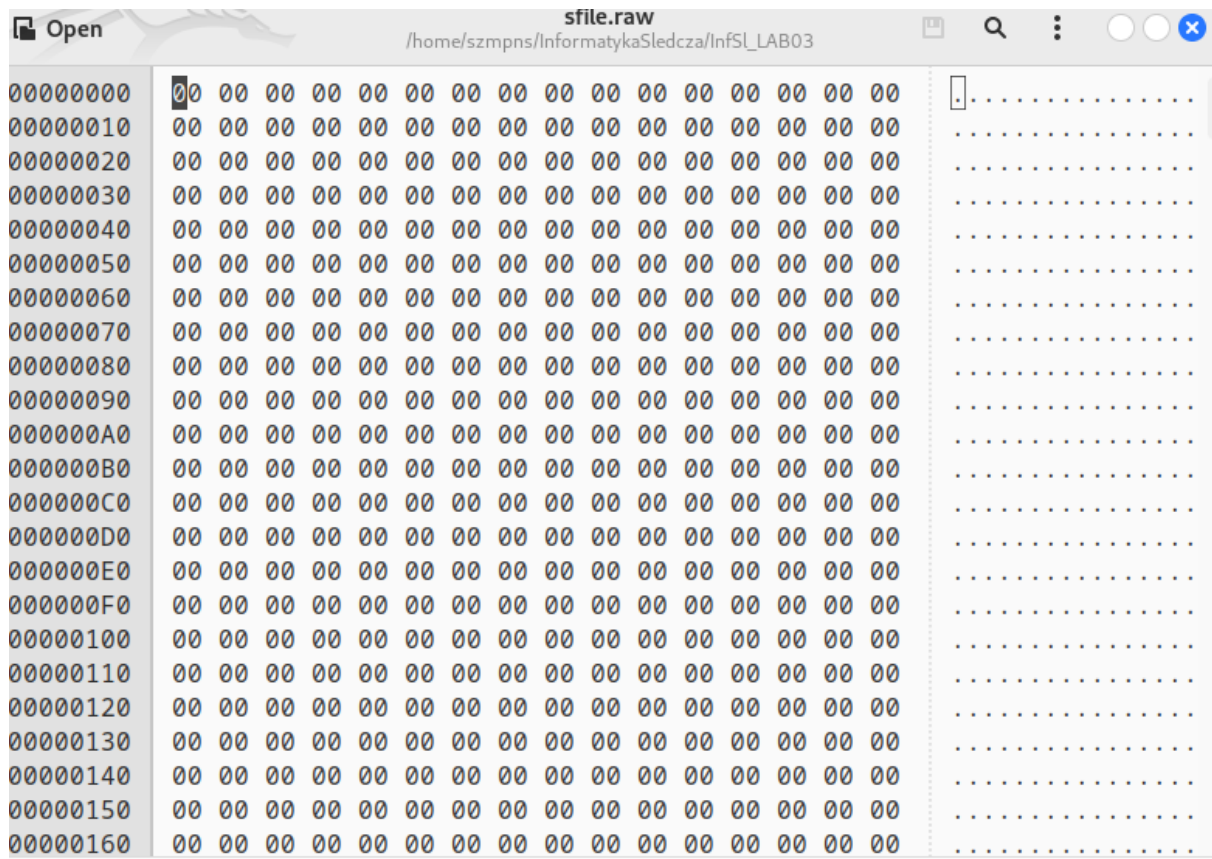


Pytanie 3:

Zawartość nie będzie czytelna, całość jest skompresowana, co oznacza, że została zastąpiona znakami, które pozwalają odczytać jej nieskompresowaną wartość, natomiast nie są szczególnie czytelne dla użytkownika. Jeżeli plik jest wystarczająco mały, aby jego kompresja (działanie mające na celu zmniejszenie "wagi" pliku), użytkownik będzie zdolny odczytać zawartość.

ZADANIE 4:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSL_LAB03]  
$ dd if=/dev/zero of=sfile.raw count=100 bs=1M  
100+0 records in  
100+0 records out  
104857600 bytes (105 MB, 100 MiB) copied, 0.0452697 s, 2.3 GB/s
```



Plik nie zawiera żadnych danych


```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ mkfs.fat sfile.raw
mkfs.fat 4.2 (2021-01-31)
```

```
Open sfile.raw /home/szmpns/InformatykaSledcza/InfSl_LAB03
00000000 EB 3C 90 6D 6B 66 73 2E 66 61 74 00 02 04 04 00 02 00 02 00 00 F8 C8 00
00000018 20 00 08 00 00 00 00 00 00 00 20 03 00 80 00 29 62 5C D0 FC 4E 4F 20 4E 41
00000030 4D 45 20 20 20 20 46 41 54 31 36 20 20 20 0E 1F BE 5B 7C AC 22 C0 74 0B
00000048 56 B4 0E BB 07 00 CD 10 5E EB F0 32 E4 CD 16 CD 19 EB FE 54 68 69 73 20
00000060 69 73 20 6E 6F 74 20 61 20 62 6F 6F 74 61 62 6C 65 20 64 69 73 6B 2E 20
00000078 20 50 6C 65 61 73 65 20 69 6E 73 65 72 74 20 61 20 62 6F 6F 74 61 62 6C
00000090 65 20 66 6C 6F 70 70 79 20 61 6E 64 0D 0A 70 72 65 73 73 20 61 6E 79 20
000000A8 6B 65 79 20 74 6F 20 74 72 79 20 61 67 61 69 6E 20 2E 2E 2E 20 0D 0A 00
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000D8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000108 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Nastąpiła zmiana, pierwsze wiersze pliku informują o systemie plików.

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ mke2fs -t ext4 sfile.raw
mke2fs 1.47.0 (5-Feb-2023)
sfile.raw contains a vfat file system
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 102400 1k blocks and 25584 inodes
Filesystem UUID: e27b886c-bdb9-4f64-bdfa-9d63c04a9620
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
00000401 63 00 00 00 90 01 00 00 14 00 00 CF 60 01 00 E5 63 00 00 01 00 00 00 00 00
0000041A 00 00 00 00 00 00 00 00 20 00 00 B0 07 00 00 00 00 00 00 20 B1 57
00000433 65 00 00 FF FF 53 EF 01 00 01 00 00 20 B1 57 65 00 00 00 00 00 00 00 00
0000044C 01 00 00 00 00 00 00 00 0B 00 00 00 00 01 00 00 3C 00 00 00 C2 02 00 00 6B
00000465 04 00 00 E2 7B 88 6C BD B9 4F 64 BD FA 9D 63 C0 4A 96 20 00 00 00 00 00 00
0000047E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000497 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000004B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000004C9 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 00
000004E2 00 00 00 00 00 00 00 00 00 00 F9 86 F4 D0 3D DC 49 E9 87 EA BB 5D D9 75 AC
000004FB A9 01 01 40 00 0C 00 00 00 00 00 00 20 B1 57 65 0A F3 01 00 04 00 00 00
00000514 00 00 00 00 00 00 00 00 10 00 00 01 C0 00 00 00 00 00 00 00 00 00 00 00
0000052D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000546 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 20 00 20
0000055F 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 01 00 00
00000578 17 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ dumpe2fs sfile.raw
dumpe2fs 1.47.0 (5-Feb-2023)
Filesystem volume name: <none>
Last mounted on: <not available>
Filesystem UUID: e27b886c-bdb9-4f64-bdfa-9d63c04a9620
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features: has_journal ext_attr resize_inode dir_index filetype ext
se_super large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags: signed_directory_hash
Default mount options: user_xattr acl
Filesystem state: clean
Errors behavior: Continue
Filesystem OS type: Linux
Inode count: 25584
Block count: 102400
Reserved block count: 5120
```

Odp na zagadnienia:

„Magiczny numer” badanego obrazu: 0xEF53

Numer UUID: e27b886c-bdb9-4f64-bdfa-9d63c04a9620

Wielkość bloku: 1024

Liczba wolnych bloków: 90319

Checksum typ: crc32c

ile wolnych bloków zawiera grupa nr 12?: Free blocks: 98305-102399

Zostało zużytych ok~7% bloków

ZADANIE 5:

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ sudo losetup --find --show sfile.raw
/dev/loop1

(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]
$ sudo mount /dev/loop1 /mnt/hgfs
mount: /mnt/hgfs: mount point does not exist.
dmesg(1) may have more information after failed mount system call.
```

błąd

```
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]  
$ sudo mkdir -p /mnt/hgfs  
  
(szmpns@kali)-[~/InformatykaSledcza/InfSl_LAB03]  
$ sudo mount /dev/loop1 /mnt/hgfs
```

teraz powinno działać

```
(szmpns@kali)-[/mnt]  
$ ls  
hgfs  
  
(szmpns@kali)-[/mnt]  
$ cd hgfs  
  
(szmpns@kali)-[/mnt/hgfs]  
$ ls  
lost+found
```

zgadza się

```
(szmpns@kali)-[/mnt/hgfs]  
$ ls -li  
total 12  
11 drwx----- 2 root root 12288 Nov 17 19:29 lost+found
```

Plik lost+found służy do podejrzenia zawartości plików, które zostały usunięte(lub utracone) podczas operacji na dysku. Zawiera wskaźniki na miejsce w pamięci usuniętych(utraconych) plików. Plik nie zawiera nazw usuniętych(lub utraconych) plików.

```
(szmpns@kali)-[/mnt/hgfs]
$ sudo touch newitem

(szmpns@kali)-[/mnt/hgfs]
$ ls
lost+found  newitem

(szmpns@kali)-[/mnt/hgfs]
$ sudo dd if=/dev/zero of=/dev/loop1 count=1 bs=1024 seek=1
1+0 records in
1+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 4.6556e-05 s, 22.0 MB/s
```

W obrazie nie ma żadnych plików:

```
(szmpns@kali)-[/mnt/hgfs]
$ ls

(szmpns@kali)-[/mnt/hgfs]
$
```

Udało się odmontować bez większych problemów:

```
(szmpns@kali)-[/mnt/hgfs]
$ sudo umount /mnt/hgfs
umount: /mnt/hgfs: target is busy.

(szmpns@kali)-[/mnt/hgfs]
$ sudo umount -l /mnt/hgfs
```

Nastąpił problem przy ponownym montowaniu:

```
(szmpns@kali)~$ sudo mount /dev/loop1 /mnt/hgfs
mount: /mnt/hgfs: wrong fs type, bad option, bad superblock on /dev/loop1, missing codepage or help
program, or other error.
        dmesg(1) may have more information after failed mount system call.
```

Superblok jest uszkodzony, dostałem możliwość jego naprawy:

```
(szmpns@kali)~[/InformatykaSledcza/InfSI_LAB03]
$ fsck.ext4 sfile.raw
e2fsck 1.47.0 (5-Feb-2023)
ext2fs_open2: Bad magic number in super-block
fsck.ext4: Superblock invalid, trying backup blocks...
Superblock needs_recovery flag is clear, but journal has data.
Recovery flag not set in backup superblock, so running journal anyway.
sfile.raw: recovering journal
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Block bitmap differences: +(8193--8450) +(24577--24834) +(40961--41218) +(57345--57602) +(73729--
73986)
Fix<y>? yes
Free inodes count wrong for group #0 (1957, counted=1956).
Fix<y>? yes
Free inodes count wrong (25573, counted=25572).
Fix<y>? yes
Padding at end of inode bitmap is not set. Fix<y>? yes

sfile.raw: ***** FILE SYSTEM WAS MODIFIED *****
sfile.raw: 12/25584 files (0.0% non-contiguous), 12081/102400 blocks
```

```
(szmpns@kali)~[/mnt/hgfs]
$ sudo fsck -f -y -b 8193 /dev/loop1 /mnt/hgfs
fsck from util-linux 2.39.2
e2fsck 1.47.0 (5-Feb-2023)
e2fsck 1.47.0 (5-Feb-2023)
fsck.ext2: Is a directory while trying to open /mnt/hgfs

The superblock could not be read or does not describe a valid ext2/ext3/ext4
filesystem. If the device is valid and it really contains an ext2/ext3/ext4
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>
or
    e2fsck -b 32768 <device>

Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Free inodes count wrong for group #0 (1957, counted=1956).
Fix? yes

Free inodes count wrong (25573, counted=25572).
Fix? yes

Padding at end of inode bitmap is not set. Fix? yes
```

Po naprawie montaż przebiegł błyskawicznie:

```
(szmpns@kali)-[/mnt/hgfs]
$ sudo mount /dev/loop1 /mnt/hgfs
```

```
(szmpns@kali)-[/mnt/hgfs]
$ ls
lost+found  newitem
```

usunięty plik newitem znajduje się w środowisku

```
(szmpns@kali)-[/mnt/hgfs]
$ ll
total 12
drwx----- 2 root root 12288 Nov 17 19:29 lost+found
-rw-r--r-- 1 root root      0 Nov 17 19:51 newitem
```