



A Arte da Engenharia Reversa

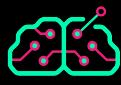


Arquivos



O que é um arquivo?

- Zero ou mais números armazenados em algum lugar.
- Pode-se dizer que contém dados.
- Além disso, há os metadados (mais sobre eles em breve).



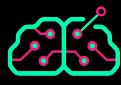
Inspecionando o conteúdo de um arquivo

```
user@DESKTOP-3QCES1T:/tmp]$ hd -n32 /bin/ls
00000000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00 |.ELF.....
00000010  03 00 3e 00 01 00 00 00  d0 61 00 00 00 00 00 00 |..>.....a....|
00000020
```

Offsets (em hexa)

Conteúdo (byte a byte)

Representação
ASCII (se houver)



Inspecionando o conteúdo de um arquivo

```
$ hexyl -n32 maf15.png
```

00000000	89 50 4e 47 0d 0a 1a 0a	00 00 00 0d 49 48 44 52	xPNG_•-	000_IHDR
00000010	00 00 01 90 00 00 01 1e	08 02 00 00 00 2b 67 d4	00•x00••	••000+g×



Inspecionando os metadados de um arquivo

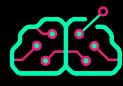
Anteriormente:

```
└$ echo -n 'ab' > arquivo.txt
└$ xxd -g1 arquivo.txt
00000000: 61 62
```

```
└$ ls -l arquivo.txt
-rw-r--r-- 1 kali kali 2 Jun  9 21:01 arquivo.txt
```

permisões usuário grupo data e hora nome
usuário dono grupo dono data e hora da última modificação nome
tipo de recurso contador de link tamanho

```
└$ stat arquivo.txt
  File: arquivo.txt
  Size: 2          Blocks: 8          IO Block: 4096   regular file
Device: 8,1      Inode: 3636452      Links: 1
Access: (0644/-rw-r--r--) Uid: ( 1000/      kali)  Gid: ( 1000/      kali)
Access: 2024-06-09 21:01:46.140544945 -0300
Modify: 2024-06-09 21:01:34.690822934 -0300
Change: 2024-06-09 21:01:34.690822934 -0300
 Birth: 2024-06-03 21:45:26.549882220 -0300
```



Inspecionando os metadados de um arquivo

```
$ ls -l a.txt
-rw-r--r-- 1 jupyter-professor jupyter-professor 3 Jul 17 23:43 a.txt
$ 
$ lsattr a.txt
-----e----- a.txt
$ 
$ stat a.txt
  File: a.txt
  Size: 3          Blocks: 8          IO Block: 4096   regular file
Device: ca01h/51713d   Inode: 289104      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/jupyter-professor)  Gid: ( 1003/jupyter-professor)
Access: 2022-07-17 23:28:43.964685618 +0000
Modify: 2022-07-17 23:43:22.048733405 +0000
Change: 2022-07-17 23:43:22.048733405 +0000
 Birth: 2022-07-17 23:28:43.964685618 +0000
```



Ordem dos bytes (para números)

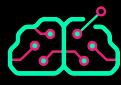
- Número 2024
- Em binário

```
obase=2  
2024  
11111101000
```

- Em hexa

```
obase=16  
2024  
7E8
```

- Em 4 bytes:
 - 00 00 07 E8
- Big-Endian (MSB):
 - 00 00 07 E8
- Little-Endian (LSB):
 - E8 07 00 00



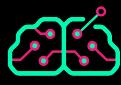
Ordem dos bytes (para números)

- Endianness / Byte order / Byte sex (IDA)
 - Least Significant Byte (LSB) / Little-Endian (LE)
 - Most Significant Byte (MSB) / Big-Endian (BE)



Exercício

- 435 em decimal
- Escrever em hexa (4 bytes)
 - Big-Endian
 - Little-Endian



Exercício - Respostas

- 435 em decimal
- Escrever em hexa (4 bytes)
 - Big-Endian: 00 00 01 B3
 - Little-Endian: B3 01 00 00



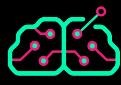
Demonstração

```
└─( kali㉿kali )-[ ~ ]  
└─$ python  
Python 3.11.7 (main, Dec  8 2023, 14:22:46) [GCC 13.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> f = open('a.txt', 'wb')  
>>>
```



Lab 03

Visualizando arquivos em hexadecimal



Lab 03 - Visualizando arquivos em hexadecimal

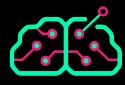
1. Abra o Debian (via WSL) e crie um arquivo contendo seu nome e sobrenome:

```
echo -n "nome sobrenome" > arquivo.txt
```

Agora execute as tarefas abaixo e veja se a saída dos três programas faz **total** sentido para você.

- a) Inspecione o conteúdo do arquivo com o hexyl.
- b) Inspecione o conteúdo do arquivo com o hd.
- c) Inspecione o conteúdo do arquivo com o od.

Se houver alguma dúvida na saída de um dos comandos, pergunte! 😊



Lab 03 - Visualizando arquivos em hexadecimal

2. Crie um arquivo que contenha o **número** 2024 (perceba que não é o texto “2024”). Para isso, use o seguinte programa em Python:

```
ano = 2024
f = open("arquivo.bin", "wb")
f.write(ano.to_bytes(4, byteorder='little'))
f.close()
```

Agora inspecione-o com qualquer visualizador hexadecimal de linha de comando e explique os bytes encontrados lá.



Lab 03 - Respostas

1. `$ echo -n "nome sobrenome" > arquivo.txt`

a) Inspecione o conteúdo do arquivo com o hexyl.

```
└─$ hexyl arquivo.txt
00000000  6e 6f 6d 65 20 73 6f 62  72 65 6e 6f 6d 65  nome sob renome
```

b) Inspecione o conteúdo do arquivo com o hd.

```
└─$ hd arquivo.txt
00000000  6e 6f 6d 65 20 73 6f 62  72 65 6e 6f 6d 65  |nome sobrenome|
0000000e
```

c) Inspecione o conteúdo do arquivo com o od.

```
└─$ od -t x1 arquivo.txt
00000000  6e 6f 6d 65 20 73 6f 62  72 65 6e 6f 6d 65
00000016
```

-t é para definir formato
x1 é hexa de um em um byte



Lab 03 - Respostas

2. `$ python3 lab3.py`

```
ano = 2024
f = open("arquivo.bin", "wb")
f.write(ano.to_bytes(4, byteorder='little'))
f.close()
```

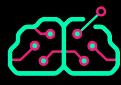
```
└─$ hexyl arquivo.bin
00000000 e8 07 00 00 |          | x 00 |
```



4 bytes:
00 00 07 E8 -> big endian
E8 07 00 00 -> little endian

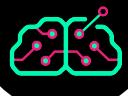


Arquivos Binários



Arquivos Binários

- Depende do contexto:
 - Qualquer arquivo que não seja texto puro, ou seja, que não tenha todos os seus bytes dentro da faixa ASCII/Unicode (que veremos a seguir).
 - Arquivos executáveis (por isso /bin, /sbin, etc).
- No fim, todo arquivo pode, em teoria, ser chamado de binário.



Cadeias de Texto



ASCII 7-bits

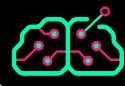
00	nul	01	soh	02	stx	03	etx	04	eot	05	enq	06	ack	07	bel
08	bs	09	ht	0a	nl	0b	vt	0c	np	0d	cr	0e	so	0f	si
10	dle	11	dc1	12	dc2	13	dc3	14	dc4	15	nak	16	syn	17	etb
18	can	19	em	1a	sub	1b	esc	1c	fs	1d	gs	1e	rs	1f	us
20	sp	21	!	22	"	23	#	24	\$	25	%	26	&	27	'
28	(29)	2a	*	2b	+	2c	,	2d	-	2e	.	2f	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3a	:	3b	;	3c	<	3d	=	3e	>	3f	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4a	J	4b	K	4c	L	4d	M	4e	N	4f	0
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5a	Z	5b	[5c	\	5d]	5e	^	5f	_
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6a	j	6b	k	6c	l	6d	m	6e	n	6f	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7a	z	7b	{	7c		7d	}	7e	~	7f	del



Exercício

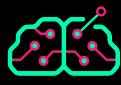
```
└$ echo -n seu_nome | hd  
00000000  73 65 75 5f 6e 6f 6d 65          | seu_nome |  
00000008
```

00	nul	01	soh	02	stx	03	etx	04	eot	05	enq	06	ack	07	bel
08	bs	09	ht	0a	nl	0b	vt	0c	np	0d	cr	0e	so	0f	si
10	dle	11	dc1	12	dc2	13	dc3	14	dc4	15	nak	16	syn	17	etb
18	can	19	em	1a	sub	1b	esc	1c	fs	1d	gs	1e	rs	1f	us
20	sp	21	!	22	"	23	#	24	\$	25	%	26	&	27	'
28	(29)	2a	*	2b	+	2c	,	2d	-	2e	.	2f	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3a	:	3b	;	3c	<	3d	=	3e	>	3f	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4a	J	4b	K	4c	L	4d	M	4e	N	4f	O
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5a	Z	5b	[5c	\	5d]	5e	^	5f	_
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6a	j	6b	k	6c	l	6d	m	6e	n	6f	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7a	z	7b	{	7c		7d	}	7e	~	7f	del



ASCII 7-bits

```
$ sudo apt install ascii
$ ascii -b
 0000000 NUL    0010000 DLE    0100000      0110000  0    1000000 @    1010000 P    1100000 `    1110000 p
 0000001 SOH    0010001 DC1    0100001 !    0110001 1    1000001 A    1010001 Q    1100001 a    1110001 q
 0000010 STX    0010010 DC2    0100010 "    0110010 2    1000010 B    1010010 R    1100010 b    1110010 r
 0000011 ETX    0010011 DC3    0100011 #    0110011 3    1000011 C    1010011 S    1100011 c    1110011 s
 0000100 EOT    0010100 DC4    0100100 $    0110100 4    1000100 D    1010100 T    1100100 d    1110100 t
 0000101 ENQ    0010101 NAK    0100101 %    0110101 5    1000101 E    1010101 U    1100101 e    1110101 u
 0000110 ACK    0010110 SYN    0100110 &    0110110 6    1000110 F    1010110 V    1100110 f    1110110 v
 0000111 BEL    0010111 ETB    0100111 '    0110111 7    1000111 G    1010111 W    1100111 g    1110111 w
 0001000 BS     0011000 CAN    0101000 (    0111000 8    1001000 H    1011000 X    1101000 h    1111000 x
 0001001 HT     0011001 EM     0101001 )    0111001 9    1001001 I    1011001 Y    1101001 i    1111001 y
 0001010 LF     0011010 SUB    0101010 *    0111010 :    1001010 J    1011010 Z    1101010 j    1111010 z
 0001011 VT     0011011 ESC    0101011 +    0111011 ;    1001011 K    1011011 [    1101011 k    1111011 {
 0001100 FF     0011100 FS     0101100 ,    0111100 <   1001100 L    1011100 \    1101100 l    1111100 |
 0001101 CR     0011101 GS     0101101 -    0111101 =   1001101 M    1011101 ]    1101101 m    1111101 }
 0001110 SO     0011110 RS     0101110 .    0111110 >   1001110 N    1011110 ^    1101110 n    1111110 ~
 0001111 SI     0011111 US     0101111 /    0111111 ?   1001111 O    1011111 _    1101111 o    1111111 DEL
```



Relações

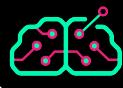
- Dígitos de 0x30 a 0x39
- Maiúsculas de 0x41 a 0x5a
- Minúsculas de 0x61 a 0x7a
- Imprimíveis de 0x20 a 0x7f
- Maiúscula = minúsculas - 32 (0x20)
- Equivalente numérico = Dígito - 0x30



Upper do Python

No terminal do Python, o que faz a função upper?

```
>>> 'simplesmente'.upper()  
'SIMPLESMENDE'
```



O fim de linha

- Windows (herança do MS-DOS, que herdou do CP/M) → \r\n ou CrLf ou 0x0d 0x0a
- Linux (herança do Unix) → \n ou LF ou 0x0a
 - O comando **dos2unix** "converte" ao remover os \r dos pares \r\n encontrados num arquivo.



Lab 04

Convertendo strings para maiúsculo



Lab 04 - Convertendo strings para maiúsculo

1. Escreva uma função chamada *toup()*, na linguagem que quiser, que receba um caractere e, caso este seja minúsculo, converta-o para maiúsculo. Sua função não deve utilizar **nenhuma** função de conversão de caracteres para maiúsculos, mas você pode usar *strlen()* para obter o tamanho de uma string. Teste sua função.

Se for escrever em C/C++, utilize o DevC++ em **retoolkit → Programming → DevCpp**. Abra o arquivo %userprofile%\desktop\binarios\toup.c e altere nele o que precisar.

Se preferir Python, abra o prompt em **retoolkit → Programming → Python Command Prompt** e depois vá até o diretório onde o esqueleto está:

```
cd %userprofile%/desktop/binarios  
python toup.py
```

Os esqueletos não estão alterando a string. Você precisa implementar a lógica que vai realizar essa ação.

Teste sua função com diferentes strings, de diferentes tamanhos e contendo caracteres alfanuméricos e de pontuação.



Lab 04 - Convertendo strings para maiúsculo

C

```
#include <stdio.h>
#include <string.h>

char* toup(char *s) {
    for (int i=0; i < strlen(s); i++) {
        s[i] = s[i];
    }
    return s;
}

int main(void) {
    char s[] = "menteb.in";
    puts(toup(s));
}
```

Python

```
def toup(s):
    temp = ""
    for c in s:
        temp = temp + chr(ord(c))
    return temp

s = "menteb.in"
print(toup(s))
```



Lab 04 - Strings para maiúsculo - Respostas

C

```
#include <stdio.h>
#include <string.h>

char* toup(char *s) {
    for (int i=0; i < strlen(s); i++) {
        if (s[i] >= 0x61 && s[i] <= 0x7a) s[i] = s[i] - 32;
    }
    return s;
}
int main(void) {
    char s[] = "menteb.in";
    puts(toup(s));
}
```



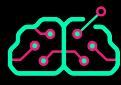
ASCII 7-bits - Curiosidades

- 0x00 é NUL / nullbyte
- 0x08 é o BS / backspace

```
└$ echo -e "menteb\x08.in"
mente.in
```

- 0x07 é o BEL
- 0x09 é o HT / horizontal tab

```
└$ echo -e "menteb\x07.in"
menteb .in
```



ASCII estendida

- 8 bits
- Não é um padrão
- Centenas de versões diferentes
- A mais famosa é a ISO 8859-1, ou ISO Latin 1, ou o seu superconjunto Windows (Code Page) CP-1252
- Altamente dependente do *encoding*



Encoding

The screenshot shows the Notepad++ interface with the 'Formatar' (Format) menu open, specifically the 'Codificação' (Encoding) submenu. The submenu lists various encoding options, with 'Conjunto de caracteres' (Character Set) currently selected. A secondary dropdown menu is open under 'Conjunto de caracteres', showing options like Árabe, Báltico, Celta, Cirílico, etc., with 'ISO 8859-5' highlighted.

*novo 12 - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?

Novo 2 combinacoes.py

1 >
2 [-]+
3 >
4 [-]
5 <<
6 [
7 [-]>-<
8 >>
9 ++++++++-+-----+
10 <<
11]
12 >
13 [
14 [-]
15 >
16 ++++++++-+-----+
17 <

Codificação em ANSI
Codificação em UTF-8
Codificação em UTF-8 BOM
Codificação em UCS-2 Big Endian BOM
Codificação em UCS-2 Little Endian BOM
Conjunto de caracteres
Converter para ANSI
Converter para UTF-8
Converter para UTF-8-BOM
Converter para UCS-2 Big Endian BOM
Converter para UCS-2 Little Endian BOM

Árabe
Báltico
Celta
Cirílico
Europa Central
Chinês
Europa Oriental
Grego
Hebraico
Japonês
Coreano
Norte Europeu
Tailandês
Turco
Europa Ocidental
Vietnamita

ISO 8859-5
KOI8-R
KOI8-U
Macintosh
OEM 855
OEM 866
Windows-1251



ASCII estendida (Windows CP-1252 / ISO 8859-1)

Ø	Latin capital letter O with slash	216	D8	Ø	Ø	+	Box drawings vertical single and horizontal double
Ù	Latin capital letter U with grave	217	D9	Ù	Ù	└	Box drawings light up and left
Ú	Latin capital letter U with acute	218	DA	Ú	Ú	Γ	Box drawings light down and right
Û	Latin capital letter U with circumflex	219	DB	Û	Û	█	Full block
Ü	Latin capital letter U with diaeresis	220	DC	Ü	Ü	▬	Lower half block
Ý	Latin capital letter Y with acute	221	DD	Ý	Ý	▀	Left half block
Þ	Latin capital letter THORN	222	DE	Þ	Þ	█	Right half block
ß	Latin small letter sharp s - ess-zed	223	DF	ß	ß	■	Upper half block
à	Latin small letter a with grave	224	E0	à	à	¤	Greek lower case alpha
á	Latin small letter a with acute	225	E1	á	á	þ	Lower case sharp s
â	Latin small letter a with circumflex	226	E2	â	â	Γ	Greek upper case letter gamma
ã	Latin small letter a with tilde	227	E3	ã	ã	π	Greek lower case pi
ä	Latin small letter a with diaeresis	228	E4	ä	ä	Σ	Greek upper case letter sigma
å	Latin small letter a with ring above	229	E5	å	å	σ	Greek lower case sigma
æ	Latin small letter ae	230	E6	æ	æ	μ	Micro sign
ç	Latin small letter c with cedilla	231	E7	ç	ç	τ	Greek lower case tau
è	Latin small letter e with grave	232	E8	è	è	Φ	Greek upper case letter phi
é	Latin small letter e with acute	233	E9	é	é	Θ	Greek upper case letter theta
ê	Latin small letter e with circumflex	234	EA	ê	ê	Ω	Greek upper case letter omega
ë	Latin small letter e with diaeresis	235	EB	ë	ë	δ	Greek lower case delta
í	Latin small letter i with grave	236	FC	ì	ì	∞	Infinity

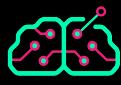
(cortada)



Unicode

- Também chamada de *wide strings*
- UTF-8
 - Caracteres de 1 a 4 bytes
 - Ex.: 'á' é a dupla 0xc3 0xa1
- UTF-16 / UCS-2
 - LE ou BE
 - 2 bytes para cada caractere, com zeros se preciso.
 - Windows "encoda" CP-1252 em UTF-16-LE.
 - Usada em .NET no Windows.
 - Termina com **dois** null bytes.

```
$ echo -n 'água' | hd  
00000000 c3 a1 67 75 61 | .. gual  
00000005
```



Unicode

- UTF-32
 - LE ou BE
 - 4 bytes para cada caractere, com zeros se preciso.
 - Usada pela `wchar.h` no Linux.

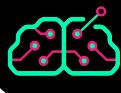


C Strings

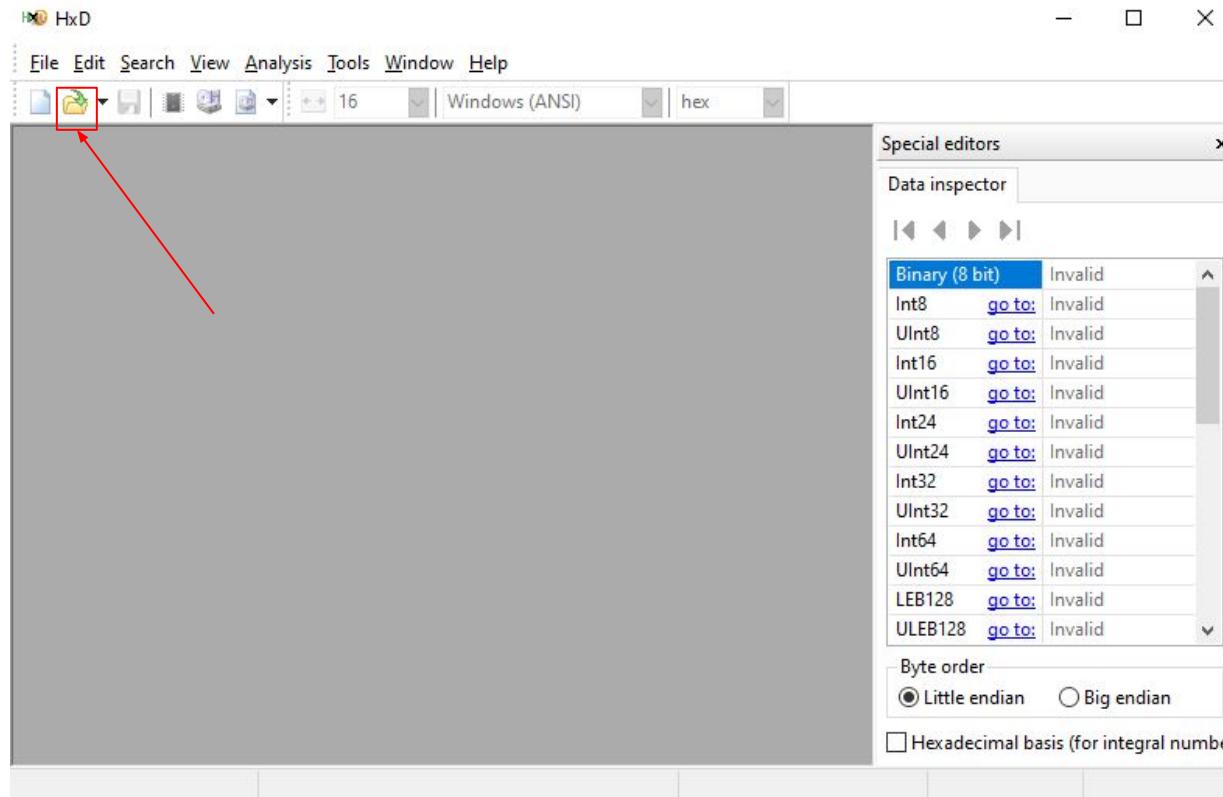
- Programas em C adicionam um caractere nulo no final da string.
- Saber disso é útil, por exemplo, para cortar uma string quando precisamos.



Editores Hexadecimais

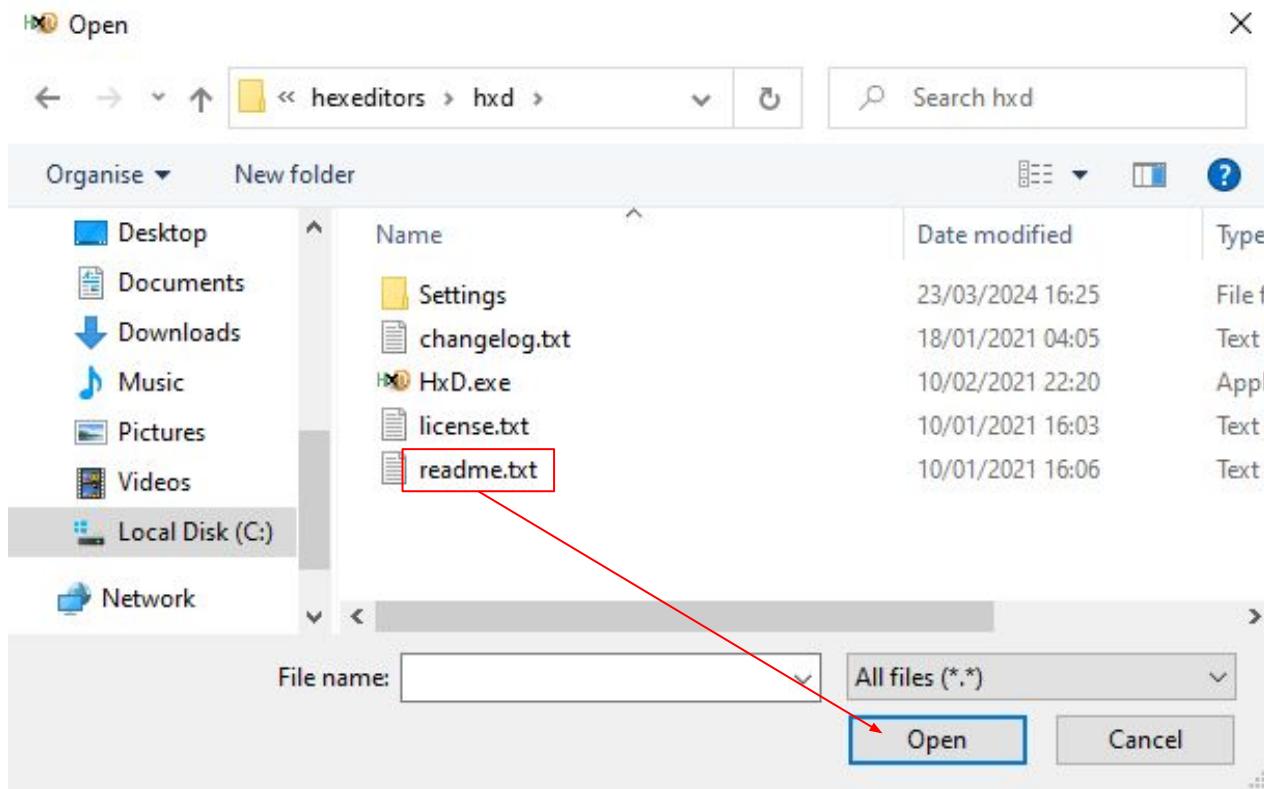


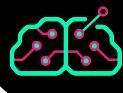
Usando o HxD





Usando o HxD





Usando o HxD

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\readme.txt]

File Edit Search View Analysis Tools Window Help

readme.txt

Offset(h)	Decoded text
00000000	EF BB BF
00000010	48 78 44 20 48 65 78 20 45 64 69 74 6F
00000020	72 20 52 45 41 44 4D 45 0D 0A 3D 3D 3D 3D 3D 3D
00000030	3D 0D
00000040	0A 0D 0A 48 78 44 20 43 6F 70 79 72 69 67 68 74
00000050	C2 A9 20 32 30 30 32 2D 32 30 32 31 20 62 79 20
00000060	4D 61 C3 AB 6C 20 48 C3 B6 72 7A 2E 20 41 6C 6C
00000070	20 72 69 67 68 74 73 20 72 65 73 65 72 76 65 64
00000080	2E 0D 0A 53 65 65 20 61 6C 73 6F 20 74 68 65 20
00000090	6C 69 63 65 6E 73 65 20 66 69 6C 65 2E 0D 0A 0D
000000A0	0A 0D 0A 46 65 61 74 75 72 65 73 0D 0A 3D 3D 3D
000000B0	3D 3D 3D 3D 3D 0D 0A 2D 20 44 61 74 61 20 69 6E
000000C0	73 70 65 63 74 6F 72 0D 0A 20 20 2D 20 69 6E 74
000000D0	65 72 70 72 65 74 73 20 62 79 74 65 73 20 61 74
000000E0	20 74 68 65 20 63 75 72 72 65 6E 74 20 63 61 72
000000F0	65 74 20 70 6F 73 69 74 69 6F 6E 20 69 6E 74 6F
00000100	20 76 61 72 69 6F 75 73 20 64 61 74 61 74 79 70
00000110	65 73 3A 20 0D 0A 20 20 20 2D 20 62 69 6E 61
00000120	72 79 20 28 62 69 74 20 73 65 71 75 65 6E 63 65
00000130	29 2C 20 69 6E 74 65 67 65 72 2C 20 66 6C 6F 61
00000140	74 73 2C 20 74 69 6D 65 20 61 6E 64 20 64 61 74
00000150	65 2C 20 63 68 61 72 61 63 74 65 72 2C 20 47 55
00000160	49 44 20 61 6E 64 20 64 69 73 61 73 73 65 6D 62
00000170	6C 79 20 28 78 38 36 20 61 6E 64 20 41 4D 44 36

Offset(h): 0 conteúdo em hexadecimal representação ASCII

Special editors

Data inspector

Binary (8 bit) 11101111

Int8 go to: -17

UInt8 go to: 239

Int16 go to: -17425

UInt16 go to: 48111

Int24 go to: -4211729

UInt24 go to: 12565487

Int32 go to: 1220525039

UInt32 go to: 1220525039

Int64 go to: 519722925339284

UInt64 go to: 519722925339284

LEB128 go to: -116400657

ULEB128 go to: 152034799

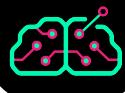
AnsiChar / char8. T

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)

Overwrite



Usando o HxD - offset

Qual o byte nas seguintes posições?

- 0

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\readme.txt]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

Special editors

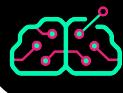
Data inspector

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text

00000000	EF	BB	BF	48	78	44	20	48	65	78	20	45	64	69	74	6F	»»HxD Hex Edito
00000010	72	20	52	45	41	44	4D	45	0D	0A	3D	3D	3D	3D	3D	3D	r README..=====
00000020	3D	0D	=====.														
00000030	0A	0D	0A	48	78	44	20	43	6F	70	79	72	69	67	68	74	...HxD Copyright
00000040	C2	A9	20	32	30	30	32	20	32	30	32	31	20	62	79	20	Â© 2002-2021 by
00000050	4D	61	C3	AB	6C	20	48	C3	B6	72	7A	2E	20	41	6C	6C	MaÃ«l HÃ¶rz. All
00000060	20	72	69	67	68	74	73	20	72	65	73	65	72	76	65	64	rights reserved
00000070	2E	0D	0A	53	65	65	20	61	6C	73	6F	20	74	68	65	20	...See also the
00000080	6C	69	63	65	6E	73	65	20	66	69	6C	65	2E	0D	0A	0D	license file....
00000090	0A	0D	0A	46	65	61	74	75	72	65	73	0D	0A	3D	3D	3D	...Features..====
000000A0	3D	3D	3D	3D	3D	0D	0A	2D	20	44	61	74	61	20	69	6E	=====.. Data in
000000B0	73	70	65	63	74	6F	72	0D	0A	20	20	2D	20	69	6E	74	spector.. - int
000000C0	65	72	70	72	65	74	73	20	62	79	74	65	73	20	61	74	erprets bytes at
000000D0	20	74	68	65	20	63	75	72	72	65	6E	74	20	63	61	72	the current car
000000E0	65	74	20	70	6F	73	69	74	69	6F	6E	20	69	6E	74	6F	et position into
000000F0	20	76	61	72	69	6F	75	73	20	64	61	74	61	74	79	70	various datatype
00000100	65	73	3A	20	0D	0A	20	20	20	2D	20	62	69	6E	61	0E	.. - binary (bit sequence
00000110	72	79	20	28	62	69	74	20	73	65	71	75	65	6E	63	65), integer, float
00000120	29	2C	20	69	6E	74	65	67	65	72	2C	20	66	6C	6F	61	, time and date
00000130	74	73	2C	20	74	69	6D	65	20	61	6E	64	20	64	61	74	character, GUI
00000140	65	2C	20	63	68	61	72	61	63	74	65	72	2C	20	47	55	ID and disassembly
00000150	49	44	20	61	6E	64	20	64	69	73	61	73	73	65	6D	62	(x86 and AMD64)
00000160	6C	79	20	28	78	38	36	20	61	6E	64	20	41	4D	44	36	Instant on
00000170	34	29	0D	0A	2D	20	49	6F	73	74	61	6F	74	20	6F	70	41.. - Instant on

Offset(h): 0

Overwrite



Usando o HxD - offset

Qual o byte nas seguintes posições?

- A

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\readme.txt]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

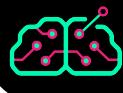
Special editors

Data inspector

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text

00000000	EF BB BF	48 78 44 20	48 65 78	20 45	64 69 74 6F	»»¿HxD Hex Edito
00000010	72 20 52 45	41 44 4D 45	0D 0A 3D	3D 3D 3D 3D 3D	3D 3D 3D 3D	r README..=====
00000020	3D 3D 3D 3D	3D 3D 3D 3D	=====.			
00000030	0A 0D 0A 48	78 44 20 43	6F 70 79 72	69 67 68 74	...HxD Copyright	
00000040	C2 A9 20 32	30 30 32 32	30 32 31 20	62 79 20	Â© 2002-2021 by	
00000050	4D 61 C3 AB	6C 20 48	C3 B6 72	7A 2E 20 41	6C 6C MaÃ«l HÃ¶rz. All	
00000060	20 72 69 67	68 74 73 20	72 65 73 65	72 76 65 64	rights reserved	
00000070	2E 0D 0A 53	65 65 20 61	6C 73 6F 20	74 68 65 20	...See also the	
00000080	6C 69 63 65	6E 73 65 20	66 69 6C 65	2E 0D 0A 0D	license file....	
00000090	0A 0D 0A 46	65 61 74 75	72 65 73 0D	0A 3D 3D 3D	...Features..====	
000000A0	3D 3D 3D 3D	0D 0A 2D 20	44 61 74 61	20 69 6E	=====.. Data in	
000000B0	73 70 65 63	74 6F 72 0D	0A 20 20 2D	69 6E 74	spector.. - int	
000000C0	65 72 70 72	65 74 73 20	62 79 74 65	73 20 61 74	erprets bytes at	
000000D0	20 74 68 65	20 63 75 72	72 65 6E 74	20 63 61 72	the current car	
000000E0	65 74 20 70	6F 73 69 74	69 6F 6E 20	69 6E 74 6F	et position into	
000000F0	20 76 61 72	69 6F 75 73	20 64 61 74	61 74 79 70	various datatype	
00000100	65 73 3A 20	0D 0A 20 20	20 20 2D 20	62 69 6E 61	es: .. - binary	
00000110	72 79 20 28	62 69 74 20	73 65 71 75	65 6E 63 65	(bit sequence	
00000120	29 2C 20 69	6E 74 65 67	65 72 2C 20	66 6C 6F 61), integer, float	
00000130	74 73 2C 20	74 69 6D 65	20 61 6E 64	20 64 61 74	, time and date	
00000140	65 2C 20 63	68 61 72 61	63 74 65 72	2C 20 47 55	, character, GUI	
00000150	49 44 20 61	6E 64 20 64	69 73 61 73	73 65 6D 62	ID and disassembly	
00000160	6C 79 20 28	78 38 36 20	61 6E 64 20	41 4D 44 36	(x86 and AMD64)	
00000170	34 29 0D 0A	2D 20 49 6F	73 74 61 6F	74 20 6F 70	41.. Instant on	

Offset(h): 0 Overwrite



Usando o HxD - offset

Qual o byte nas seguintes posições?

- 5A

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\readme.txt]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

Special editors

Data inspector

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text

00000000	EF BB BF	48 78 44 20	48 65 78 20	45 64 69 74	6F »»HxD Hex Edito	
00000010	72 20 52 45	41 44 4D 45	0D 0A 3D 3D	3D 3D 3D 3D	3D 3D 3D 3D	r README..=====
00000020	3D 3D 3D 3D	3D 3D 3D 3D	3D 3D 3D 3D	3D 3D 3D 3D	3D 3D 3D 3D	=====.
00000030	0A 0D 0A 48	78 44 20 43	6F 70 79 72	69 67 68 74	...HxD Copyright	
00000040	C2 A9 20 32	30 30 32 2D	32 30 30 31	20 62 79 20	20 62 79 20	© 2002-2021 by
00000050	4D 61 C3 AD 6C	20 40 C3 B6 72	7A 2E 20 41	6C 6C MaÃ«l HÃ¶rz. All	rights reserved	
00000060	20 72 69 67	68 74 73 20	72 65 73 65	72 65 72 65	64 ...See also the	
00000070	2E 0D 0A 53	65 65 20 61	6C 73 6F 20	74 68 65 20	6C 65 2E 0D 0A 0D license file....	
00000080	6C 69 63 65	6E 73 65 20	66 69 6C 65	2E 65 2E 0D 0A 0D	...Features..====	
00000090	0A 0D 0A 46	65 61 74 75	72 65 73 0D	0A 3D 3D 3D	0A 3D 3D 3D	
000000A0	3D 3D 3D 3D	0D 0A 2D 20	44 61 74 61	20 69 6E	=====.. Data in	
000000B0	73 70 65 63	74 6F 72 0D	0A 20 20 2D	69 6E 74	spector.. - int	
000000C0	65 72 70 72	65 74 73 20	62 79 74 65	73 20 61 74	erprets bytes at	
000000D0	20 74 68 65	20 63 75 72	72 65 6E 74	20 63 61 72	the current car	
000000E0	65 74 20 70	6F 73 69 74	69 6F 6E 20	69 6E 74 6F	et position into	
000000F0	20 76 61 72	69 6F 75 73	20 64 61 74	61 74 79 70	various datatype	
00000100	65 73 3A 20	0D 0A 20 20	20 20 2D 20	62 69 6E 61	es: .. - binary	
00000110	72 79 20 28	62 69 74 20	73 65 71 75	65 6E 63 65	(bit sequence	
00000120	29 2C 20 69	6E 74 65 67	65 72 2C 20	66 6C 6F 61), integer, float	
00000130	74 73 2C 20	74 69 6D 65	20 61 6E 64	20 64 61 74	, time and date	
00000140	65 2C 20 63	68 61 72 61	63 74 65 72	2C 20 47 55	, character, GUI	
00000150	49 44 20 61	6E 64 20 64	69 73 61 73	73 65 6D 62	ID and disassembly	
00000160	6C 79 20 28	78 38 36 20	61 6E 64 20	41 4D 44 36	(x86 and AMD64.. - Instant on	
00000170	34 29 0D 0A	2D 20 49 6F	73 74 61 6F	74 20 6F 70	Overwrite	

Offset(h): 0

Binary (8 bit) 11101111

Int8 go to: -17

UInt8 go to: 239

Int16 go to: -17425

UInt16 go to: 48111

Int24 go to: -4211729

UInt24 go to: 12565487

Int32 go to: 1220525039

UInt32 go to: 1220525039

Int64 go to: 519722925339284

UInt64 go to: 519722925339284

LEB128 go to: -116400657

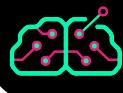
ULEB128 go to: 152034799

AnsiChar / char8: T

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Usando o HxD - offset

Qual o byte nas seguintes posições?

- 164

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\readme.txt]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

Special editors

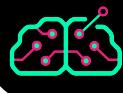
Data inspector

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text

00000000	EF BB BF	48	78	44	20	48	65	78	20	45	64	69	74	6F	»»¿HxD Hex Edito	
00000010	72	20	52	45	41	44	4D	45	0D	0A	3D	3D	3D	3D	r README..=====	
00000020	3D	3D	3D	3D	3D	3D	3D	3D	3D	3D	3D	3D	3D	3D	=====.	
00000030	0A	0D	0A	48	78	44	20	43	6F	70	79	72	69	67	68	74
00000040	C2	A9	20	32	30	30	32	20	43	6F	70	79	72	69	70	20
00000050	4D	61	C3	AB	6C	20	48	C3	B6	72	7A	2E	20	41	6C	6C
00000060	20	72	69	67	68	74	73	20	72	65	73	65	72	76	65	64
00000070	2E	0D	0A	53	65	65	20	61	6C	73	6F	20	74	68	65	20
00000080	6C	69	63	65	6E	73	65	20	66	69	6C	65	2E	0D	0A	0D
00000090	0A	0D	0A	46	65	61	74	75	72	65	73	0D	0A	3D	3D	3D
000000A0	3D	3D	3D	3D	3D	0D	0A	2D	20	44	61	74	61	20	69	6E
000000B0	73	70	65	63	74	6F	72	0D	0A	20	20	2D	20	69	6E	74
000000C0	65	72	70	72	65	74	73	20	62	79	74	65	73	20	61	74
000000D0	20	74	68	65	20	63	75	72	72	65	6E	74	20	63	61	72
000000E0	65	74	20	70	6F	73	69	74	69	6F	6E	20	69	6E	74	6F
000000F0	20	76	61	72	69	6F	75	73	20	64	61	74	61	74	79	70
00000100	65	73	3A	20	0D	0A	20	20	20	2D	20	62	69	6E	61	es: .. - bina
00000110	72	79	20	28	62	69	74	20	73	65	71	75	65	6E	63	65
00000120	29	2C	20	69	6E	74	65	67	65	72	2C	20	66	6C	6F	61
00000130	74	73	2C	20	74	69	6D	65	20	61	6E	64	20	64	61	74
00000140	65	2C	20	63	68	61	72	61	63	74	65	72	2C	20	47	55
00000150	49	44	20	61	6E	64	20	64	69	73	61	73	73	65	6D	62
00000160	6C	79	20	28	78	38	36	20	61	6E	64	20	41	4D	44	36
00000170	34	29	0D	0A	2D	20	49	6E	73	74	61	6E	74	20	6F	70

Offset(h): 0

Overwrite



Usando o HxD - Go to

Screenshot of the HxD Hex Editor showing the "readme.txt" file. A "Go to" dialog box is open, prompting for an offset. The "Offset:" field contains the value 194. The "Decoded text" column shows the ASCII representation of the file's content.

Special editors

Data inspector

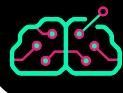
Binary (8 bit)	Value
Int8	go to: -17
UInt8	go to: 239
Int16	go to: -17425
UInt16	go to: 48111
Int24	go to: -4211729
UInt24	go to: 12565487
Int32	go to: 1220525039
UInt32	go to: 1220525039
Int64	go to: 519722925339284
UInt64	go to: 519722925339284
LEB128	go to: -116400657
ULEB128	go to: 152034799
AnsiChar / char8	i

Byte order

- Little endian
- Big endian

Hexadecimal basis (for integral numbers)

Offset(h): 0 Overwrite



Usando o HxD - Go to

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\readme.txt]

File Edit Search View Analysis Tools Window Help

readme.txt

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000030	0A 0D 0A 48 78 44 20 43 6F 70 79 72 69 67 68 74	...HxD Copyright
00000040	C2 A9 20 32 30 30 32 2D 32 30 32 31 20 62 79 20	Â© 2002-2021 by
00000050	4D 61 C3 AB 6C 20 48 C3 B6 72 7A 2E 20 41 6C 6C	MaÃ±el HÃ¶rz. All
00000060	20 72 69 67 68 74 73 20 72 65 73 65 72 76 65 64	rights reserved
00000070	2E 0D 0A 53 65 65 20 61 6C 73 6F 20 74 68 65 20	...See also the
00000080	6C 69 63 65 6E 73 65 20 66 69 6C 65 2E 0D 0A 0D	license file....
00000090	0A 0D 0A 46 65 61 74 75 72 65 73 0D 0A 3D 3D 3D	...Features..==
000000A0	3D 3D 3D 3D 0D 0A 2D 20 44 61 74 61 20 69 6E	=====.. Data in
000000B0	73 70 65 63 74 6F 72 0D 0A 20 20 2D 20 69 6E 74	spector.. - int
000000C0	65 72 70 72 65 74 73 20 62 79 74 65 73 20 61 74	erprets bytes at
000000D0	20 74 68 65 20 63 75 72 72 65 6E 74 20 63 61 72	the current car
000000E0	65 74 20 70 6F 73 69 74 69 6F 6E 20 69 6E 74 6F	et position into
000000F0	20 76 61 72 69 6F 75 73 20 64 61 74 61 74 79 70	various datatype
00000100	65 73 3A 20 0D 0A 20 20 20 2D 20 62 69 6E 61	es: .. - bina
00000110	72 79 20 28 62 69 74 20 73 65 71 75 65 6E 63 65	ry (bit sequence
00000120	29 2C 20 69 6E 74 65 67 65 72 2C 20 66 6C 6F 61), integer, floa
00000130	74 73 2C 20 74 69 6D 65 20 61 6E 64 20 64 61 74	ts, time and dat
00000140	65 2C 20 63 68 61 72 61 63 74 65 72 2C 20 47 55	e, character, GU
00000150	49 44 20 61 6E 64 20 64 69 73 61 73 73 65 6D 62	ID and disassemb
00000160	6C 79 20 28 78 38 36 20 61 6E 64 20 41 4D 44 36	ly (x86 and AMD6
00000170	34 29 0D 0A 2D 20 49 6E 73 74 61 6E 74 20 6F 70	4)..- Instant op
00000180	65 6E 69 6E 67 20 72 65 67 61 72 64 6C 65 73 73	ening regardless
00000190	20 6F 66 20 56 69 6C 65 2D 73 69 7A 65 0D 0A 20	of file-size..
000001A0	20 28 3F 34 47 42 20 69 73 20 6F 6F 20 70 72 6F	(>4GB is no pro

Offset(h): 194

Special editors

Data inspector

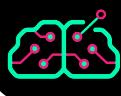
Binary (8 bit)	01100110
Int8	go to: 102
UInt8	go to: 102
Int16	go to: 26982
UInt16	go to: 26982
Int24	go to: 7104870
UInt24	go to: 7104870
Int32	go to: 1701603686
UInt32	go to: 1701603686
Int64	go to: 882070798399415:
UInt64	go to: 882070798399415:
LEB128	go to: -26
ULEB128	go to: 102
AnsiChar / char8	f

Byte order

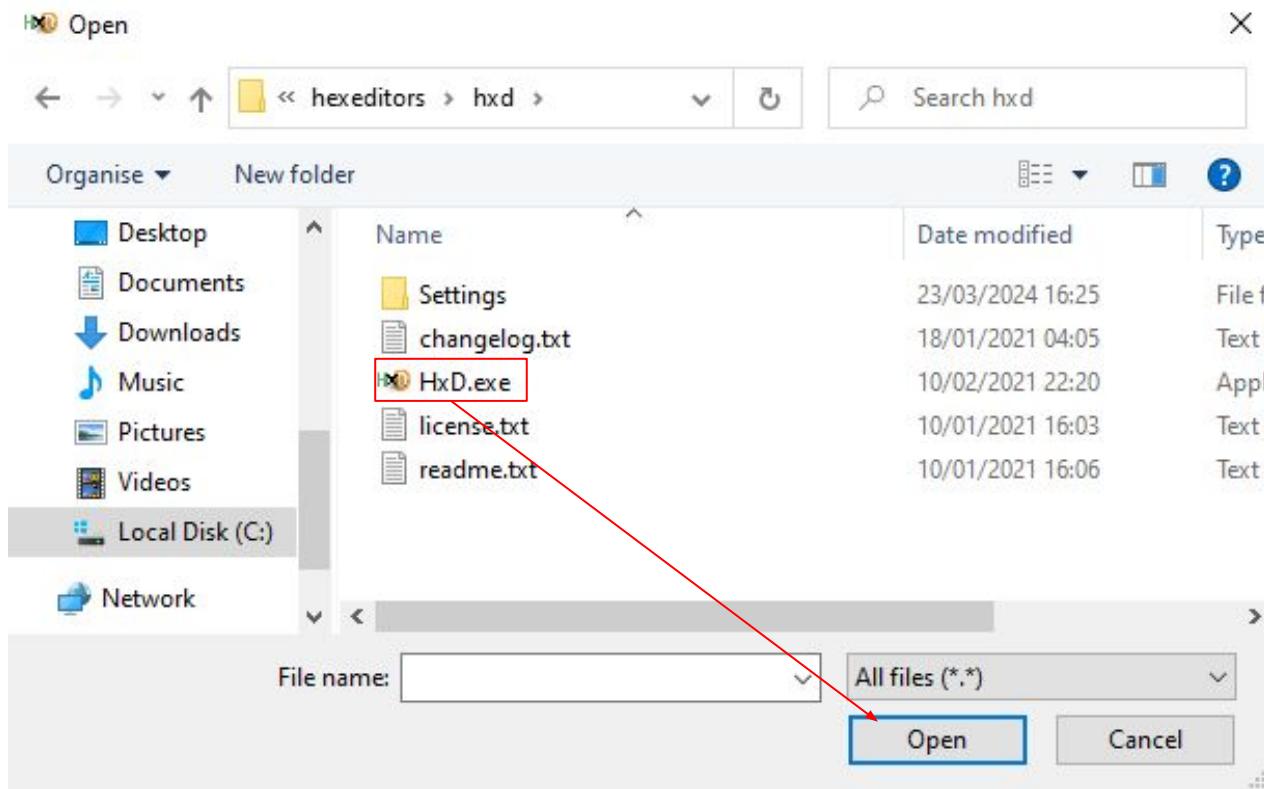
Little endian Big endian

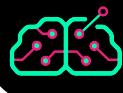
Hexadecimal basis (for integral numbers)

Overwrite



Usando o HxD





Usando o HxD - fim de linha

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

readme.txt HxD.exe

Offset(h)	Decoded text
00000000	ÍD 5A 00 02 00 00 00 04 00 0F 00 FF FF 00 00
00000010	B8 00 00 00 00 00 40 00 1A 00 00 00 00 00 00 00
00000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00000040	BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90
00000050	54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73
00000060	74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57
00000070	69 6E 36 34 0D 0A 24 37 00 00 00 00 00 00 00 00 00
00000080	This program must be run under Win64..\$7.....
00000090
000000A0
000000B0
000000C0
000000D0
000000E0
000000F0
00000100	PE..dt..ùMS`....
00000110ñ.#.....ns.
00000120í.....0(S.....
00000130	...@.....
00000140
00000150	..ej.....í*i...@.
00000160@.....
00000170

Special editors

Data inspector

Binary (8 bit) 01001101

Int8 go to: 77

UInt8 go to: 77

Int16 go to: 23117

UInt16 go to: 23117

Int24 go to: 5265997

UInt24 go to: 5265997

Int32 go to: 5265997

UInt32 go to: 5265997

Int64 go to: 8595200589

UInt64 go to: 8595200589

LEB128 go to: -51

ULEB128 go to: 77

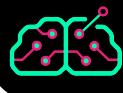
AnsiChar / char8_ M

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)

Offset(h): 0 Overwrite



Usando o HxD - encoding

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

readme.txt HxD.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000840	04 00 00 00 38 11 40 00 00 00 00 00 00 00 00 008.@.....
00000850	00 00 00 00 02 02 44 31 02 00 F8 10 40 00 00 00D1..@.0@..
00000860	00 00 04 00 00 00 00 00 00 02 02 44 32 02 00D2..
00000870	F8 10 40 00 00 00 00 00 06 00 00 00 00 00 00 00	@.0.....
00000880	02 02 44 33 02 00 00 00 00 00 00 00 00 00 00 08	..D3.....
00000890	00 00 00 00 00 00 02 02 44 34 02 00 02 00 03 00D4.....
000008A0	0B 80 4D 41 00 00 00 00 00 OC 26 6F 70 5F 45 71	.EMA.....&op_Eq
000008B0	75 61 6C 69 74 79 00 00 00 10 40 00 00 00 00 00	uality....@.....
000008C0	02 12 28 14 40 00 00 00 00 00 04 4C 65 66 74 02	..(.@.....Left.
000008D0	00 12 28 14 40 00 00 00 00 05 52 69 67 68 74	..(.@.....Right
000008E0	02 00 02 00 0B 08 B3 93 00 00 00 00 00 0E 26 6F".....&c
000008F0	70 5F 49 6E 65 71 75 61 6C 69 74 79 00 00 10	p_inequality....
00000900	40 00 00 00 00 00 02 12 28 14 40 00 00 00 00 00	@.....(.@.....
00000910	04 4C 65 66 74 02 00 12 28 14 40 00 00 00 00 00	.Left.....(.@.....
00000920	05 52 69 67 68 74 02 00 02 00 09 08 B3 93 00 00	.Right.....".."
00000930	00 00 00 05 45 6D 70 74 79 00 00 28 14 40 00 00	...Empty...(.@..
00000940	00 00 00 00 02 00 00 00 50 15 40 00 00 00 00 00P.@.....
00000950	14 0F 50 49 6E 74 65 72 66 61 63 65 45 6E 74 72	.PInterfaceEntr
00000960	79 70 15 40 00 00 00 00 00 02 00 00 00 00 00 00	yp.@.....
00000970	78 15 40 00 00 00 00 00 0E 0F 54 49 6E 74 65 72	x.@.....TInter
00000980	66 61 63 65 45 6E 74 72 79 28 00 00 00 00 00 00	faceEntry.....
00000990	00 00 05 00 00 00 28 14 40 00 00 00 00 00 00 00(.@.....
000009A0	00 00 00 00 00 02 03 49 49 44 02 00 18 11 40IID....@
000009B0	00 00 00 00 00 10 00 00 00 00 00 00 00 02 06 56	V ..

Offset(h): 0

Special editors

Data inspector

Binary (8 bit) 01001101

Int8 go to: 77

UInt8 go to: 77

Int16 go to: 23117

UInt16 go to: 23117

Int24 go to: 5265997

UInt24 go to: 5265997

Int32 go to: 5265997

UInt32 go to: 5265997

Int64 go to: 8595200589

UInt64 go to: 8595200589

LEB128 go to: -51

ULEB128 go to: 77

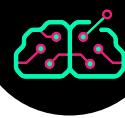
AnsiChar / char8 M

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)

Overwrite

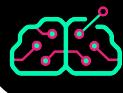


Usando o HxD - encoding



Usando o HxD - encoding

The screenshot shows the HxD hex editor interface. The main window displays the file 'readme.txt' with its content in ASCII format. A context menu is open over the file name, showing options like 'Text encoding', 'Windows (ANSI)', 'DOS/IBM-ASCII (OEM)', 'Macintosh', and 'EBCDIC'. Other menu items include 'File', 'Edit', 'Search', 'View', 'Analysis', 'Tools', 'Window', and 'Help'. A toolbar at the top includes icons for file operations like Open, Save, and Print. On the right side, there's a 'Special editors' panel with a 'Data inspector' tab showing binary data as 8-bit values. Below it is a list of data types with their corresponding go-to addresses: Int8, UInt8, Int16, UInt16, Int24, UInt24, Int32, UInt32, Int64, UInt64, LEB128, ULEB128, and AnsiChar / char8. At the bottom, there are settings for 'Byte order' (radio buttons for 'Little endian' and 'Big endian') and a checkbox for 'Hexadecimal basis (for integral numbers)'.



Usando o HxD - encoding

HxD - [C:\Users\admin\AppData\Local\Programs\retookit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

readme.txt HxD.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
000B8730	04 53 65 6C 66 02 00 0A E0 24 40 00 00 00 00 00	.Self...à\$@.....
000B8740	00 00 04 49 74 65 6D 02 00 02 00 5F 00 A0 72 4E	...Item...._.rN
000B8750	00 00 00 00 00 0A 52 65 6D 6F 76 65 49 74 65 6DRemoveItem
000B8760	03 00 B8 10 40 00 00 00 00 28 00 03 08 F0 94	...,@....(....δ"
000B8770	4B 00 00 00 00 00 00 00 04 53 65 6C 66 02 00 0A	K.....Self...
000B8780	E0 24 40 00 00 00 00 00 00 04 49 74 65 6D 02	à\$@.....Item.
000B8790	00 00 B0 64 41 00 00 00 00 00 00 09 44 69 72	..°dA.....Dir
000B87A0	65 63 74 69 6F 6E 02 00 02 00 30 00 D0 74 4E 00	ection....0.BtN.
000B87B0	00 00 00 00 04 4C 6F 63 6B 03 00 00 00 00 00 00Lock.....
000B87C0	00 00 00 18 00 01 08 F0 94 4B 00 00 00 00 00 008"K.....
000B87D0	00 04 53 65 6C 66 02 00 02 00 32 00 F0 74 4E 00	0.Self....2.δtN.
000B87E0	00 00 00 00 06 55 6E 6C 6F 63 6B 03 00 00 00 00Unlock....
000B87F0	00 00 00 00 00 18 00 01 08 F0 94 4B 00 00 00 008"K.....
000B8800	00 00 00 04 53 65 6C 66 02 00 02 00 39 00 D0 6ESelf....9.Dn
000B8810	4E 00 00 00 00 00 OD 47 65 74 45 6E 75 6D 65 72	N.....GetEnume
000B8820	61 74 6F 72 03 00 A8 8B 4B 00 00 00 00 00 18 00	ator.. <k.....< td=""></k.....<>
000B8830	01 08 F0 94 4B 00 00 00 00 00 00 04 53 65 6C	.8"K.....Sel
000B8840	66 02 00 02 00 51 00 C0 6C 4E 00 00 00 00 00 03	f....Q.ÀlN.....
000B8850	47 65 74 03 00 E0 24 40 00 00 00 00 28 00 03	Get..à\$@....(.
000B8860	08 F0 94 4B 00 00 00 00 00 00 04 53 65 6C 66	.8"K.....Self
000B8870	02 00 00 B8 10 40 00 00 00 00 00 00 05 49 6E@.....In
000B8880	64 65 78 02 00 48 E0 24 40 00 00 00 00 00 00 00	dex..Ha\$@.....
000B8890	01 01 02 00 02 00 54 00 70 71 4E 00 00 00 00 00T.pqN.....
000B88A0	03 50 75 74 03 00 00 00 00 00 00 00 00 28 00	.Put.....(.

Offset(h): B87D0

Special editors x

Data inspector

Binary (8 bit) 00000000

Int8 go to: 0

UInt8 go to: 0

Int16 go to: 1024

UInt16 go to: 1024

Int24 go to: 5440512

UInt24 go to: 5440512

Int32 go to: 1699939328

UInt32 go to: 1699939328

Int64 go to: 675565695861760

UInt64 go to: 675565695861760

LEB128 go to: 0

ULEB128 go to: 0

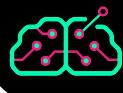
AnsiChar / char8:

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)

Overwrite



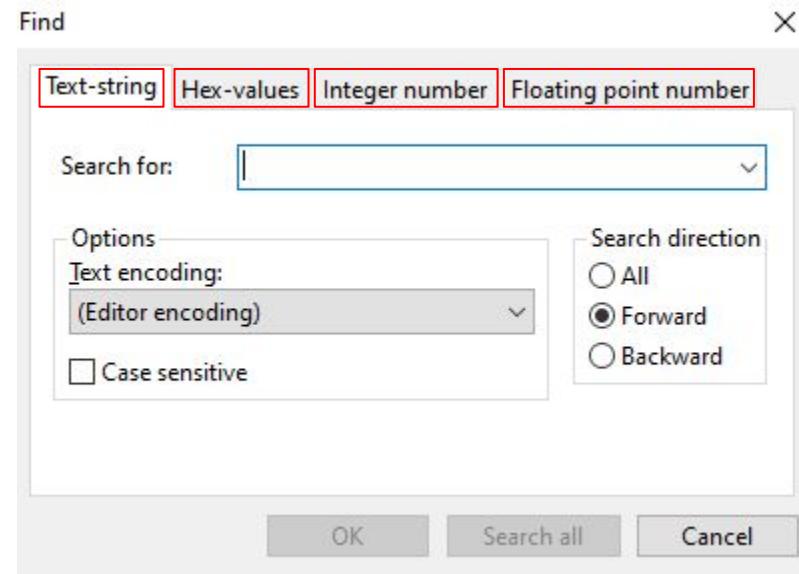
Usando o HxD - Busca

The screenshot shows the HxD hex editor interface. The menu bar is visible at the top, with the 'Search' option highlighted and a red box drawn around it. A sub-menu is open under 'Search' containing options: 'Find' (Ctrl+F), 'Replace...', 'Find again', 'Find again (reversed)', and 'Go to...'. The main window displays a hex dump of a file named 'HxD.exe'. The left pane shows the offset in hexadecimal (Offset(h)) from 00000000 to 00000170. The right pane shows the raw bytes and their corresponding decoded text. A 'Special editors' panel is open on the right, titled 'Data inspector'. It contains a table for 'Binary (8 bit)' with various data types and their addresses. At the bottom of the 'Data inspector' panel, there are buttons for 'Byte order' (radio buttons for 'Little endian' and 'Big endian') and a checkbox for 'Hexadecimal basis (for integral numbers)'. The status bar at the bottom shows 'Offset(h): 0' and 'Overwrite'.

Binary (8 bit)	01001101
Int8	go to: 77
UInt8	go to: 77
Int16	go to: 23117
UInt16	go to: 23117
Int24	go to: 5265997
UInt24	go to: 5265997
Int32	go to: 5265997
UInt32	go to: 5265997
Int64	go to: 8595200589
UInt64	go to: 8595200589
LEB128	go to: -51
ULEB128	go to: 77
AnsiChar / char8	M

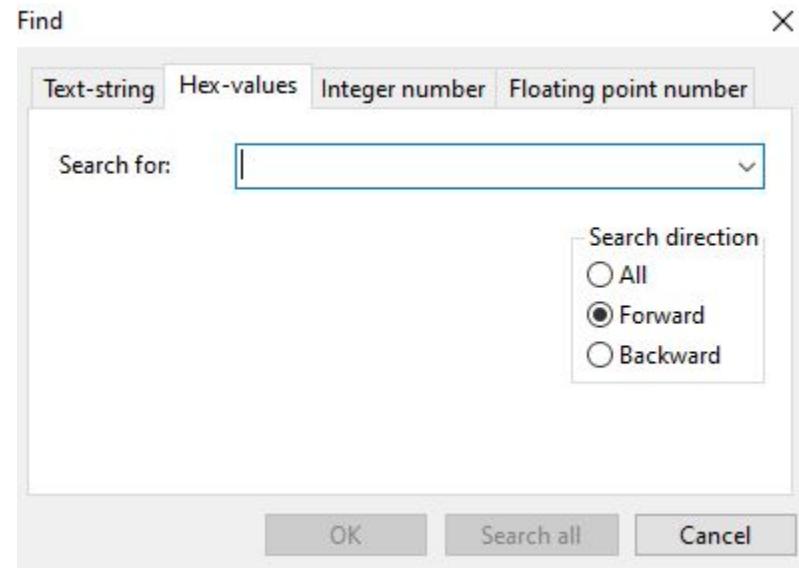


Usando o HxD - Busca





Usando o HxD - Busca hexa por string “pro”



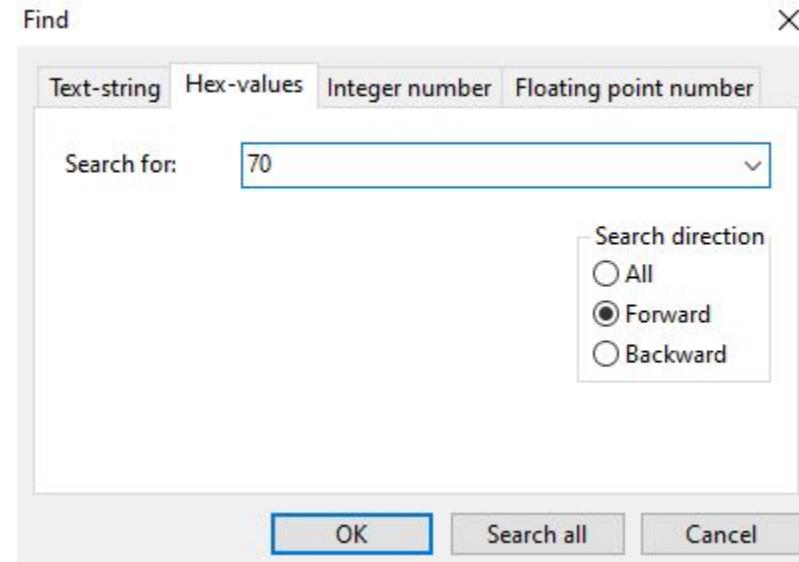


Usando o HxD - Busca hexa por string “pro”

00	nul	01	soh	02	stx	03	etx	04	eot	05	enq	06	ack	07	bel
08	bs	09	ht	0a	nl	0b	vt	0c	np	0d	cr	0e	so	0f	si
10	dle	11	dc1	12	dc2	13	dc3	14	dc4	15	nak	16	syn	17	etb
18	can	19	em	1a	sub	1b	esc	1c	fs	1d	gs	1e	rs	1f	us
20	sp	21	!	22	"	23	#	24	\$	25	%	26	&	27	'
28	(29)	2a	*	2b	+	2c	,	2d	-	2e	.	2f	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3a	:	3b	;	3c	<	3d	=	3e	>	3f	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4a	J	4b	K	4c	L	4d	M	4e	N	4f	0
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5a	Z	5b	[5c	\	5d]	5e	^	5f	_
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6a	j	6b	k	6c	l	6d	m	6e	n	6f	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7a	z	7b	{	7c		7d	}	7e	~	7f	del



Usando o HxD - Busca hexa por string “pro”



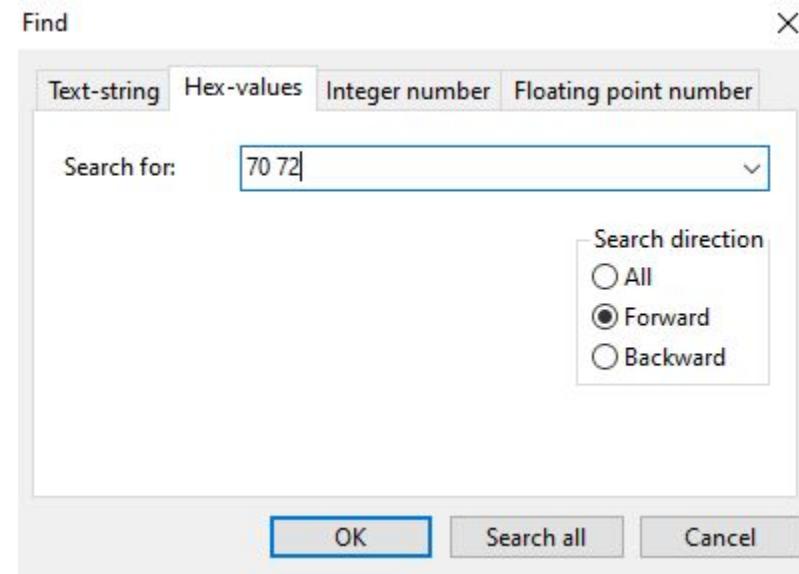


Usando o HxD - Busca hexa por string “pro”

00	nul	01	soh	02	stx	03	etx	04	eot	05	enq	06	ack	07	bel
08	bs	09	ht	0a	nl	0b	vt	0c	np	0d	cr	0e	so	0f	si
10	dle	11	dc1	12	dc2	13	dc3	14	dc4	15	nak	16	syn	17	etb
18	can	19	em	1a	sub	1b	esc	1c	fs	1d	gs	1e	rs	1f	us
20	sp	21	!	22	"	23	#	24	\$	25	%	26	&	27	'
28	(29)	2a	*	2b	+	2c	,	2d	-	2e	.	2f	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3a	:	3b	;	3c	<	3d	=	3e	>	3f	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4a	J	4b	K	4c	L	4d	M	4e	N	4f	0
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5a	Z	5b	[5c	\	5d]	5e	^	5f	_
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6a	j	6b	k	6c	l	6d	m	6e	n	6f	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7a	z	7b	{	7c		7d	}	7e	~	7f	del



Usando o HxD - Busca hexa por string “pro”



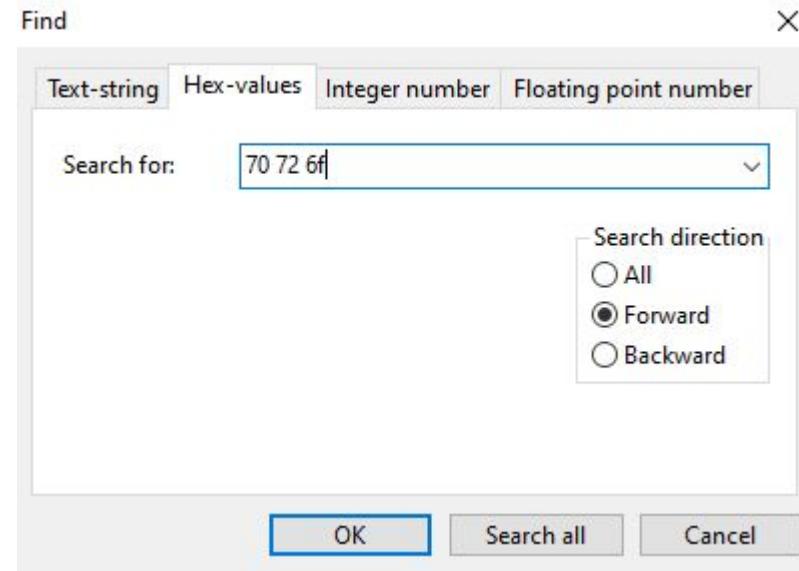


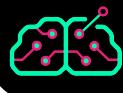
Usando o HxD - Busca hexa por string “pro”

00	nul	01	soh	02	stx	03	etx	04	eot	05	enq	06	ack	07	bel
08	bs	09	ht	0a	nl	0b	vt	0c	np	0d	cr	0e	so	0f	si
10	dle	11	dc1	12	dc2	13	dc3	14	dc4	15	nak	16	syn	17	etb
18	can	19	em	1a	sub	1b	esc	1c	fs	1d	gs	1e	rs	1f	us
20	sp	21	!	22	"	23	#	24	\$	25	%	26	&	27	'
28	(29)	2a	*	2b	+	2c	,	2d	-	2e	.	2f	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3a	:	3b	;	3c	<	3d	=	3e	>	3f	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4a	J	4b	K	4c	L	4d	M	4e	N	4f	0
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5a	Z	5b	[5c	\	5d]	5e	^	5f	_
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6a	j	6b	k	6c	l	6d	m	6e	n	6f	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7a	z	7b	{	7c		7d	}	7e	~	7f	del



Usando o HxD - Busca hexa por string “pro”





Usando o HxD - Busca hexa por string “pro”

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

HxD.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000000	4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00	MZP.....ÿÿ..
00000010	B8 00 00 00 00 00 00 40 00 1A 00 00 00 00 00 00@.....
00000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00000040	BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90	°.....í!,Lí!..
00000050	54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73	This program mus
00000060	74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57	t be run under W
00000070	69 6E 36 34 0D 0A 24 37 00 00 00 00 00 00 00 00	in64..\$7.....
00000080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000E0	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
00000100	50 45 00 00 64 86 09 00 F9 4D 24 60 00 00 00 00	PE..dt..ùM\$`....
00000110	00 00 00 00 F0 00 23 00 0B 02 08 00 00 6E 53 00	...\$.#.....ns.
00000120	00 EE 15 00 00 00 00 30 7B 53 00 00 10 00 00	.i.....0{s.....
00000130	00 00 40 00 00 00 00 00 10 00 00 00 02 00 00	...@.....
00000140	05 00 01 00 05 00 02 00 05 00 01 00 00 00 00 00
00000150	00 80 6A 00 00 04 00 00 CE 95 69 00 02 00 40 01	.ej.....í*i...@.
00000160	00 00 10 00 00 00 00 00 40 00 00 00 00 00 00 00@.....
00000170	00 00 10 00 00 00 00 00 20 00 00 00 00 00 00 00

Offset(h): 55 Block(h): 55-57 Length(h): 3 Overwrite

Special editors x

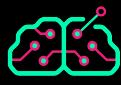
Data inspector

Binary (8 bit)	01110000
Int8	go to: 112
UInt8	go to: 112
Int16	go to: 29296
UInt16	go to: 29296
Int24	go to: 7303792
UInt24	go to: 7303792
Int32	go to: Invalid
UInt32	go to: Invalid
Int64	go to: Invalid
UInt64	go to: Invalid
LEB128	go to: -16
ULEB128	go to: 112
AnsiChar / char8_p	

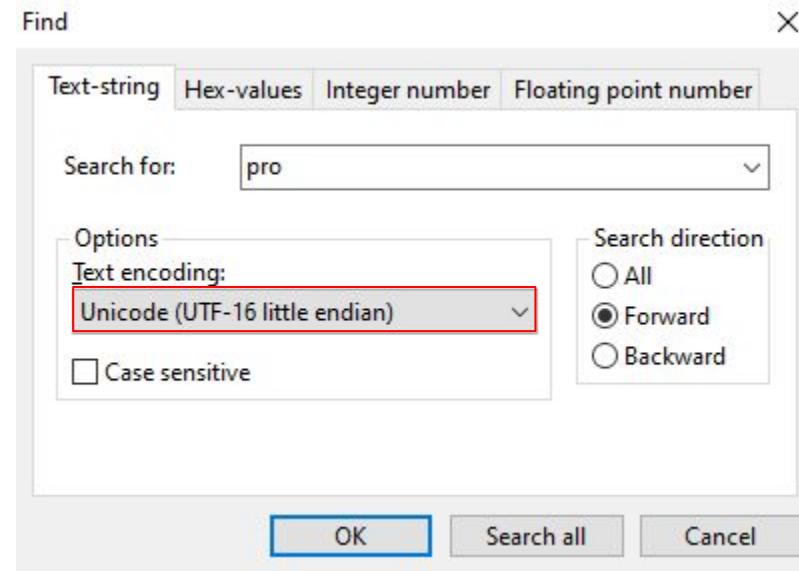
Byte order

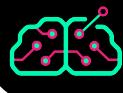
Little endian Big endian

Hexadecimal basis (for integral numbers)



Usando o HxD - Busca por string “pro”





Usando o HxD - Busca por string “pro”

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text

0001C080	C7 C0 C5 00 00 00 E8 A5 1E FF FF E8 10 D7 FF FF	ÇÀ...èÿ..ÿè..ÿÿ
0001C090	48 89 C1 48 33 D2 4D 33 C0 E8 32 D6 FF FF 48 8D	H»ÄH3ÖM3Àè2ÖÿÿH.
0001C0A0	65 60 5D C3 48 8D 40 00 48 8D 04 05 00 00 00 00	e`]ÄH..@.H.....
0001C0B0	55 48 83 EC 20 48 8B EC 48 8D 44 2A 58 48 8B CD	UHfi H<îH.D*XH<í
0001C0C0	48 F7 D9 48 8B OC 08 48 8D 44 2A 54 4C 8B C5 49	H-ÜH<..H.D*T _L <ÅI
0001C0D0	F7 D8 4A 63 04 00 4C 8D 44 2A 50 4C 8B CD 49 F7	±ØJc..L.D*PL<ÍI±
0001C0E0	D9 47 8B 04 08 48 8D 54 2A 4C 4C 8B CD 49 F7 D9	ÜG<..H.T*LL<ÍI±Ù
0001C0F0	4E 8D OC 0A 48 89 C2 E8 E4 DD FF FF 85 C0 75 1A	N...HtÀæaÿÿ..Au.
0001C100	48 8D 0D 25 00 00 00 48 8D 15 56 00 00 00 41 C7	H..%...H..V...AC
0001C110	C0 C5 00 00 00 E8 16 1E FF FF 48 8D 65 20 5D C3	À...è..ÿYH.e]À
0001C120	B0 04 02 00 FF FF FF FF 15 00 00 00 56 00 69 00	°...ÿÿÿ...V.i.
0001C130	72 00 74 00 75 00 61 00 6C 00 50 00 72 00 6F 00	r.t.u.a.l.P.r.o.
0001C140	74 00 65 00 63 00 74 00 20 00 66 00 61 00 69 00	t.e.c.t. .f.a.i.
0001C150	6C 00 65 00 64 00 00 00 B0 04 02 00 FF FF FF FF	l.e.d.°...ÿÿÿ
0001C160	31 00 00 00 44 00 3A 00 5C 00 51 00 75 00 65 00	l...D..\\Q.u.e.
0001C170	6C 00 6C 00 65 00 6E 00 5C 00 4B 00 6F 00 6D 00	l.l.e.n.\\K.o.m.
0001C180	70 00 6F 00 6E 00 65 00 6E 00 74 00 65 00 6E 00	p.o.n.e.n.t.e.n.
0001C190	5C 00 58 00 6D 00 4D 00 69 00 73 00 63 00 5C 00	\.X.m.M.i.s.c.\.
0001C1A0	53 00 6F 00 75 00 72 00 63 00 65 00 5C 00 58 00	S.o.u.r.c.e.\.X.
0001C1B0	6D 00 53 00 79 00 73 00 74 00 65 00 6D 00 2E 00	m.S.y.s.t.e.m...
0001C1C0	70 00 61 00 73 00 00 00 CC CC CC CC CC CC CC CC	p.a.s....fffffiiii
0001C1D0	55 48 83 EC 30 48 8B EC 48 89 4D 40 48 8B 45 40	UHfi0H<îH»M@H<E@
0001C1E0	48 83 38 00 0F 84 DD 00 00 00 48 8B 45 40 48 8B	Hf8...Í...H<E@H<
0001C1F0	49 09 FF 50 52 FF FF 49 09 4D 40 49 FF 09 49 09	H ÁVDH@H@H@H@H@H@H@

Offset(h): 1C13A Block(h): 1C13A-1C13F Length(h): 6 Overwrite

Special editors

Data inspector

Binary (8 bit) 01010000

Int8 go to: 80

UInt8 go to: 80

Int16 go to: 80

UInt16 go to: 80

Int24 go to: 7471184

UInt24 go to: 7471184

Int32 go to: 7471184

UInt32 go to: 7471184

Int64 go to: Invalid

UInt64 go to: Invalid

LEB128 go to: -48

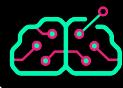
ULEB128 go to: 80

AnsiChar / char8_P P

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Usando o HxD - Busca por string “pro”

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) dec

HxD.exe

Offset(d)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0G 0H 0I 0J 0K 0L 0M 0N 0O 0P 0Q 0R 0S 0T 0U 0V 0W 0X 0Y 0Z	Decoded text
00114816	C7 C0 C5 00 00 00 E8 A5 1E FF FF E8 10 D7 FF FF	ÇÀ...è¥.ÿyè.xÿ
00114832	48 89 C1 48 33 D2 4D 33 C0 E8 32 D6 FF FF 48 8D	HÁH3ÒM3Àè2ÖÿyH.
00114848	65 60 5D C3 48 8D 40 00 48 8D 04 05 00 00 00 00	e]ÀH.@.H.....
00114864	55 48 83 EC 20 48 8B EC 48 8D 44 2A 54 58 48 8B CD	UHfi H <i>c</i> iH.D*XHí
00114880	48 F7 D9 48 8B OC 08 48 8D 44 2A 54 48 8B C5 49	H-ÜH<...H.D*TlÅÍ
00114896	F7 D8 4A 63 04 00 4C 8D 44 2A 50 4C 8B CD 49 F7	+ØJc..L.D*PL<ÍI=
00114912	D9 47 8B 04 08 48 8D 54 2A 4C 4C 8B CD 49 F7 D9	ÜG<...H.T*LL<ÍI=Ü
00114928	4E 8D 0C 0A 48 89 C2 E8 E4 DD FF FF 85 C0 75 1A	N...HtÀeaÿy...Au.
00114944	48 8D OD 25 00 00 00 48 8D 15 56 00 00 00 41 C7	H..%...H.V...Ac
00114960	C0 C5 00 00 00 E8 16 1E FF FF 48 8D 65 20 5D C3	À...è...ÿyH.e]À
00114976	B0 04 02 00 FF FF FF FF 15 00 00 00 56 00 69 00	°...ÿÿÿ...V.i.
00114992	72 00 74 00 75 00 61 00 6C 00 50 00 72 00 6F 00	r.t.u.a.l.F.r.o.
00115008	74 00 65 00 63 00 74 00 20 00 66 00 61 00 69 00	t.e.c.t. .f.a.i.
00115024	6C 00 65 00 64 00 00 B0 04 02 00 FF FF FF FF	l.e.d.°...ÿÿÿ
00115040	31 00 00 00 44 00 3A 00 5C 00 51 00 75 00 65 00	1...D...\Q.u.e.
00115056	6C 00 6C 00 65 00 6E 00 5C 00 4B 00 6F 00 6D 00	1.l.e.n.\K.o.m.
00115072	70 00 6F 00 6E 00 65 00 6E 00 74 00 65 00 6E 00	p.o.n.e.n.t.e.n.
00115088	5C 00 58 00 6D 00 4D 00 69 00 73 00 63 00 5C 00	\.X.m.M.i.s.c.\.
00115104	53 00 6F 00 75 00 72 00 63 00 65 00 5C 00 58 00	S.o.u.r.c.e.\.X.
00115120	6D 00 53 00 79 00 73 00 74 00 65 00 6D 00 2E 00	m.S.y.s.t.e.m...
00115136	70 00 61 00 73 00 00 00 CC CC CC CC CC CC CC	p.a.s...iiiiiiii
00115152	55 48 83 EC 30 48 8B EC 48 89 4D 40 48 8B 45 40	UHfiOH <i>c</i> HwM@HxE@
00115168	48 83 38 00 0F 84 DD 00 00 00 48 8B 45 40 48 8B	Hf8...Ý...HxE@Hx
00115184	48 08 F8 F0 F2 FF FF 48 8B 4D 40 48 8B 00 48 80	H_AVDGHHH.MAHx_Hx

Offset(d): 115002 Block(d): 115002-115031 Length(d): 30 Overwrite

Special editors

Data inspector

Binary (8 bit) 01010000

Int8 go to: 80

UInt8 go to: 80

Int16 go to: 80

UInt16 go to: 80

Int24 go to: 7471184

UInt24 go to: 7471184

Int32 go to: 7471184

UInt32 go to: 7471184

Int64 go to: 326515740472771:

UInt64 go to: 326515740472771:

LEB128 go to: -48

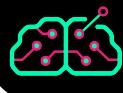
ULEB128 go to: 80

AnsiChar / char8_P

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Usando o HxD - Busca por inteiro

HxD - [C:\Users\admin\AppData\Local\Programs\reto toolkit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

HxD.exe

Offset(h)	Decoded text
00000060	t be run under W
00000070	in64..\$7.....
00000080
00000090
000000A0
000000B0
000000C0
000000D0
000000E0
000000F0
00000100	PE..dt..ùMS`.....
000001108.#.....nS.
00000120	.i.....0{S.....
00000130	..@.....
00000140
00000150	€j.....í•H...@.
00000160@.....
00000170
00000180
00000190	Ð`..8Y...0e..n..
000001A0	..`a. Á.....
000001B0
000001C0
000001D0

Offset(h): 15A

Special editors

Data inspector

Binary (8 bit) 01101001

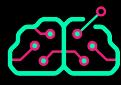
Int8	go to: 105
UInt8	go to: 105
Int16	go to: 105
UInt16	go to: 105
Int24	go to: 131177
UInt24	go to: 131177
Int32	go to: 131177
UInt32	go to: 131177
Int64	go to: 1374389665897
UInt64	go to: 1374389665897
LEB128	go to: -23
ULEB128	go to: 105
AnsiChar / char8_t	i

Byte order

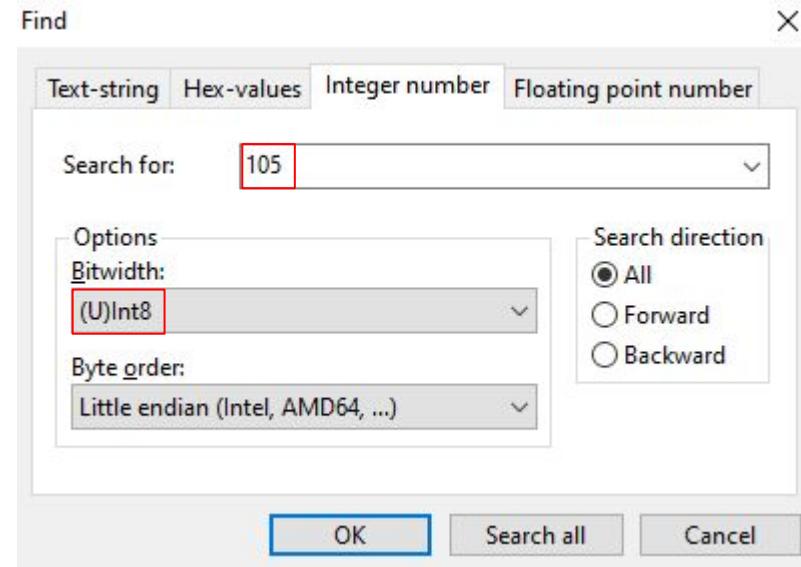
Little endian Big endian

Hexadecimal basis (for integral numbers)

Overwrite



Usando o HxD - Busca por inteiro





Usando o HxD - Busca por inteiro

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

HxD.exe

Offset(h)	Decoded text
00000000	MZP.....ÿ..
00000010	,.....@.....
00000020
00000030
00000040	°.....Í!,..LÍ!..
00000050	This program mus
00000060	t be run under W
00000070	in64..\$7.....
00000080
00000090
000000A0
000000B0
000000C0
000000D0
000000E0
000000F0
00000100	PE..dt..üMS`....
00000110\$.#..nS.
00000120	.i.....0{S..
00000130	..@.....
00000140
00000150	.€j.....í•i...@.
00000160@.....
00000170

Special editors

Data inspector

Binary (8 bit) 01101001

Int8	go to:	105
UInt8	go to:	105
Int16	go to:	Invalid
UInt16	go to:	Invalid
Int24	go to:	Invalid
UInt24	go to:	Invalid
Int32	go to:	Invalid
UInt32	go to:	Invalid
Int64	go to:	Invalid
UInt64	go to:	Invalid
LEB128	go to:	-23
ULEB128	go to:	105
AnsiChar / char8_i	go to:	

Byte order

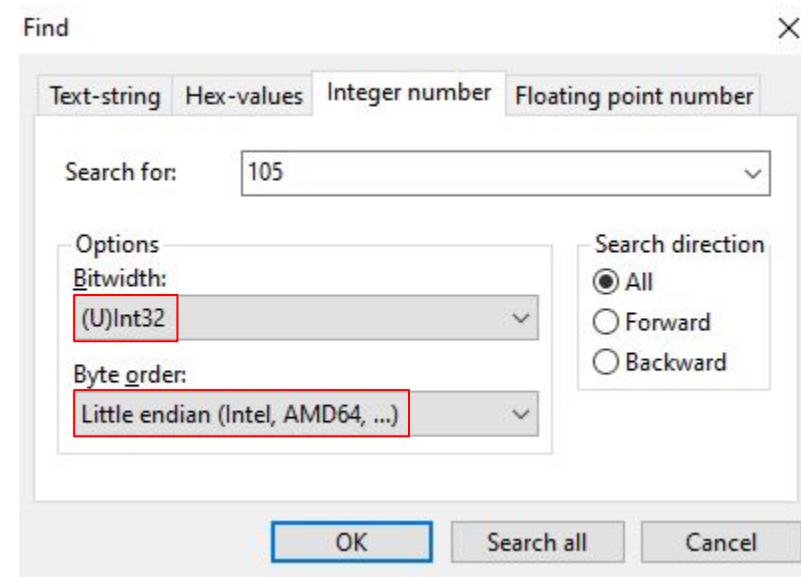
Little endian Big endian

Hexadecimal basis (for integral numbers)

Offset(h): 52 Block(h): 52-52 Length(h): 1 Overwrite

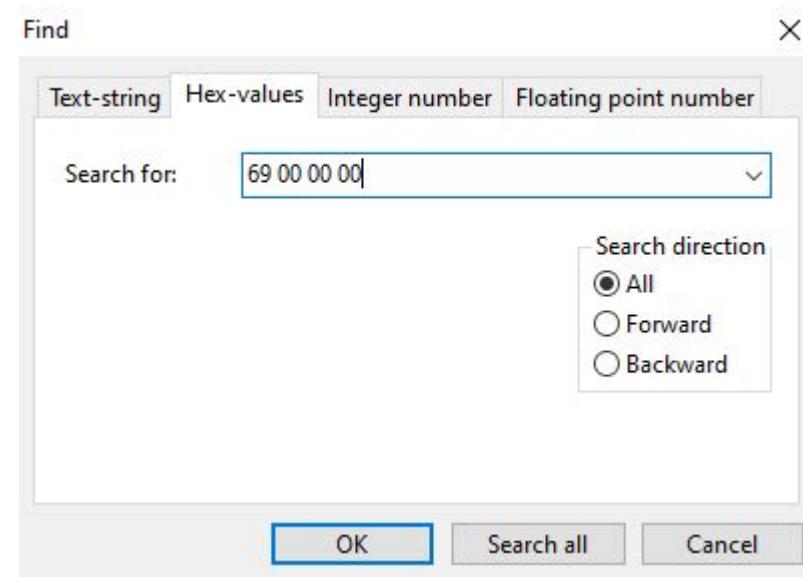


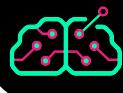
Usando o HxD - Busca por inteiro





Usando o HxD - Busca por inteiro





Usando o HxD - Busca por inteiro

HxD - [C:\Users\admin\AppData\Local\Programs\retookit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

Special editors

Data inspector

Binary (8 bit) 01101001

Type	Value	Description
Int8	go to: 105	
UInt8	go to: 105	
Int16	go to: 105	
UInt16	go to: 105	
Int24	go to: 105	
UInt24	go to: 105	
Int32	go to: 105	
UInt32	go to: 105	
Int64	go to: Invalid	
UInt64	go to: Invalid	
LEB128	go to: -23	
ULEB128	go to: 105	
AnsiChar / char8_i	01101001	

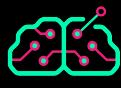
Byte order

Little endian Big endian

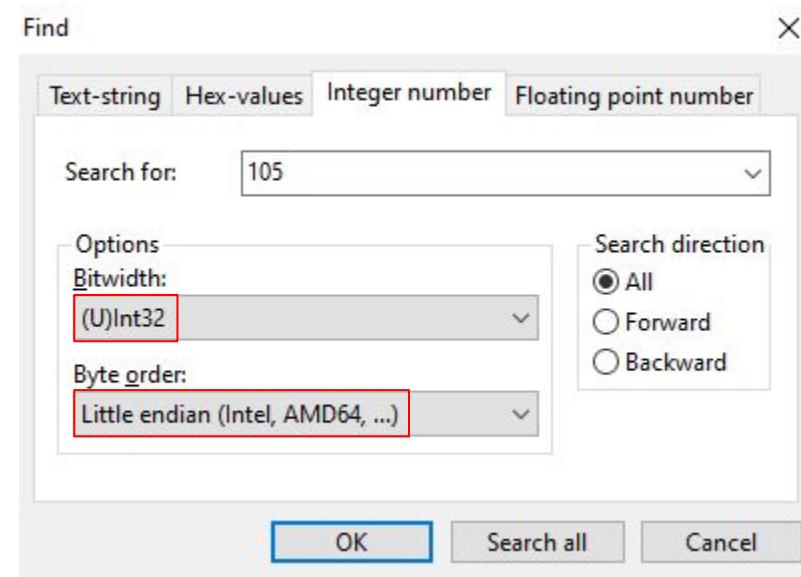
Hexadecimal basis (for integral numbers)

Offset(h): 386 Block(h): 386-389 Length(h): 4 Overwrite

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
000002D0	2E 74 6C 73 00 00 00 00 CC 02 00 00 00 40 61 00	.tls....i....@a.
000002E0	00 00 00 00 00 52 60 00 00 00 00 00 00 00 00 00R'.....
000002F0	00 00 00 00 00 00 00 C0 2E 72 64 61 74 61 00 00Â.rdata..
00000300	28 00 00 00 00 50 61 00 00 02 00 00 00 52 60 00	(....Pa.....R`.
00000310	00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 40@..@
00000320	2E 70 64 61 74 61 00 00 20 C1 03 00 00 60 61 00	.pdata.. Á...a.
00000330	00 C2 03 00 00 54 60 00 00 00 00 00 00 00 00 00	Â...T`.....
00000340	00 00 00 00 40 00 00 40 2E 72 73 72 63 00 00 00@.@.rsrc...
00000350	94 48 05 00 00 30 65 00 00 4A 05 00 00 16 64 00	"H...0e..J...d.
00000360	00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 40@..@
00000370	00 00 00 00 00 00 00 00 00 00 00 00 00 A0 6A 00 j.
00000380	00 00 00 00 00 84 69 00 00 00 00 00 00 00 00 00L...
00000390	00 00 00 00 40 00 00 40 00 00 00 00 00 00 00 00@.0...
000003A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000400	08 10 40 00 00 00 00 00 03 07 42 6F 6C 65 61	...@.....Boolean
00000410	6E 01 00 00 00 00 01 00 00 00 00 10 40 00 00 00	n.....@...
00000420	00 00 05 46 61 6C 73 65 04 54 72 75 65 06 53 79	...False.True.Sy
00000430	73 74 65 6D 02 00 00 00 40 10 40 00 00 00 00 00	stem....@.0....
00000440	02 08 41 FF 73 60 43 FF 61 72 01 00 00 00 00 FF	AnsiChar

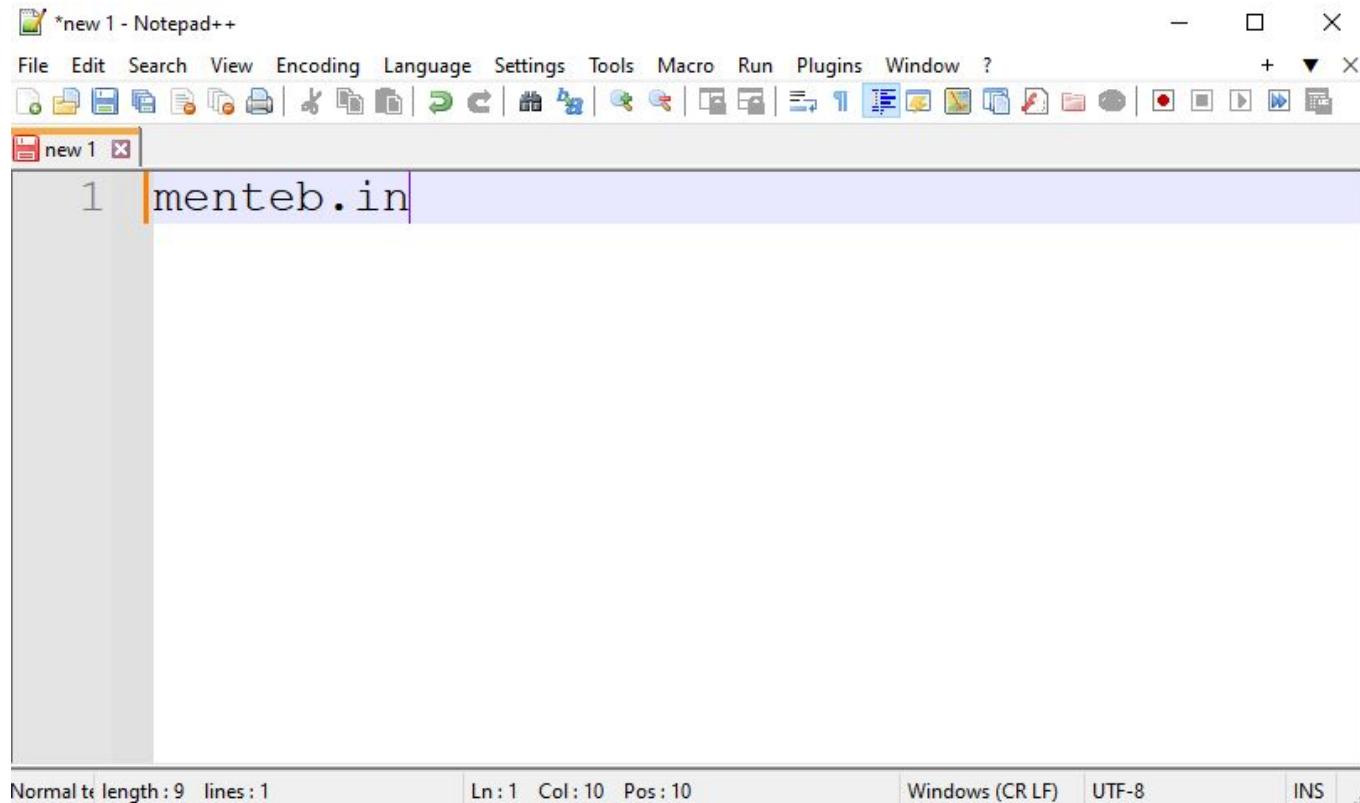


Usando o HxD - Busca por inteiro





Usando o HxD - Editar e salvar





Usando o HxD - Editar e salvar

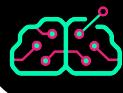
The screenshot shows the HxD hex editor interface. The main window displays the file 'new1.txt' with the following hex dump:

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000000	6D 65 6E 74 65 62 2E 69 6E	mentebinaria.com.br

The 'Special editors' panel on the right lists various data types, all of which are currently invalid:

Binary (8 bit)	Invalid
Int8	go to: Invalid
UInt8	go to: Invalid
Int16	go to: Invalid
UInt16	go to: Invalid
Int24	go to: Invalid
UInt24	go to: Invalid
Int32	go to: Invalid
UInt32	go to: Invalid
Int64	go to: Invalid
UInt64	go to: Invalid
LEB128	go to: Invalid
ULEB128	go to: Invalid
AnsiChar / char8	go to: Invalid

At the bottom, there are options for 'Byte order' (radio buttons for 'Little endian' and 'Big endian') and a checkbox for 'Hexadecimal basis (for integral numbers)'.



Usando o HxD - Editar e salvar

HxD - [C:\Users\admin\Desktop\new1.txt]

File Edit Search View Analysis Tools Window Help

new1.txt

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000000	6D 65 6E 74 65 63 2E 69 6E	mentec.in

Special editors

Data inspector

Binary (8 bit) Invalid

- Int8 [go to:](#) Invalid
- UInt8 [go to:](#) Invalid
- Int16 [go to:](#) Invalid
- UInt16 [go to:](#) Invalid
- Int24 [go to:](#) Invalid
- UInt24 [go to:](#) Invalid
- Int32 [go to:](#) Invalid
- UInt32 [go to:](#) Invalid
- Int64 [go to:](#) Invalid
- UInt64 [go to:](#) Invalid
- LEB128 [go to:](#) Invalid
- ULEB128 [go to:](#) Invalid
- AnsiChar / char8 [go to:](#) Invalid

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)

Offset(h): 9 * Modified * Overwrite



Usando o HxD - Editar e salvar

HxD - [C:\Users\admin\Desktop\new1.txt]

File Edit Search View Analysis Tools Window Help

new1.txt

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text

00000000 6D 64 6E 74 65 63 2E 69 6E mdntec.in

Special editors x

Data inspector

Binary (8 bit) 01101110

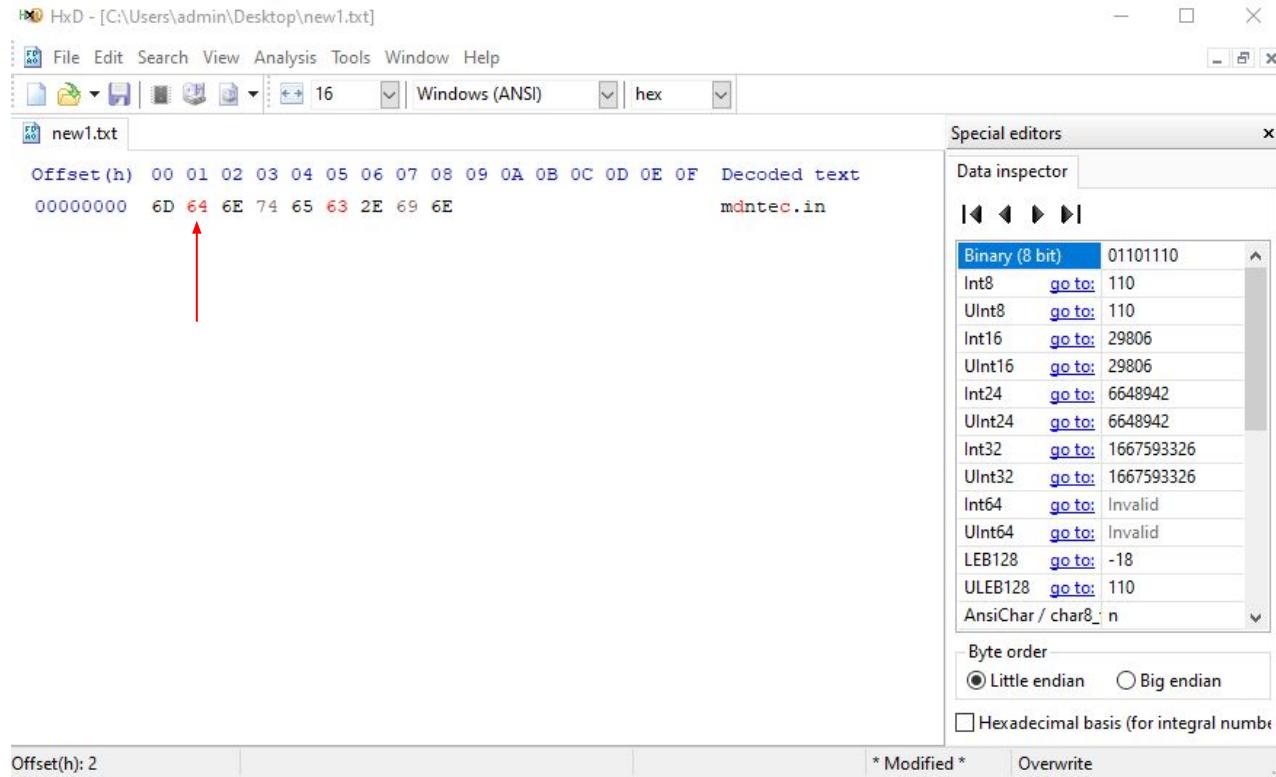
Int8	go to:	110
UInt8	go to:	110
Int16	go to:	29806
UInt16	go to:	29806
Int24	go to:	6648942
UInt24	go to:	6648942
Int32	go to:	1667593326
UInt32	go to:	1667593326
Int64	go to:	Invalid
UInt64	go to:	Invalid
LEB128	go to:	-18
ULEB128	go to:	110
AnsiChar / char8	n	

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)

Offset(h): 2 * Modified * Overwrite





Usando o HxD - Editar e salvar

HxD - [C:\Users\admin\Desktop\new1.txt]

File Edit Search View Analysis Tools Window Help

new1.txt

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000000	6D 64 6E 74 65 63 2E 69 6E	mdntec.in

Special editors

Data inspector

Binary (8 bit)	01101110
Int8	go to: 110
UInt8	go to: 110
Int16	go to: 29806
UInt16	go to: 29806
Int24	go to: 6648942
UInt24	go to: 6648942
Int32	go to: 1667593326
UInt32	go to: 1667593326
Int64	go to: Invalid
UInt64	go to: Invalid
LEB128	go to: -18
ULEB128	go to: 110
AnsiChar / char8_t	n

Byte order

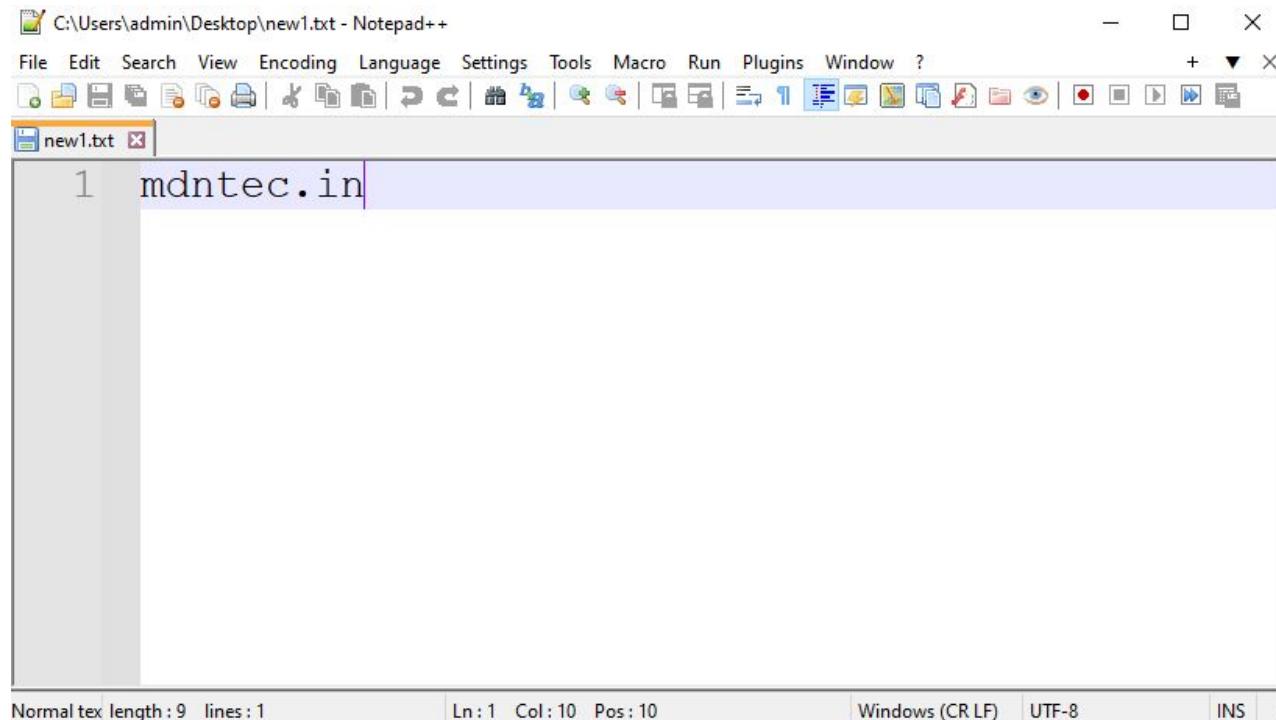
Little endian Big endian

Hexadecimal basis (for integral numbers)

Offset(h): 2 * Modified * Overwrite



Usando o HxD - Editar e salvar





Lab 05

Manipulando strings



Lab 05 - Manipulando strings

1. Abra o programa **strings1.exe** no HxD e responda:

- a) Em que offset está a string “ELFParser-NG”?
- b) Que tipo de string é esta?

2. Execute o programa **strings2.exe** e veja a string que ele exibe. Agora feche-o, abra-o no HxD e altere a string de forma que o programa exiba o seu primeiro nome (respeite o limite de caracteres da string atual).



Lab 05 - Manipulando strings - Respostas

1.

HxD - [C:\Users\admin\Desktop\binarios\strings1.exe]

File Edit Search View Analysis Tools Window Help

strings1.exe

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	5A	90	00	03	00											
00000010	B8	00	00	00	00	00											
00000020	00	00	00	00	00	00											
00000030	00	00	00	00	00	00											
00000040	0E	1F	BA	0E	00	B4											
00000050	69	73	20	70	72	6F											
00000060	74	20	62	65	20	72											
00000070	6D	6F	64	65	2E	0D											
00000080	86	B0	6A	16	C2	D1											
00000090	CB	A9	97	45	C8	D1											
000000A0	A2	AB	01	44	D1	D1											
000000B0	A2	AB	07	44	C3	D1											
000000C0	C2	D1	05	45	EC	D1											
000000D0	A6	AB	FB	45	C3	D1											
000000E0	52	69	63	68	C2	D1											
000000F0	50	45	00	00	4C	01											
00000100	00	00	00	00	E0	00											
00000110	00	14	00	00	00	00											
00000120	00	20	00	00	00	00	40	00	00	10	00	00	02	00	00	00@.....
00000130	06	00	00	00	00	00	00	00	06	00	00	00	00	00	00	00
00000140	00	60	00	00	00	04	00	00	00	00	00	03	00	40	81@.....	
00000150	00	00	10	00	00	10	00	00	00	00	10	00	00	10	00	00
00000160	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
00000170	3C	26	00	00	70	00	00	00	00	40	00	00	50	01	00	00

Offset(h): 0

Find

Text-string Hex-values Integer number Floating point number

Search for: ELFParser-NG

Options

Text encoding: (Editor encoding)

Search direction: All Forward Backward

Case sensitive

OK Search all Cancel

Special editors

Data inspector

Binary (8 bit) 01001101

Int8 go to: 77

UInt8 go to: 77

Int16 go to: 23117

UInt16 go to: 23117

Int24 go to: -7316915

UInt24 go to: 9460301

Int32 go to: 9460301

UInt32 go to: 9460301

Int64 go to: 12894362189

UInt64 go to: 12894362189

LEB128 go to: -51

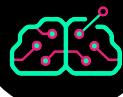
ULEB128 go to: 77

AnsiChar / char8_ M

Byte order: Little endian Big endian

Hexadecimal basis (for integral numbers)

Overwrite



Lab 05 - Manipulando strings - Respostas

1.

HxD - [C:\Users\admin\Desktop\binarios\strings1.exe]

File Edit Search View Analysis Tools Window Help

strings1.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
000012B0	06 28 00 00 5E 29 00 00 CC 28 00 00 00 00 00 00	(..^)....í(.....
000012C0	FE 16 40 00 00 00 00 00 00 00 00 18 11 40 00	p.@.....@.
000012D0	00 00 00 00 00 00 00 00 65 10 40 00 10 11 40 00e.@...@.
000012E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000012F0	00 00 00 00 00 00 00 00 18 30 40 00 68 30 40 000@.h0@.
00001300	4D 65 6E 74 65 20 42 69 6E 61 72 69 61 00 00 00	Mente Binaria...
00001310	50 61 70 6F 20 42 69 6E 61 72 69 6F 00 00 00 00	Papo Binario....
00001320	44 6F 20 5A 65 72 6F 20 41 6F 20 55 6D 00 00 00	Do Zero Ao Um...
00001330	45 6E 67 65 6E 68 61 72 69 61 20 52 65 76 65 72	Engenharia Rever
00001340	73 61 20 2D 20 46 75 6E 64 61 6D 65 6E 74 6F 73	sa - Fundamentos
00001350	20 65 20 50 72 61 74 69 63 61 00 00 45 4C 46 50	e Pratica...ELFF
00001360	61 72 73 65 72 2D 4E 47 00 00 00 00 4A 41 43 48	arser-NG.....JACK
00001370	00 00 00 00 00 00 00 00 C0 00 00 00 00 00 00 00À.....
00001380	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001390	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013B0	00 00 00 00 04 30 40 00 C0 22 40 00 01 00 00 000@.À"@"....
000013C0	C0 20 40 00 00 00 00 00 00 00 00 00 00 00 00 00	À @.....
000013D0	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001410	00 00 00 00 00 00 00 00 C4 22 40 00 00 00 00 00À"@"....
00001420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Offset(h): 135C
Block(h): 135C-1367
Length(h): C
Overwrite

Special editors

Data inspector

Binary (8 bit) 01000101

Int8 go to: 69

UInt8 go to: 69

Int16 go to: 19525

UInt16 go to: 19525

Int24 go to: 4607045

UInt24 go to: 4607045

Int32 go to: 1346784325

UInt32 go to: 1346784325

Int64 go to: 731031238243662:

UInt64 go to: 731031238243662:

LEB128 go to: -59

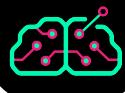
ULEB128 go to: 69

AnsiChar / char8_E

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Lab 05 - Manipulando strings - Respostas

1.

HxD - [C:\Users\admin\Desktop\binarios\strings1.exe]

File Edit Search View Analysis Tools Window Help

strings1.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
000012B0	06 28 00 00 5E 29 00 00 CC 28 00 00 00 00 00 00	(..^)....í(.....
000012C0	FE 16 40 00 00 00 00 00 00 00 00 18 11 40 00	p.@.....@.
000012D0	00 00 00 00 00 00 00 00 65 10 40 00 10 11 40 00e.@...@.
000012E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000012F0	00 00 00 00 00 00 00 00 18 30 40 00 68 30 40 000@.h0@.
00001300	4D 65 6E 74 65 20 42 69 6E 61 72 69 61 00 00 00	Mente Binaria...
00001310	50 61 70 6F 20 42 69 6E 61 72 69 6F 00 00 00 00	Papo Binario....
00001320	44 6F 20 5A 65 72 6F 20 41 6F 20 55 6D 00 00 00	Do Zero Ao Um...
00001330	45 6E 67 65 6E 68 61 72 69 61 20 52 65 76 65 72	Engenharia Rever
00001340	73 61 20 2D 20 46 75 6E 64 61 6D 65 6E 74 6F 73	sa - Fundamentos
00001350	20 65 20 50 72 61 74 69 63 61 00 00 45 4C 46 50	e Pratica...ELFF
00001360	61 72 73 65 72 2D 4E 47 00 00 00 00 4A 41 43 48	arser-NG.....JACK
00001370	00 00 00 00 00 00 00 00 C0 00 00 00 00 00 00 00À.....
00001380	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001390	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013B0	00 00 00 00 04 30 40 00 C0 22 40 00 01 00 00 000@.À"@"....
000013C0	C0 20 40 00 00 00 00 00 00 00 00 00 00 00 00 00	À @.....
000013D0	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001410	00 00 00 00 00 00 00 00 C4 22 40 00 00 00 00 00À"@"....
00001420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Offset(h): 135C Block(h): 135C-1367 Length(h): C Overwrite

Special editors

Data inspector

Binary (8 bit) 01000101

Int8 go to: 69

UInt8 go to: 69

Int16 go to: 19525

UInt16 go to: 19525

Int24 go to: 4607045

UInt24 go to: 4607045

Int32 go to: 1346784325

UInt32 go to: 1346784325

Int64 go to: 731031238243662:

UInt64 go to: 731031238243662:

LEB128 go to: -59

ULEB128 go to: 69

AnsiChar / char8_E

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Lab 05 - Manipulando strings - Respostas

1.

HxD - [C:\Users\admin\Desktop\binarios\strings1.exe]

File Edit Search View Analysis Tools Window Help

strings1.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
000012B0	06 28 00 00 5E 29 00 00 CC 28 00 00 00 00 00 00	(..^)....í(.....
000012C0	FE 16 40 00 00 00 00 00 00 00 00 00 18 11 40 00	p.º.....@.
000012D0	00 00 00 00 00 00 00 00 65 10 40 00 10 11 40 00e.º@.º@.
000012E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000012F0	00 00 00 00 00 00 00 00 18 30 40 00 68 30 40 00@.h0@.
00001300	4D 65 6E 74 65 20 42 69 6E 61 72 69 61 00 00 00	Mente Binaria...
00001310	50 61 70 6F 20 42 69 6E 61 72 69 6F 00 00 00 00	Papo Binario....
00001320	44 6F 20 5A 65 72 6F 20 41 6F 20 55 6D 00 00 00	Do Zero Ao Um...
00001330	45 6E 67 65 6E 68 61 72 69 61 20 52 65 76 65 72	Engenharia Rever
00001340	73 61 20 2D 20 46 75 6E 64 61 6D 65 6E 74 6F 73	sa - Fundamentos
00001350	20 65 20 50 72 61 74 69 63 61 00 00 45 4C 46 50	e Pratica...ELFF
00001360	61 72 73 65 72 2D 4E 47 00 00 00 00 4A 41 43 48	arser-NG.....JACK
00001370	00 00 00 00 00 00 00 00 C0 00 00 00 00 00 00 00À.....
00001380	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001390	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013B0	00 00 00 00 04 30 40 00 C0 22 40 00 01 00 00 00@.À"º@....
000013C0	C0 20 40 00 00 00 00 00 00 00 00 00 00 00 00 00	À @.....
000013D0	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001410	00 00 00 00 00 00 00 00 C4 22 40 00 00 00 00 00À"º@....
00001420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Offset(h): 135C Block(h): 135C-1367 Length(h): C Overwrite

Special editors

Data inspector

Binary (8 bit) 01000101

Int8 go to: 69

UInt8 go to: 69

Int16 go to: 19525

UInt16 go to: 19525

Int24 go to: 4607045

UInt24 go to: 4607045

Int32 go to: 1346784325

UInt32 go to: 1346784325

Int64 go to: 731031238243662:

UInt64 go to: 731031238243662:

LEB128 go to: -59

ULEB128 go to: 69

AnsiChar / char8_E

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Lab 05 - Manipulando strings - Respostas

1.

HxD - [C:\Users\admin\Desktop\binarios\strings1.exe]

File Edit Search View Analysis Tools Window Help

strings1.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
000012B0	06 28 00 00 5E 29 00 00 CC 28 00 00 00 00 00 00 00	(..^)....í(.....
000012C0	FE 16 40 00 00 00 00 00 00 00 00 00 18 11 40 00	p.º.....@.
000012D0	00 00 00 00 00 00 00 00 65 10 40 00 10 11 40 00e.º@.º@.
000012E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000012F0	00 00 00 00 00 00 00 00 18 30 40 00 68 30 40 00@.h0@.
00001300	4D 65 6E 74 65 20 42 69 6E 61 72 69 61 00 00 00	Mente Binaria...
00001310	50 61 70 6F 20 42 69 6E 61 72 69 6F 00 00 00 00	Papo Binario....
00001320	44 6F 20 5A 65 72 6F 20 41 6F 20 55 6D 00 00 00	Do Zero Ao Um...
00001330	45 6E 67 65 6E 68 61 72 69 61 20 52 65 76 65 72	Engenharia Rever
00001340	73 61 20 2D 20 46 75 6E 64 61 6D 65 6E 74 6F 73	sa - Fundamentos
00001350	20 65 20 50 72 61 74 69 63 61 00 00 45 4C 46 50	e Pratica...ELFF
00001360	61 72 73 65 72 2D 4E 47 00 00 00 00 4A 41 43 48	arser-NG.....JACK
00001370	00 00 00 00 00 00 00 00 C0 00 00 00 00 00 00 00À.....
00001380	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001390	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013B0	00 00 00 00 04 30 40 00 C0 22 40 00 01 00 00 00@.À"º@....
000013C0	C0 20 40 00 00 00 00 00 00 00 00 00 00 00 00 00	À @.....
000013D0	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001410	00 00 00 00 00 00 00 00 C4 22 40 00 00 00 00 00À"º@....
00001420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Offset(h): 135C Block(h): 135C-1367 Length(h): C Overwrite

Special editors

Data inspector

Binary (8 bit) 01000101

Int8 go to: 69

UInt8 go to: 69

Int16 go to: 19525

UInt16 go to: 19525

Int24 go to: 4607045

UInt24 go to: 4607045

Int32 go to: 1346784325

UInt32 go to: 1346784325

Int64 go to: 731031238243662:

UInt64 go to: 731031238243662:

LEB128 go to: -59

ULEB128 go to: 69

AnsiChar / char8_E

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Lab 05 - Manipulando strings - Extra

HxD - [C:\Users\admin\Desktop\binarios\strings1.exe]

File Edit Search View Analysis Tools Window Help

strings1.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
000012B0	06 28 00 00 5E 29 00 00 CC 28 00 00 00 00 00 00	.(..^)...í(.....
000012C0	FE 16 40 00 00 00 00 00 00 00 00 00 18 11 40 00	p.º.....º.
000012D0	00 00 00 00 00 00 00 00 65 10 40 00 10 11 40 00e.º.º.
000012E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000012F0	00 00 00 00 00 00 00 00 18 30 40 00 68 30 40 00º.º.º.º.
00001300	4D 65 6E 74 65 20 42 69 6E 61 72 69 61 00 00 00	Mente Binaria...
00001310	50 61 70 6F 20 42 69 6E 61 72 69 6F 00 00 00 00	Papo Binario....
00001320	44 6F 20 5A 65 72 6F 20 41 6F 20 55 6D 00 00 00	Do Zero Ao Um...
00001330	45 6E 67 65 6E 68 61 72 69 61 20 52 65 76 65 72	Engenharia Reversa - Fundamentos
00001340	73 61 20 2D 20 46 75 6E 64 61 6D 65 6E 74 6F 73	e Pratica..ELFP
00001350	20 65 20 50 72 61 74 69 63 61 00 00 45 4C 46 50	arser-NG....JACK
00001360	61 72 73 65 72 2D 4E 47 00 00 00 00 4A 41 43 4BÁ.....
00001370	00 00 00 00 00 00 00 00 C0 00 00 00 00 00 00 00 00Á.....
00001380	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001390	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013B0	00 00 00 00 04 30 40 00 C0 22 40 00 01 00 00 00º.º@.....
000013C0	C0 20 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00	À º.....
000013D0	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000013F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001410	00 00 00 00 00 00 00 00 C4 22 40 00 00 00 00 00 00º@.....
00001420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Offset(h): 1330 Block(h): 1330-1359 Length(h): 2A Overwrite

Special editors

Data inspector

Binary (8 bit) 01000101

Int8 go to: 69

UInt8 go to: 69

Int16 go to: 28229

UInt16 go to: 28229

Int24 go to: 6778437

UInt24 go to: 6778437

Int32 go to: 1701277253

UInt32 go to: 1701277253

Int64 go to: 8241983616421680

UInt64 go to: 8241983616421680

LEB128 go to: -59

ULEB128 go to: 69

AnsiChar / char8_E

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Lab 05 - Manipulando strings - Extra

HxD - [C:\Users\admin\AppData\Local\Programs\retoolkit\hexeditors\hxd\HxD.exe]

File Edit Search View Analysis Tools Window Help

Offset(d) 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 Decoded text

00114816	C7 C0 C5 00 00 00 E8 A5 1E FF FF E8 10 D7 FF FF	ÇÀ...è¥.ÿè..×ÿ
00114832	48 89 C1 48 33 D2 4D 33 C0 E8 32 D6 FF FF 48 8D	HtÁH3ÒM3Àè2ÖÿÿH.
00114848	65 60 5D C3 48 8D 40 00 48 8D 04 05 00 00 00 00	e`]ÃH..@.H.....
00114864	55 48 83 EC 20 48 8B EC 48 8D 44 2A 58 48 8B CD	UHfi H<íH.D*XH<í
00114880	48 F7 D9 48 8B OC 08 48 8D 44 2A 54 4C 8B C5 49	H-ÜH<..H.D*TL<ÅI
00114896	F7 D8 4A 63 04 00 4C 8D 44 2A 50 4C 8B CD 49 F7	÷ØJc..L.D*PL<ÍI+Ü
00114912	D9 47 8B 04 08 48 8D 54 2A 4C 4C 8B CD 49 F7 D9	ÜG<..H.T*LL<ÍI=Ü
00114928	4E 8D OC 0A 48 89 C2 E8 E4 DD FF FF 85 C0 75 1A	N...HtÁeaÿÿ.Au.
00114944	48 8D OD 25 00 00 00 48 8D 15 56 00 00 00 41 C7	H..%....H.V...AÇ
00114960	C0 C5 00 00 00 E8 16 1E FF FF 48 8D 65 20 5D C3	À...è...ÿÿH.e]À
00114976	B0 04 02 00 FF FF FF FF 15 00 00 00 56 00 69 00	°...ÿÿÿ...V.i.
00114992	72 00 74 00 75 00 61 00 6C 00 50 00 72 00 6F 00	r.t.u.a.l.P.r.o.
00115008	74 00 65 00 63 00 74 00 20 00 66 00 61 00 69 00	t.e.c.t. .f.a.i.
00115024	6C 00 65 00 64 00 00 00 B0 04 02 00 FF FF FF FF	l.e.d...°...ÿÿÿ
00115040	31 00 00 00 44 00 3A 00 5C 00 51 00 75 00 65 00	1...D...\\Q.u.e.
00115056	6C 00 6C 00 65 00 6E 00 5C 00 4B 00 6F 00 6D 00	1.l.e.n\\.K.o.m.
00115072	70 00 6F 00 6E 00 65 00 6E 00 74 00 65 00 6E 00	p.o.n.e.n.t.e.n.
00115088	5C 00 58 00 6D 00 4D 00 69 00 73 00 63 00 5C 00	\.X.m.M.i.s.c\\.X.
00115104	53 00 6F 00 75 00 72 00 63 00 65 00 5C 00 58 00	S.o.u.r.c.e\\.X.
00115120	6D 00 53 00 79 00 73 00 74 00 65 00 6D 00 2E 00	m.S.y.s.t.e.m...
00115136	70 00 61 00 73 00 00 00 CC CC CC CC CC CC CC	p.a.s....iiiiiiii
00115152	55 48 83 EC 30 48 8B EC 48 89 4D 40 48 8B 45 40	UHfiOH<íHtM@H<E@
00115168	48 83 38 00 0F 84 DD 00 00 00 48 8B 45 40 48 8B	Hf8...Í...H<E@H<
00115184	48 08 F8 F0 F2 FF FF 48 8B 4D 40 48 8B 00 48 8B	H AWD@H@H<M@H<H@

Offset(d): 115002 Block(d): 115002-115031 Length(d): 30 Overwrite



IDA - Manipulando strings - Extra

IDA - strings1.exe C:\Users\admin\Desktop\binarios\strings1.exe

File Edit Jump Search View Debugger Options Windows Help

Local Windows debugger

Library function Regular function Instruction Data Unexplored External symbol Lumina function

Functions IDA View-A Strings Hex View-1 Local Types Imports Exports

Address	Length	Type	String
0x401040...	0000000E	C	Mente Binaria
0x401040...	0000000D	C	Papo Binario
0x401040...	0000000C	C	Do Zero Ao Um
0x401040...	0000002B	C	Engenharia Reversa - Fundamentos e Pratica
0x401040...	0000000D	C	ELFParser-NG
0x401040...	00000034	C - U...	C:\Users\admin\source\repos\teste\Release\teste.pdb
0x401040...	00000009	C	.text\$mn
0x401040...	00000009	C	.idata\$5
0x401040...	00000007	C	.0cfg
0x401040...	00000009	C	.CRT\$KCA
0x401040...	0000000A	C	.CRT\$KCAA
0x401040...	00000009	C	.CRT\$KZ
0x401040...	00000009	C	.CRT\$KIA
0x401040...	0000000A	C	.CRT\$KIAA
0x401040...	0000000A	C	.CRT\$KAC
0x401040...	00000009	C	.CRT\$KIZ
0x401040...	00000009	C	.CRT\$KPA
0x401040...	00000009	C	.CRT\$KPZ
0x401040...	00000009	C	.CRT\$KT
0x401040...	00000007	C	.rdata
0x401040...	0000000E	C	.rdata\$sxdata
0x401040...	0000000E	C	.rdata\$volmd
0x401040...	0000000E	C	.rdata\$zzzdb
0x401040...	00000009	C	.rtc\$IAA
0x401040...	00000009	C	.rtc\$IZZ
0x401040...	00000009	C	.rtc\$TAA
0x401040...	00000009	C	.rtc\$TZZ
0x401040...	00000009	C	.xdata\$x
0x401040...	00000009	C	.idata\$2
0x401040...	00000009	C	.idata\$3
0x401040...	00000009	C	.idata\$4
0x401040...	00000009	C	.idata\$6
0x401040...	00000006	C	.data
0x401040...	00000009	C	.rcrc\$01

Line 1 of 82

Graph overview

Please check the Edit/Plugins menu for more information.
Using FLIRT signature: SEH for vc7-14
Propagating type information...
40193A: propagate_starargs: function is already typed
Function argument information has been propagated
The initial autoanalysis has been finished.

IDC

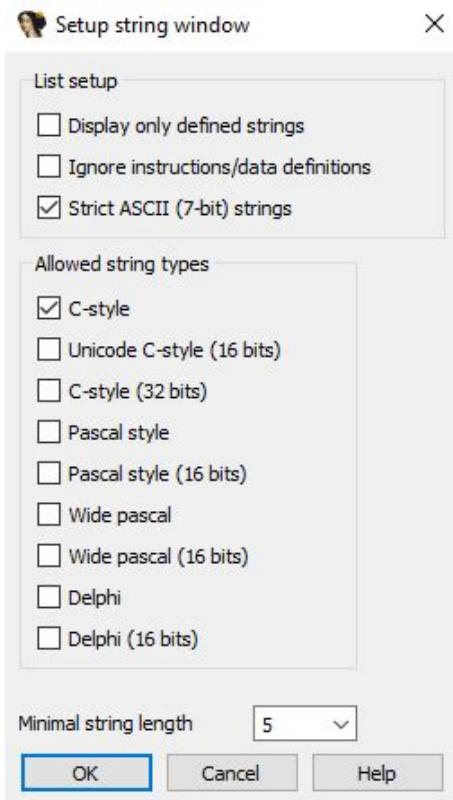
AU: idle Down Disk: 26GB

Activate Windows
Go to Settings to activate Windows.



IDA - Manipulando strings - Extra

Address	Length	Type	String
.rdata:0040...	0000000E	C	Mente Binaria
.rdata:0040...	0000000D	C	Papo Binario
.rdata:0040...	0000000E	C	Do Zero Ao Um
.rdata:0040...	0000002B	C	Engenharia Reversa - Fundamentos e Pratica
.rdata:0040...	0000000D	C	ELFParse-NG
.rdata:0040...	00000034	C - U...	C:\Users\admin\source\repos\teste\Release\teste.pdb
.rdata:0040...	00000009	C	.text\$mn
.rdata:0040...	00000009	C	.idata\$5
.rdata:0040...	00000007	C	.00cfg
.rdata:0040...	00000009	C	.CRT\$XCAA
.rdata:0040...	0000000A	C	.CRT\$XCAA
.rdata:0040...	00000009	C	.CRT\$XCZ
.rdata:0040...	00000009	C	.CRT\$XIA
.rdata:0040...	0000000A	C	.CRT\$XIAA
.rdata:0040...	0000000A	C	.CRT\$XIAC
.rdata:0040...	00000009	C	.CRT\$XIZ
.rdata:0040...	00000009	C	.CRT\$XPA
.rdata:0040...	00000009	C	.CRT\$XPZ
.rdata:0040...	00000009	C	.CRT\$XTA
.rdata:0040...	00000009	C	.CRT\$TZ
.rdata:0040...	00000007	C	.rdata
.rdata:0040...	0000000E	C	.rdata\$sxdata
.rdata:0040...	0000000E	C	.rdata\$volmd
.rdata:0040...	0000000E	C	.rdata\$zzzdbg
.rdata:0040...	00000009	C	.rtc\$IAA
.rdata:0040...	00000009	C	.rtc\$IZZ
.rdata:0040...	00000009	C	.rtc\$TAA
.rdata:0040...	00000009	C	.rtc\$TZZ
.rdata:0040...	00000009	C	.xdata\$x
.rdata:0040...	00000009	C	.idata\$2
.rdata:0040...	00000009	C	.idata\$3
.rdata:0040...	00000009	C	.idata\$4
.rdata:0040...	00000009	C	.idata\$6
.rdata:0040...	00000006	C	.data
.rdata:0040...	00000000	C

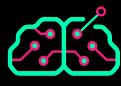




Lab 05 - Manipulando strings - Respostas

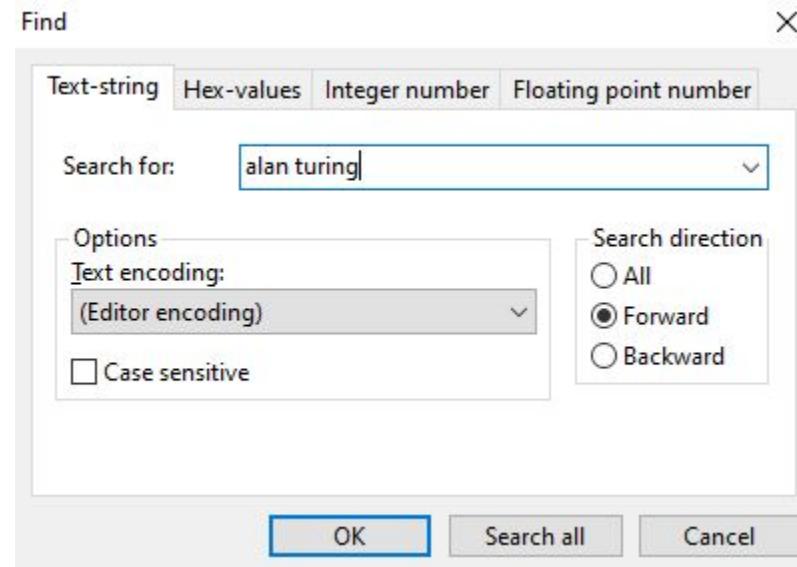
2.





Lab 05 - Manipulando strings - Respostas

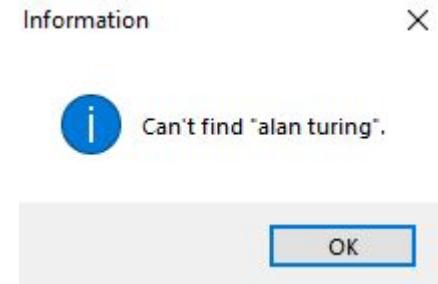
2.





Lab 05 - Manipulando strings - Respostas

2.





Lab 05 - Manipulando strings - Respostas

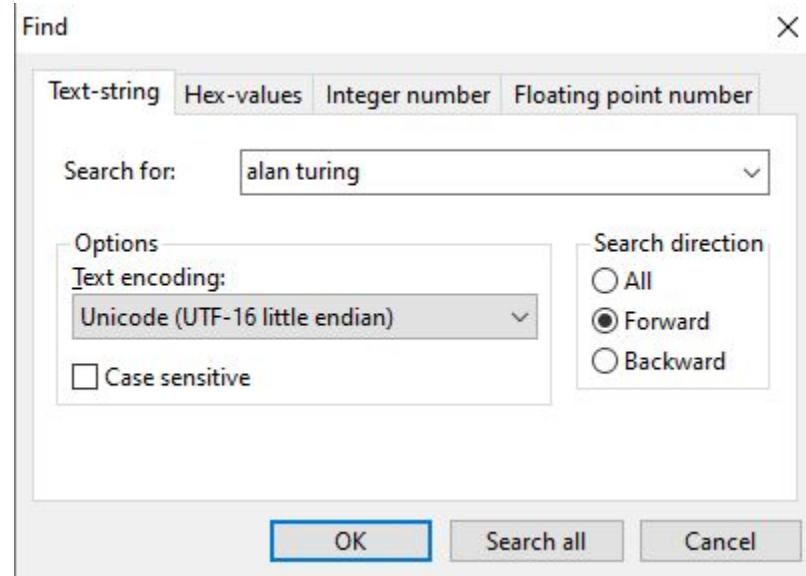
2.

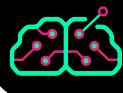




Lab 05 - Manipulando strings - Respostas

2.





Lab 05 - Manipulando strings - Respostas

2.

HxD - [C:\Users\admin\Desktop\binarios\strings2.exe]

File Edit Search View Analysis Tools Window Help

strings1.exe strings2.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000F30	54 65 78 74 52 65 6E 64 65 72 69 6E 67 44 65 66	TextRenderingDef
00000F40	61 75 6C 74 00 49 6E 69 74 69 61 6C 69 7A 65 43	ault.InitializeC
00000F50	6F 6D 70 6F 6E 65 6E 74 00 50 6F 69 6E 74 00 73	omponent.Point.s
00000F60	65 74 5F 46 6F 6E 74 00 53 75 73 70 65 6E 64 4C	et_Font.SuspendL
00000F70	61 79 6F 75 74 00 52 65 73 75 6D 65 4C 61 79 6F	ayout.ResumeLayo
00000F80	75 74 00 50 65 72 66 6F 72 6D 4C 61 79 6F 75 74	ut.PerformLayout
00000F90	00 73 65 74 5F 54 65 78 74 00 73 65 74 5F 54 61	.set_Text.set_Ta
00000FA0	62 49 6E 64 65 78 00 67 65 74 5F 41 73 73 65 6D	bIndex.get_Assem
00000FB0	62 6C 79 00 00 0F 56 00 65 00 72 00 64 00 61 00	bly...V.e.r.d.a.
00000FC0	6E 00 61 00 00 0D 6C 00 61 00 62 00 65 00 6C 00	n.a...1.a.b.e.l.
00000FD0	31 00 00 17 41 00 6C 00 61 00 6E 00 20 00 54 00	1...A.l.a.n. .T.
00000FE0	75 00 72 00 69 00 6E 00 67 00 00 0B 46 00 6F 00	u.r.i.n.g...F.o.
00000FF0	72 00 6D 00 31 00 00 51 73 00 74 00 72 00 69 00	r.m.l.Qs.t.r.i.
00001000	6E 00 67 00 73 00 20 00 32 00 20 00 20 00 20 00	n.g.s. .2. .-..
00001010	41 00 20 00 41 00 72 00 74 00 65 00 20 00 64 00	A. .A.r.t.e. .d.
00001020	61 00 20 00 45 00 6E 00 67 00 65 00 6E 00 68 00	a. .E.n.g.e.n.h.
00001030	61 00 72 00 69 00 61 00 20 00 52 00 65 00 76 00	a.r.i.a. .R.e.v.
00001040	65 00 72 00 73 00 61 00 01 3B 73 00 74 00 72 00	e.r.s.a.;s.str.
00001050	69 00 6E 00 67 00 73 00 32 00 2E 00 50 00 72 00	i.n.g.s.2...P.r.
00001060	6F 00 70 00 65 00 72 00 74 00 69 00 65 00 73 00	o.p.e.r.t.i.e.s.
00001070	2E 00 52 00 65 00 73 00 6F 00 75 00 72 00 63 00	..R.e.s.o.u.r.c.
00001080	65 00 73 00 00 00 00 96 41 5B EA D5 69 2D 46	e.s.....-A[éÖi-F
00001090	91 EE 58 28 DA A1 FF E7 00 04 20 01 01 08 03 20	'iX(Ú;Ýç...
000010A0	00 01 05 20 01 01 11 11 04 20 01 01 0F 04 20 01	

Special editors x

Data inspector

Binary (8 bit) 01000001

Int8 go to: 65

UInt8 go to: 65

Int16 go to: 65

UInt16 go to: 65

Int24 go to: 7077953

UInt24 go to: 7077953

Int32 go to: 7077953

UInt32 go to: 7077953

Int64 go to: 309626640570778;

UInt64 go to: 309626640570778;

LEB128 go to: -63

ULEB128 go to: 65

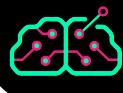
AnsiChar / char8 A

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)

Offset(h): FD4 Block(h): FD4-FE9 Length(h): 16 Overwrite



Lab 05 - Manipulando strings - Respostas

2.

HxD - [C:\Users\admin\Desktop\binarios\strings2.exe]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

Special editors

Data inspector

Offset(h)	Decoded text
00000F30	TextRenderingDef
00000F40	ault.InitializeC
00000F50	omponent.Point.s
00000F60	et_Font.SuspendL
00000F70	ayout.ResumeLayo
00000F80	ut.PerformLayout
00000F90	.set_Text.set_Ta
00000FA0	bIndex.get_Asym
00000FB0	bly...V.e.r.d.a.
00000FC0	n.a...l.a.b.e.l.
00000FD0	l...T.h.a.y.s.e.
00000FF0	u.r.i.n.g...F.o.
00001000	r.m.l..Qs.t.r.i.
00001010	n.g.s...2...-.-
00001020	A...A.r.t.e...d.
00001030	a...E.n.g.e.n.h.
00001040	a.r.i.a...R.e...v.
00001050	e.r.s.a...;s.str.
00001060	i.n.g.s.2...P.r.
00001070	o.p.e.r.t.i.e.s.
00001080	..R.e.s.o.u.r.c.
00001090	e.s....-A[éõi-F
000010A0	'iX(Ú;ÿç...

Binary (8 bit) 00000000

Int8 go to: 0

UInt8 go to: 0

Int16 go to: 30208

UInt16 go to: 30208

Int24 go to: 30208

UInt24 go to: 30208

Int32 go to: 1694529024

UInt32 go to: 1694529024

Int64 go to: 8286748660381808

UInt64 go to: 8286748660381808

LEB128 go to: 0

ULEB128 go to: 0

AnsiChar/char8_

Byte order

Little endian Big endian

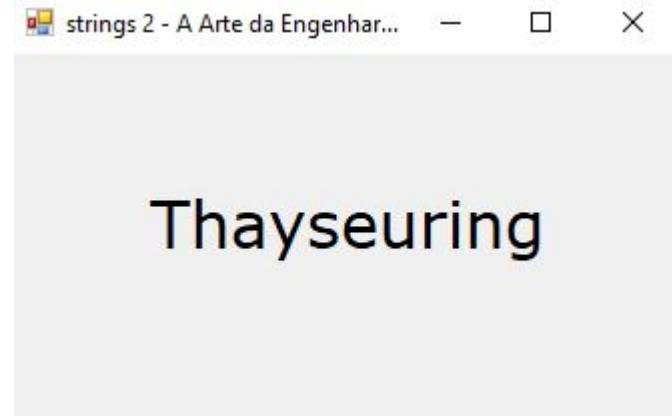
Hexadecimal basis (for integral numbers)

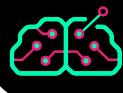
Offset(h): 103D * Modified * Overwrite



Lab 05 - Manipulando strings - Respostas

2.





Lab 05 - Manipulando strings - Respostas

2.

HxD - [C:\Users\admin\Desktop\binarios\strings2.exe]

File Edit Search View Analysis Tools Window Help

strings1.exe strings2.exe

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000F30	54 65 78 74 52 65 6E 64 65 72 69 6E 67 44 65 66	TextRenderingDef
00000F40	61 75 6C 74 00 49 6E 69 74 69 61 6C 69 7A 65 43	ault.InitializeC
00000F50	6F 6D 70 6F 6E 65 6E 74 00 50 6F 69 6E 74 00 73	omponent.Point.s
00000F60	65 74 5F 46 6F 6E 74 00 53 75 73 70 65 6E 64 4C	et_Font.SuspendL
00000F70	61 79 6F 75 74 00 52 65 73 75 6D 65 4C 61 79 6F	ayout.ResumeLayo
00000F80	75 74 00 50 65 72 66 6F 72 6D 4C 61 79 6F 75 74	ut.PerformLayout
00000F90	00 73 65 74 5F 54 65 78 74 00 73 65 74 5F 54 61	.set_Text.set_Ta
00000FA0	62 49 6E 64 65 78 00 67 65 74 5F 41 73 73 65 6D	bIndex.get_Asym
00000FB0	62 6C 79 00 00 0F 56 00 65 00 72 00 64 00 61 00	bly...V.e.r.d.a.
00000FC0	6E 00 61 00 00 0D 6C 00 61 00 62 00 65 00 6C 00	n.a...l.a.b.e.l.
00000FD0	31 00 00 17 54 00 68 00 61 00 79 00 73 00 65 00	l...T.h.a.y.s.e.
00000FE0	00 00 72 00 69 00 6E 00 67 00 00 0B 46 00 6F 00	.g.r.i.n.g...F.o.
00000FF0	72 00 6D 00 31 00 00 51 73 00 74 00 72 00 69 00	r.m.l..Qs.t.r.i.
00001000	6E 00 67 00 73 00 20 00 32 00 20 00 2D 00 20 00	n.g.s...2. - . .
00001010	41 00 20 00 41 00 72 00 74 00 65 00 20 00 64 00	A. .A.r.t.e. .d.
00001020	61 00 20 00 45 00 6E 00 67 00 65 00 6E 00 68 00	a. .E.n.g.e.n.h.
00001030	61 00 72 00 69 00 61 00 20 00 52 00 65 00 76 00	a.r.i.a. .R.e.v.
00001040	65 00 72 00 73 00 61 00 01 3B 73 00 74 00 72 00	e.r.s.a.;s.str.
00001050	69 00 6E 00 67 00 73 00 32 00 2E 00 50 00 72 00	i.n.g.s.2...P.r.
00001060	6F 00 70 00 65 00 72 00 74 00 69 00 65 00 73 00	o.p.e.r.t.i.e.s.
00001070	2E 00 52 00 65 00 73 00 6F 00 75 00 72 00 63 00	..R.e.s.c.u.r.c.
00001080	65 00 73 00 00 00 00 96 41 5B EA D5 69 2D 46	e.s.....-A[êÖi-F
00001090	91 EE 58 28 DA A1 FF E7 00 04 20 01 01 08 03 20	'iX(Ü;ÿç...
000010A0	00 01 05 20 01 01 11 11 04 20 01 01 0F 04 20 01	

Offset(h): FE1

* Modified * Overwrite

Special editors x

Data inspector

Binary (8 bit) 00000000

Int8 go to: 0

UInt8 go to: 0

Int16 go to: 29184

UInt16 go to: 29184

Int24 go to: 29184

UInt24 go to: 29184

Int32 go to: 1761636864

UInt32 go to: 1761636864

Int64 go to: 7422053133947265

UInt64 go to: 7422053133947265

LEB128 go to: 0

ULEB128 go to: 0

AnsiChar / char8:

Byte order

Little endian Big endian

Hexadecimal basis (for integral numbers)



Lab 05 - Manipulando strings - Respostas

2.

