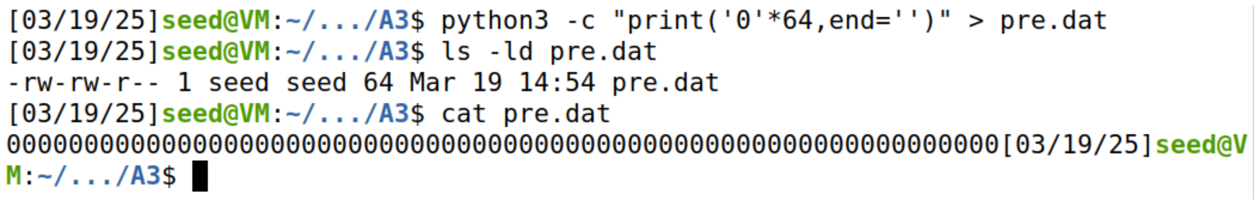
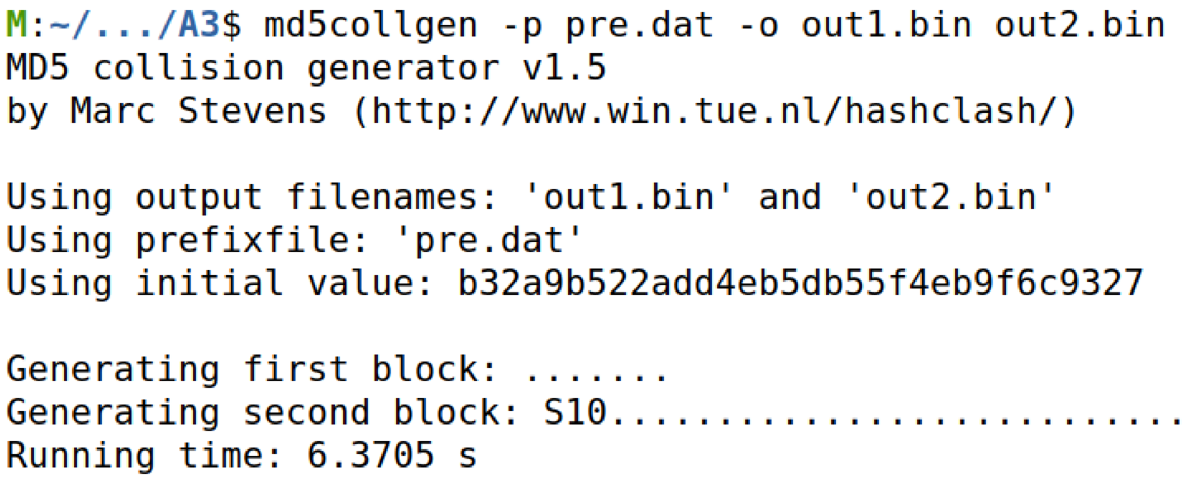
Task 1:

Generate `pre.dat` with `python3 -c "print('9'*64,end='')" > pre.dat`



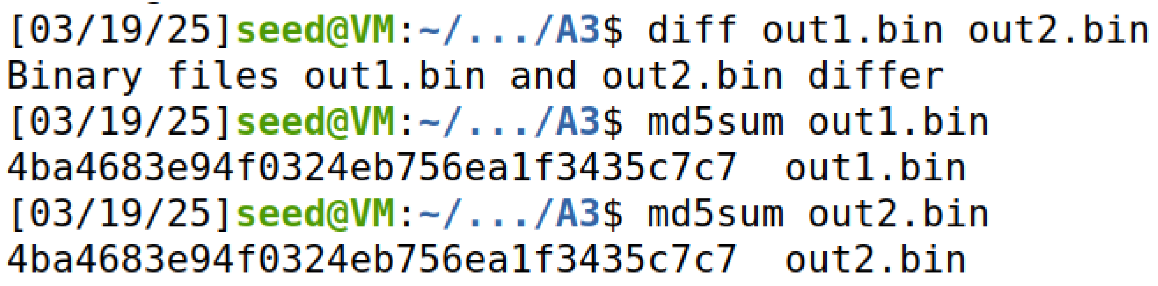
```
[03/19/25]seed@VM:~/.../A3$ python3 -c "print('0'*64,end='')" > pre.dat
[03/19/25]seed@VM:~/.../A3$ ls -ld pre.dat
-rw-rw-r-- 1 seed seed 64 Mar 19 14:54 pre.dat
[03/19/25]seed@VM:~/.../A3$ cat pre.dat
0000000000000000000000000000000000000000000000000000000000000000[03/19/25]seed@V
M:~/.../A3$ 
```

Generate `out1.bin & out2.bin`



```
M:~/.../A3$ md5collgen -p pre.dat -o out1.bin out2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1.bin' and 'out2.bin'
Using prefixfile: 'pre.dat'
Using initial value: b32a9b522add4eb5db55f4eb9f6c9327

Generating first block: .......
Generating second block: S10.............................
Running time: 6.3705 s
```

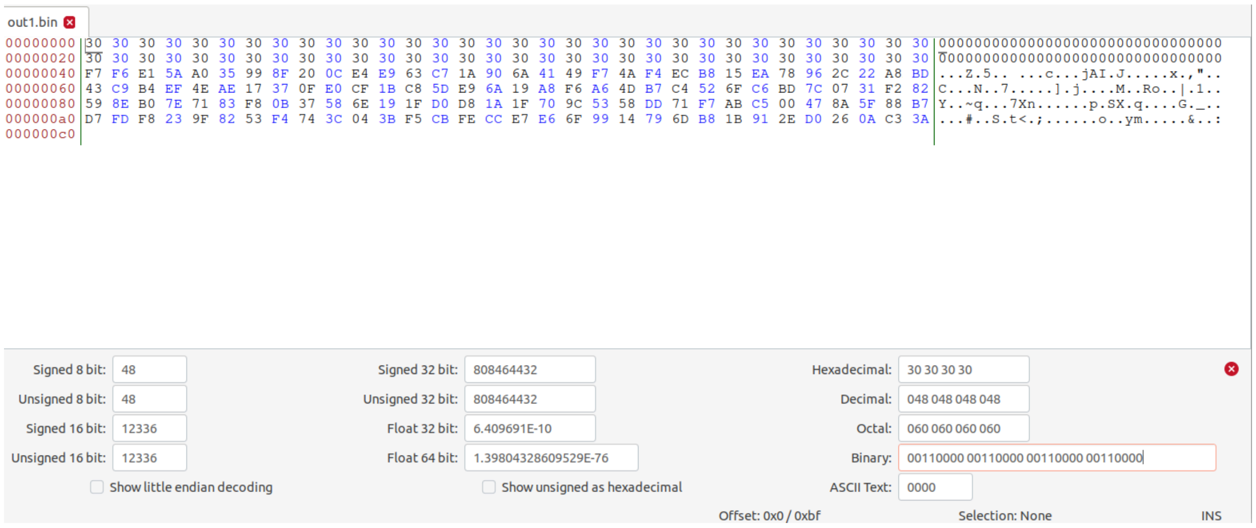
Run given commands



```
[03/19/25]seed@VM:~/.../A3$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[03/19/25]seed@VM:~/.../A3$ md5sum out1.bin
4ba4683e94f0324eb756ea1f3435c7c7  out1.bin
[03/19/25]seed@VM:~/.../A3$ md5sum out2.bin
4ba4683e94f0324eb756ea1f3435c7c7  out2.bin
```

Run Bless

out1.bin



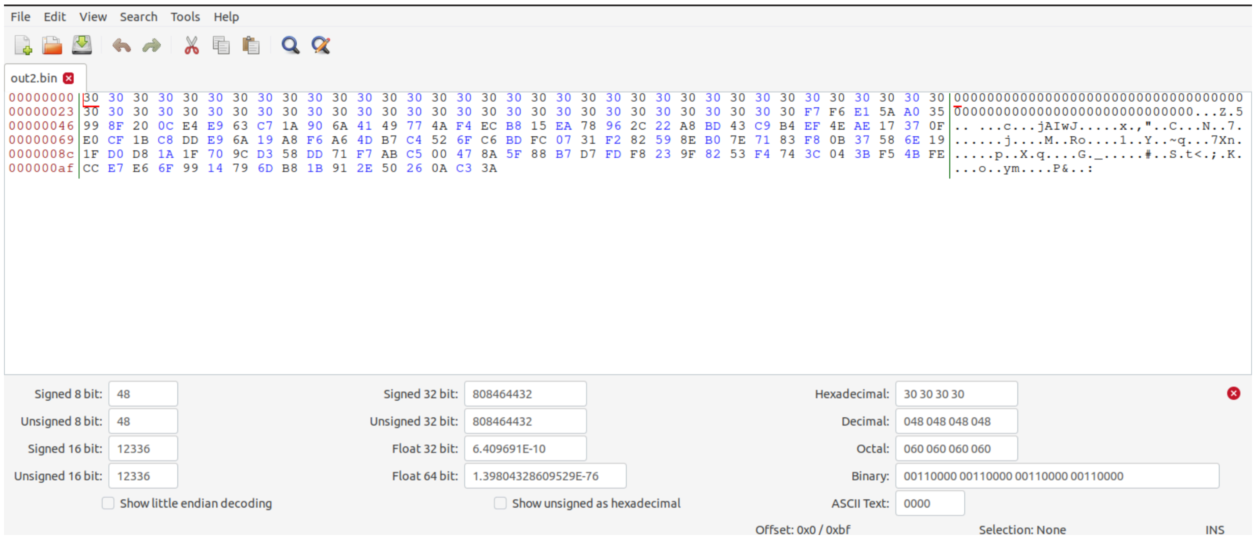
out1.bin

```
00000000 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000000000000000000000000000
00000020 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000000000000000000000000000
00000040 F7 F6 E1 5A A0 35 99 8F 20 0C E4 E9 63 C7 1A 90 6A 41 49 F7 4A F4 EC B8 15 EA 78 96 2C 22 A8 BD ...Z.5.. ...c...jAI.J.....x.,"..
00000060 43 C9 B4 EF 4E AE 17 37 0F E0 CF 1B C8 5D E9 6A 19 A8 F6 A6 4D B7 C4 52 6F C6 BD 7C 07 31 F2 82 C...N..7.....].j....M..Ro..|.1..
00000080 59 8E B0 7E 71 83 F8 0B 37 58 6E 19 1F D0 D8 1A 1F 70 9C 53 58 DD 71 F7 AB C5 00 47 8A 5F 88 B7 Y..~q...7Xn......p.SX.q....G._..
000000a0 D7 FD F8 23 9F 82 53 F4 74 3C 04 3B F5 CB FE CC E7 E6 6F 99 14 79 6D B8 1B 91 2E D0 26 0A C3 3A ...#..S.t<.;......o..ym.....&..:
000000c0
```

Signed 8 bit:	48	Signed 32 bit:	808464432	Hexadecimal:	30 30 30 30
Unsigned 8 bit:	48	Unsigned 32 bit:	808464432	Decimal:	048 048 048 048
Signed 16 bit:	12336	Float 32 bit:	6.409691E-10	Octal:	060 060 060 060
Unsigned 16 bit:	12336	Float 64 bit:	1.39804328609529E-76	Binary:	00110000 00110000 00110000 00110000
Show little endian decoding		Show unsigned as hexadecimal		ASCII Text:	0000

Offset: 0x0 / 0xbf Selection: None INS

Out2.bin



out2.bin

```
00000000 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000000000000000000000000000000
00000023 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 F7 F6 E1 5A A0 35 00000000000000000000000000000...Z.5
00000046 99 8F 20 0C E4 E9 63 C7 1A 90 6A 41 49 77 4A F4 EC B8 15 EA 78 96 2C 22 A8 BD 43 C9 B4 EF 4E AE 17 37 0F .. ...c...jAIwJ.....x.,"..C...N..7.
00000069 E0 CF 1B C8 DD E9 6A 19 A8 F6 A6 4D B7 C4 52 6F C6 BD FC 07 31 F2 82 59 8E B0 7E 71 83 F8 0B 37 58 6E 19 ......j....M..Ro....1..Y..~q...7Xn.
0000008c 1F D0 D8 1A 1F 70 9C D3 58 DD 71 F7 AB C5 00 47 8A 5F 88 B7 D7 FD F8 23 9F 82 53 F4 74 3C 04 3B F5 4B FE .....p..X.q....G._.....#..S.t<.;.K.
000000af CC E7 E6 6F 99 14 79 6D B8 1B 91 2E 50 26 0A C3 3A                                                       ...o..ym....P&..:
```

Signed 8 bit:	48	Signed 32 bit:	808464432	Hexadecimal:	30 30 30 30
Unsigned 8 bit:	48	Unsigned 32 bit:	808464432	Decimal:	048 048 048 048
Signed 16 bit:	12336	Float 32 bit:	6.409691E-10	Octal:	060 060 060 060
Unsigned 16 bit:	12336	Float 64 bit:	1.39804328609529E-76	Binary:	00110000 00110000 00110000 00110000
Show little endian decoding		Show unsigned as hexadecimal		ASCII Text:	0000

Offset: 0x0 / 0xbf Selection: None INS

1. If the length of the prefix file is not a multiple of 64, MD5's padding ensures the total length (prefix + padding) aligns to a 64-byte boundary before the collision blocks are appended
2. If it is exactly 64 bytes, it fills exactly 1 MD5 block, and the collision blocks start immediately after
3. No they are not completely different.

First Block (0-63):

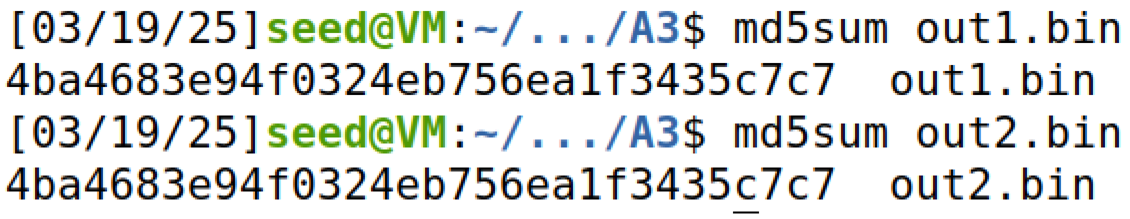
- Byte 2: f7 vs. 77.
- Byte 14: 5d vs. dd.
- Byte 21: 7c vs. fc.

Second Block (64-127):

- Byte 66: 53 vs. d3.
- Byte 78: cb vs. 4b.
- Byte 85: d0 vs. 50.

Task 2:

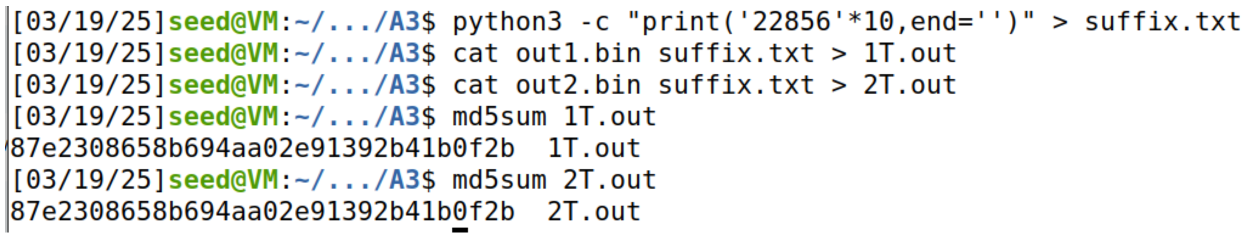
We know the files from the previous step have the same md5 value



```
[03/19/25]seed@VM:~/.../A3$ md5sum out1.bin
4ba4683e94f0324eb756ea1f3435c7c7  out1.bin
[03/19/25]seed@VM:~/.../A3$ md5sum out2.bin
4ba4683e94f0324eb756ea1f3435c7c7  out2.bin
```

So we create a new suffix with `python3 -c "print('114514'*10,end='')" > suffix.txt`

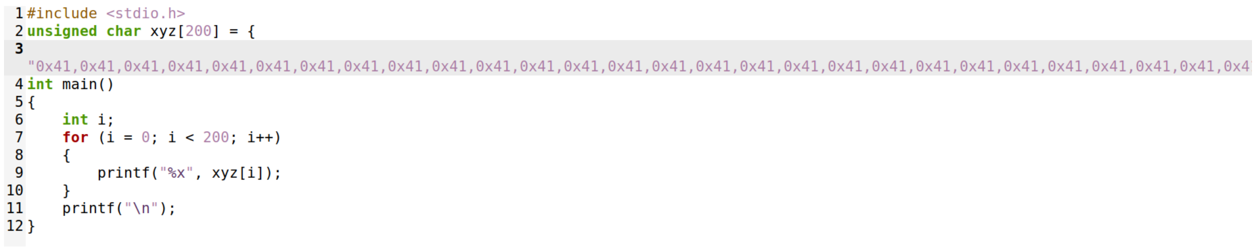
Then concatenate with out1.bin and out2.bin using `cat out1.bin suffix.txt > 1T.out` and `cat out2.bin suffix.txt > 2T.out`



```
[03/19/25]seed@VM:~/.../A3$ python3 -c "print('22856'*10,end='')" > suffix.txt
[03/19/25]seed@VM:~/.../A3$ cat out1.bin suffix.txt > 1T.out
[03/19/25]seed@VM:~/.../A3$ cat out2.bin suffix.txt > 2T.out
[03/19/25]seed@VM:~/.../A3$ md5sum 1T.out
87e2308658b694aa02e91392b41b0f2b  1T.out
[03/19/25]seed@VM:~/.../A3$ md5sum 2T.out
87e2308658b694aa02e91392b41b0f2b  2T.out
```

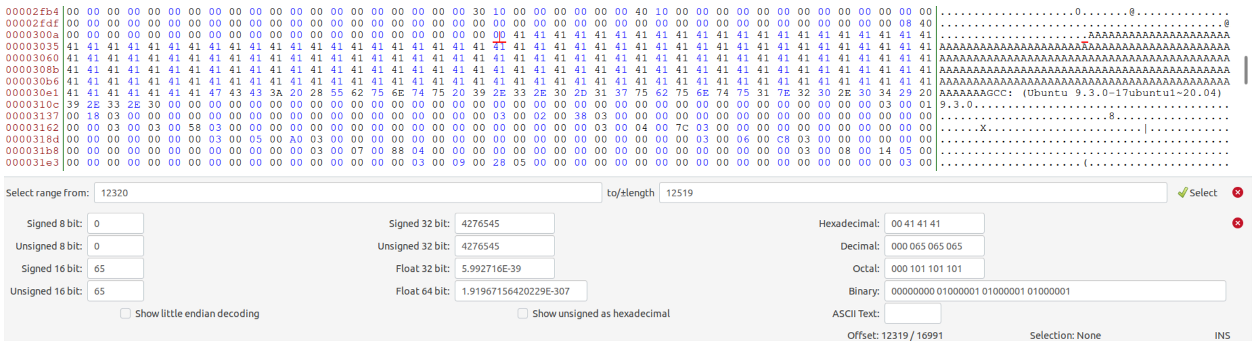
Task 3:

Create and compile `print_array.c`
`*the contents of the array are the hex coding for ‘A’ 200x`



```c
#include <stdio.h>
unsigned char xyz[200] = {
 "0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x4
int main()
{
    int i;
    for (i = 0; i < 200; i++)
    {
        printf("%x", xyz[i]);
    }
    printf("\n");
}
```

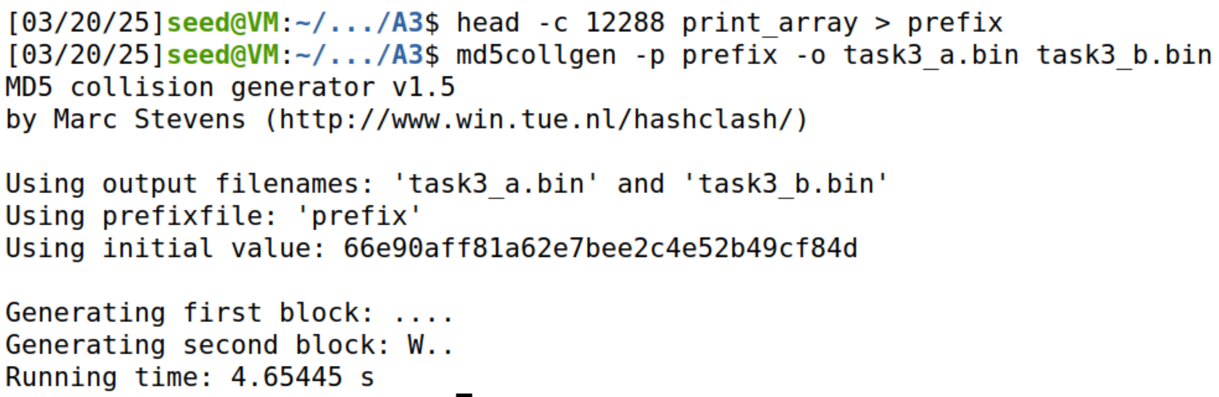
Run `bless print_array`



```
00002fb4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 30 10 00 00 00 00 00 00 40 10 00 00 00 00 00 00 00 00 00 00 00 00 00 ....................0.......@..............
00002fdf 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 40 ..........................................@
0000300a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ......................AAAAAAAAAAAAAAAAAAAAA
00003035 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00003060 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000308b 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000030b6 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000030e1 41 41 41 41 41 41 41 47 43 43 3A 20 28 55 62 75 6E 74 75 20 39 2E 33 2E 30 2D 31 37 75 62 75 6E 74 75 31 7E 32 30 2E 30 34 29 20 AAAAAAAGCC: (Ubuntu 9.3.0-17ubuntu1~20.04)
0000310c 39 2E 33 2E 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 01 9.3.0......................................
00003137 00 18 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 02 00 38 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .........................8.................
00003162 00 00 03 00 03 00 58 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04 00 7C 03 00 00 00 00 00 00 00 00 00 00 00 ......X.......................|............
0000318d 00 00 00 00 00 00 00 03 00 05 00 A0 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 06 00 C8 03 00 00 00 00 00 00 ...........................................
000031b8 00 00 00 00 00 00 00 00 00 00 00 00 03 00 07 00 88 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 08 00 14 05 00 ...........................................
000031e3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 09 00 28 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 .....................(.....................

Select range from: 12320                               to/±length 12519                      Select

    Signed 8 bit: 0                 Signed 32 bit: 4276545                 Hexadecimal: 00 41 41 41
  Unsigned 8 bit: 0               Unsigned 32 bit: 4276545                     Decimal: 000 065 065 065
   Signed 16 bit: 65                 Float 32 bit: 5.992716E-39                  Octal: 000 101 101 101
 Unsigned 16 bit: 65                 Float 64 bit: 1.91967156420229E-307        Binary: 00000000 01000001 01000001 01000001
        [ ] Show little endian decoding     [ ] Show unsigned as hexadecimal    ASCII Text:
                                                            Offset: 12319 / 16991     Selection: None     INS
```

Offset in bottom right states 12320/16991 so array starts at position 12320. Seeing as this is not divisible by 64, we use the closest starting position to the array that is divisible by 64. - 12288

`head -c 12288 print_array > prefix` and `md5collgen -p prefix -o task3_a.bin task3_b.bin`



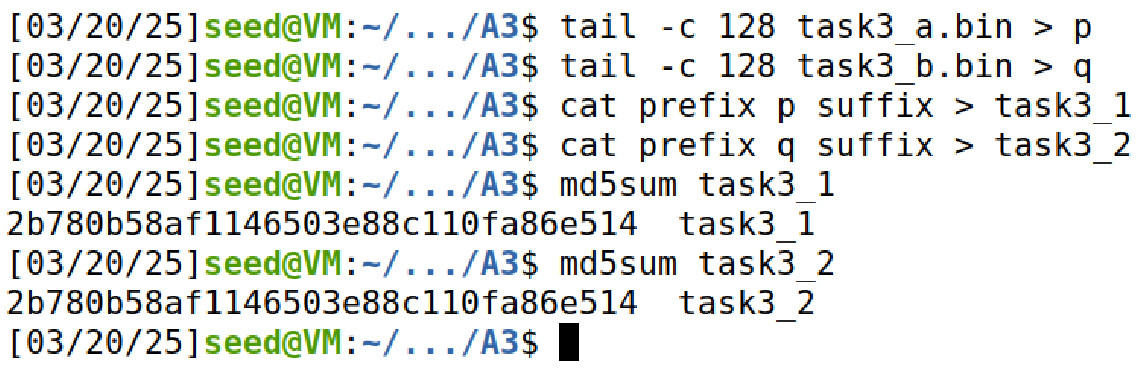
```
[03/20/25]seed@VM:~/.../A3$ head -c 12288 print_array > prefix
[03/20/25]seed@VM:~/.../A3$ md5collgen -p prefix -o task3_a.bin task3_b.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'task3_a.bin' and 'task3_b.bin'
Using prefixfile: 'prefix'
Using initial value: 66e90aff81a62e7bee2c4e52b49cf84d

Generating first block: ....
Generating second block: W..
Running time: 4.65445 s
```

Generate Suffix which should be prefix + 128 so we get 12416
`tail -c +12416 print_array > suffix`

Run

```
Tail -c 128 task3_a.bin > p
Tail -c 128 task3_b.bin > q
Cat prefix p suffix > task3_1
Cat prefix q suffix > task3_2
```

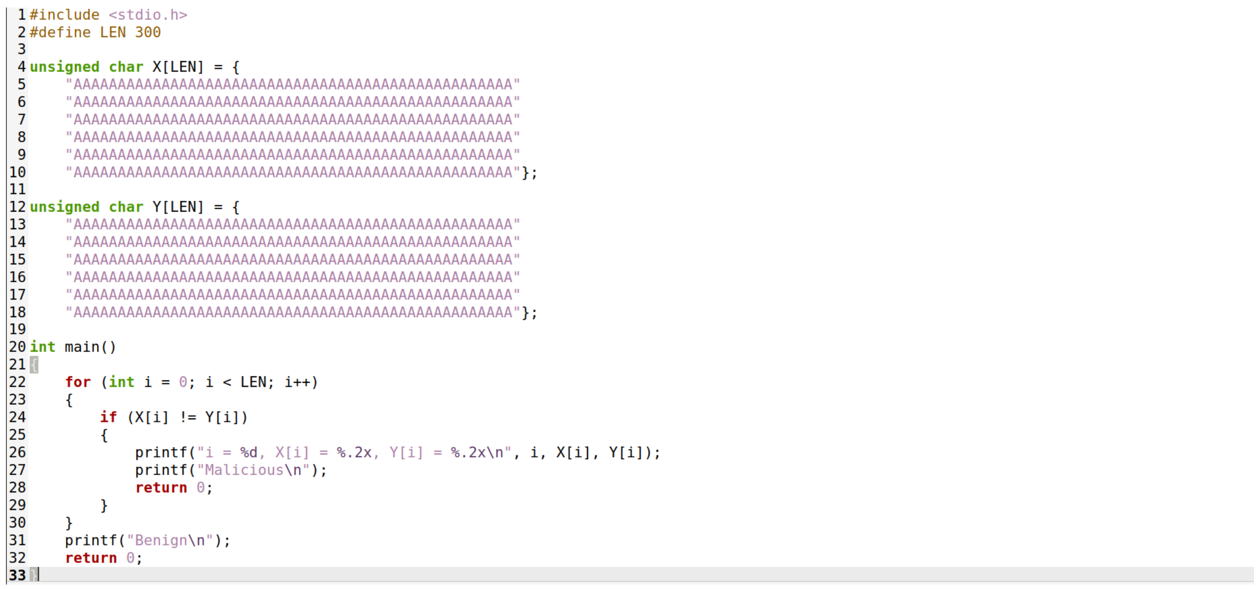
Then check if their md5sums are the same with `md5sum task3_1` & `md5sum task3_2`



```
[03/20/25]seed@VM:~/.../A3$ tail -c 128 task3_a.bin > p
[03/20/25]seed@VM:~/.../A3$ tail -c 128 task3_b.bin > q
[03/20/25]seed@VM:~/.../A3$ cat prefix p suffix > task3_1
[03/20/25]seed@VM:~/.../A3$ cat prefix q suffix > task3_2
[03/20/25]seed@VM:~/.../A3$ md5sum task3_1
2b780b58af1146503e88c110fa86e514  task3_1
[03/20/25]seed@VM:~/.../A3$ md5sum task3_2
2b780b58af1146503e88c110fa86e514  task3_2
[03/20/25]seed@VM:~/.../A3$ 
```

Task 4:

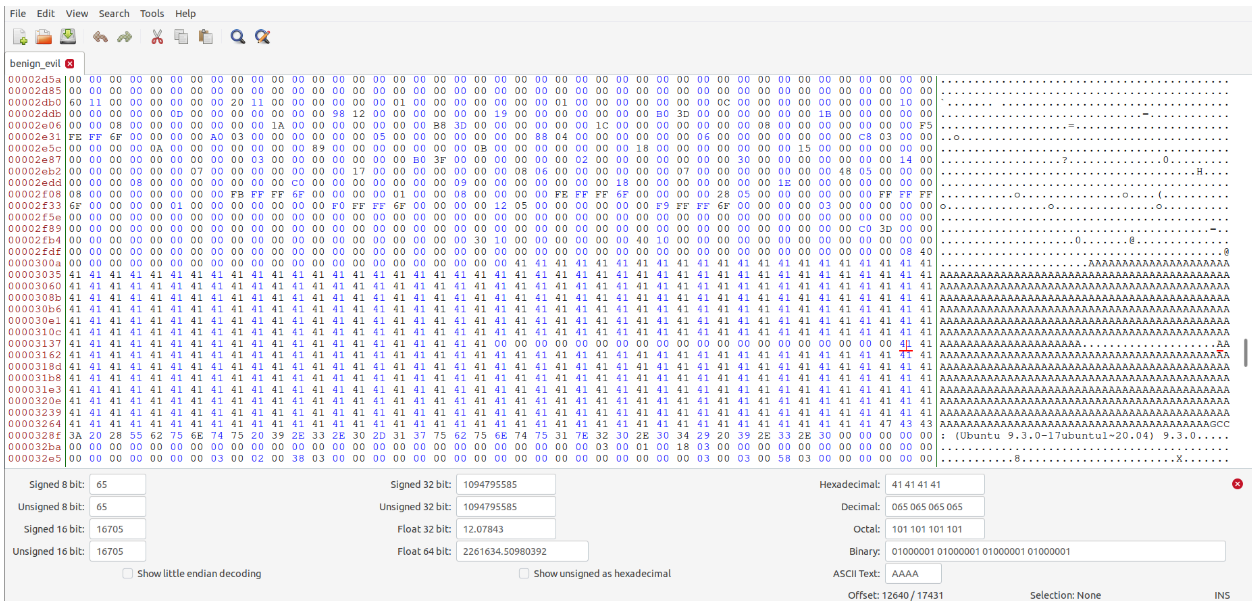
Create `benign_evil.c` then compile



```c
#include <stdio.h>
#define LEN 300

unsigned char X[LEN] = {
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"};

unsigned char Y[LEN] = {
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
    "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"};

int main()
{
    for (int i = 0; i < LEN; i++)
    {
        if (X[i] != Y[i])
        {
            printf("i = %d, X[i] = %.2x, Y[i] = %.2x\n", i, X[i], Y[i]);
            printf("Malicious\n");
            return 0;
        }
    }
    printf("Benign\n");
    return 0;
}
```

Run `bless benign_evil` and find where X and Y array starts

First array begins at 12320/17431, Second array begins at 12640/17431



```
00002d5a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...........................................
00002d85 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...........................................
00002db0 60 11 00 00 00 00 00 00 20 11 00 00 00 00 00 00 01 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 0C 00 00 00 00 00 00 00 00 10 00 `....... ..................................
00002ddb 00 00 00 00 00 0D 00 00 00 00 00 00 00 98 12 00 00 00 00 00 00 19 00 00 00 00 00 00 00 B0 3D 00 00 00 00 00 00 1B 00 00 00 00 00 ..............................=............
00002e06 00 00 08 00 00 00 00 00 00 00 1A 00 00 00 00 00 00 00 B8 3D 00 00 00 00 00 00 1C 00 00 00 00 00 00 00 08 00 00 00 00 00 00 00 F5 ...................=.......................
00002e31 FE FF 6F 00 00 00 00 A0 03 00 00 00 00 00 00 05 00 00 00 00 00 00 00 88 04 00 00 00 00 00 00 06 00 00 00 00 00 00 00 C8 03 00 00 ..o........................................
00002e5c 00 00 00 00 0A 00 00 00 00 00 00 00 89 00 00 00 00 00 00 00 0B 00 00 00 00 00 00 00 18 00 00 00 00 00 00 00 15 00 00 00 00 00 00 ...........................................
00002e87 00 00 00 00 00 00 00 00 00 03 00 00 00 00 00 00 00 B0 3F 00 00 00 00 00 00 02 00 00 00 00 00 00 00 30 00 00 00 00 00 00 00 14 00 ..................?..............0.........
00002eb2 00 00 00 00 00 00 07 00 00 00 00 00 00 00 17 00 00 00 00 00 00 00 08 06 00 00 00 00 00 00 07 00 00 00 00 00 00 00 48 05 00 00 00 ......................................H....
00002edd 00 00 00 08 00 00 00 00 00 00 00 C0 00 00 00 00 00 00 00 09 00 00 00 00 00 00 00 18 00 00 00 00 00 00 00 1E 00 00 00 00 00 00 00 ...........................................
00002f08 08 00 00 00 00 00 00 00 FB FF FF 6F 00 00 00 00 01 00 00 08 00 00 00 00 FE FF FF 6F 00 00 00 00 28 05 00 00 00 00 00 00 FF FF FF ...........o...............o....(..........
00002f33 6F 00 00 00 00 01 00 00 00 00 00 00 00 F0 FF FF 6F 00 00 00 00 12 05 00 00 00 00 00 00 F9 FF FF 6F 00 00 00 00 03 00 00 00 00 00 o...............o...............o..........
00002f5e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...........................................
00002f89 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C0 3D 00 00 ........................................=..
00002fb4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 30 10 00 00 00 00 00 00 40 10 00 00 00 00 00 00 00 00 00 00 00 00 00 ....................0.......@..............
00002fdf 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 40 ..........................................@
0000300a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ......................AAAAAAAAAAAAAAAAAAAAA
00003035 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00003060 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000308b 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000030b6 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000030e1 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000310c 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00003137 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41 41 AAAAAAAAAAAAAAAAAAAAA....................AA
00003162 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000318d 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000031b8 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000031e3 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000320e 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00003239 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00003264 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 47 43 43 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGCC
0000328f 3A 20 28 55 62 75 6E 74 75 20 39 2E 33 2E 30 2D 31 37 75 62 75 6E 74 75 31 7E 32 30 2E 30 34 29 20 39 2E 33 2E 30 00 00 00 00 00 : (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0.....
000032ba 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 01 00 18 03 00 00 00 00 00 00 00 00 00 00 00 ...........................................
000032e5 00 00 00 00 00 00 00 03 00 02 00 38 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 03 00 58 03 00 00 00 00 00 00 ...........8.......................X.......
```

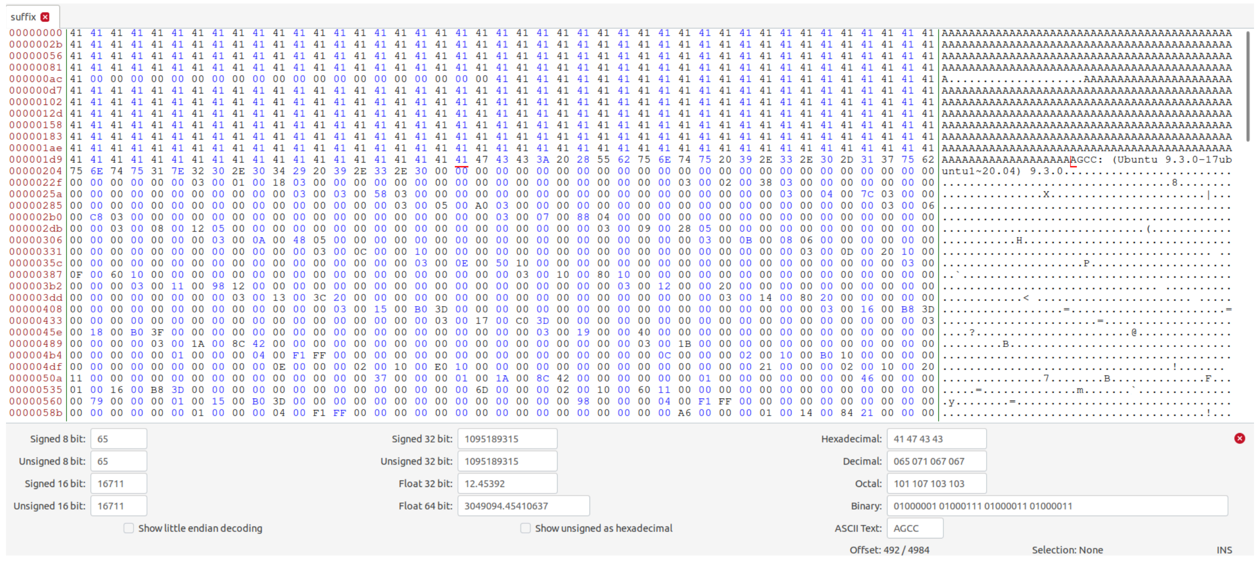
```
  Signed 8 bit: 65          Signed 32 bit: 1094795585        Hexadecimal: 41 41 41 41
Unsigned 8 bit: 65        Unsigned 32 bit: 1094795585            Decimal: 065 065 065 065
 Signed 16 bit: 16705        Float 32 bit: 12.07843                Octal: 101 101 101 101
Unsigned 16 bit: 16705       Float 64 bit: 2261634.50980392       Binary: 01000001 01000001 01000001 01000001
     Show little endian decoding    Show unsigned as hexadecimal    ASCII Text: AAAA
                                             Offset: 12640 / 17431     Selection: None     INS
```

Obtain prefix and suffix from benign_evil

```
[03/23/25]seed@VM:~/.../A3$ head -c 12320 benign_evil > prefix
[03/23/25]seed@VM:~/.../A3$ tail -c +12448 benign_evil > suffix
```

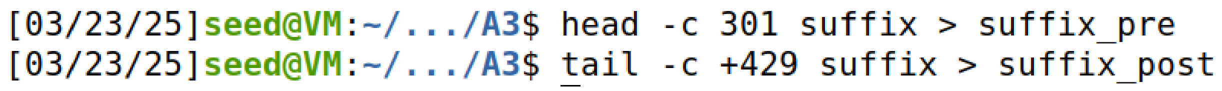
```
[03/23/25]seed@VM:~/.../A3$ md5collgen -p prefix -o s1 s2
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 's1' and 's2'
Using prefixfile: 'prefix'
Using initial value: ce32bebe2f1f2eefe11981ae5bdd94e3

Generating first block: ....
Generating second block: W......
Running time: 5.40198 s
```

Run `tail -c 128 s1 > p` and `tail -c 128 s2 > Q` then generate the 2 halves of the suffix
`Y` starts from `0xc1`(301) in `suffix`, So if we want to make `X=Y`, we should replace offset [301,429) with the same P or Q generated above



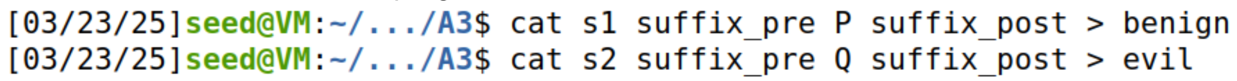
```
00000000 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000002b 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000056 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000081 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000000ac 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 A....................AAAAAAAAAAAAAAAAAAAAAA
000000d7 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000102 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000012d 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000158 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000183 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000001ae 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000001d9 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 47 43 43 3A 20 28 55 62 75 6E 74 75 20 39 2E 33 2E 30 2D 31 37 75 62 AAAAAAAAAAAAAAAAAAAAGCC: (Ubuntu 9.3.0-17ub
00000204 75 6E 74 75 31 7E 32 30 2E 30 34 29 20 39 2E 33 2E 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 untu1~20.04) 9.3.0.........................
0000022f 00 00 00 00 00 00 03 00 01 00 18 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 02 00 38 03 00 00 00 00 00 00 00 ..................................8........
0000025a 00 00 00 00 00 00 00 00 00 00 00 03 00 03 00 58 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04 00 7C 03 00 00 ...............X.......................|...
00000285 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 05 00 A0 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 06 ...........................................
000002b0 00 C8 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 07 00 88 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...........................................
000002db 00 00 03 00 08 00 12 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 09 00 28 05 00 00 00 00 00 00 00 00 00 00 00 ..............................(............
00000306 00 00 00 00 00 00 00 03 00 0A 00 48 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 0B 00 08 06 00 00 00 00 00 00 ...........H...............................
00000331 00 00 00 00 00 00 00 00 00 00 00 00 03 00 0C 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 0D 00 20 10 00 ........................................ ..
0000035c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 0E 00 50 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 .....................P.....................
00000387 0F 00 60 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 10 00 80 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..`........................................
000003b2 00 00 00 03 00 11 00 98 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 12 00 00 20 00 00 00 00 00 00 00 00 00 00 ............................... ...........
000003dd 00 00 00 00 00 00 00 00 03 00 13 00 3C 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 14 00 80 20 00 00 00 00 00 ............< ....................... .....
00000408 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 15 00 B0 3D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 16 00 B8 3D ..................=.......................=
00000433 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 17 00 C0 3D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 .......................=...................
0000045e 00 18 00 B0 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 19 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....?.......................@..............
00000489 00 00 00 00 03 00 1A 00 8C 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 1B 00 00 00 00 00 00 00 00 00 00 00 00 .........B.................................
000004b4 00 00 00 00 00 01 00 00 00 04 00 F1 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C 00 00 00 02 00 10 00 B0 10 00 00 00 ...........................................
000004df 00 00 00 00 00 00 00 00 00 00 0E 00 00 00 02 00 10 00 E0 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 21 00 00 00 02 00 10 00 20 ..................................!.......
0000050a 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 37 00 00 00 01 00 1A 00 8C 42 00 00 00 00 00 00 01 00 00 00 00 00 00 00 46 00 00 00 ...............7........B..............F...
00000535 01 00 16 00 B8 3D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6D 00 00 00 02 00 10 00 60 11 00 00 00 00 00 00 00 00 00 00 00 00 00 .....=..............m.......`..............
00000560 00 79 00 00 00 01 00 15 00 B0 3D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 98 00 00 00 04 00 F1 FF 00 00 00 00 00 00 00 00 00 00 .y........=................................
0000058b 00 00 00 00 00 00 01 00 00 00 04 00 F1 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A6 00 00 00 01 00 14 00 84 21 00 00 00 .......................................!...
```

```
Signed 8 bit: 65                 Signed 32 bit: 1095189315          Hexadecimal: 41 47 43 43
Unsigned 8 bit: 65               Unsigned 32 bit: 1095189315        Decimal: 065 071 067 067
Signed 16 bit: 16711             Float 32 bit: 12.45392             Octal: 101 107 103 103
Unsigned 16 bit: 16711           Float 64 bit: 3049094.45410637     Binary: 01000001 01000111 01000011 01000011
    Show little endian decoding      Show unsigned as hexadecimal   ASCII Text: AGCC
                                                                    Offset: 492 / 4984     Selection: None     INS
```



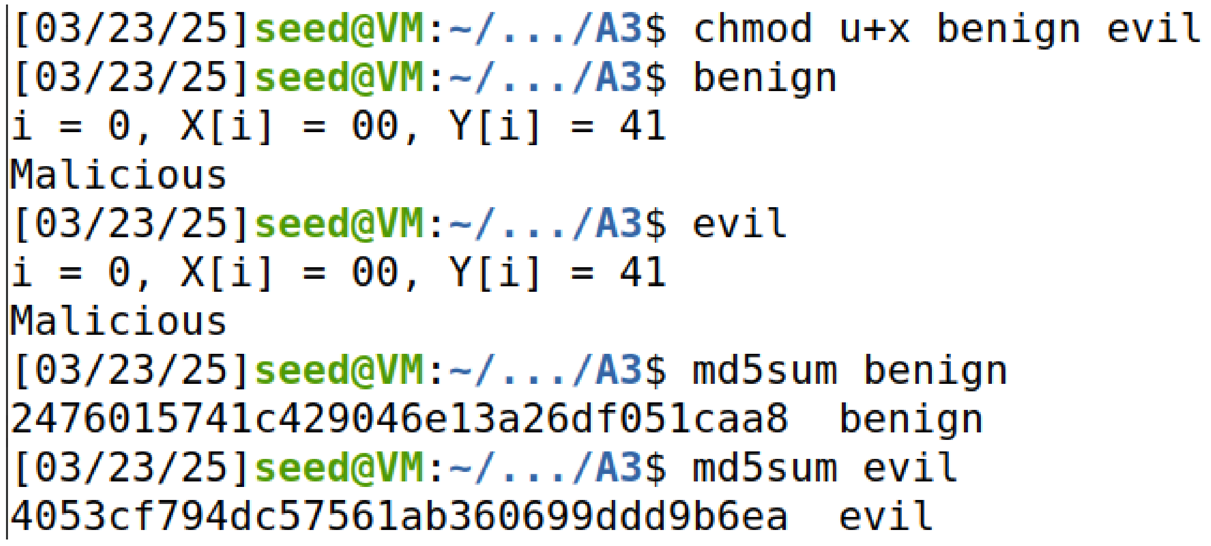
```
[03/23/25]seed@VM:~/.../A3$ head -c 301 suffix > suffix_pre
[03/23/25]seed@VM:~/.../A3$ tail -c +429 suffix > suffix_post
```

Construct the final executable programs and make them executable:



```
[03/23/25]seed@VM:~/.../A3$ cat s1 suffix_pre P suffix_post > benign
[03/23/25]seed@VM:~/.../A3$ cat s2 suffix_pre Q suffix_post > evil
```

Couldn't get this task to work



```
[03/23/25]seed@VM:~/.../A3$ chmod u+x benign evil
[03/23/25]seed@VM:~/.../A3$ benign
i = 0, X[i] = 00, Y[i] = 41
Malicious
[03/23/25]seed@VM:~/.../A3$ evil
i = 0, X[i] = 00, Y[i] = 41
Malicious
[03/23/25]seed@VM:~/.../A3$ md5sum benign
2476015741c429046e13a26df051caa8  benign
[03/23/25]seed@VM:~/.../A3$ md5sum evil
4053cf794dc57561ab360699ddd9b6ea  evil
```

Attempted multiple times to build the executable files. Couldn't match up both benign and malicious activity and also have the same hash.