



Lab 5

BÁO CÁO BÀI THỰC HÀNH SỐ 5 Kỹ thuật dịch ngược (tt)

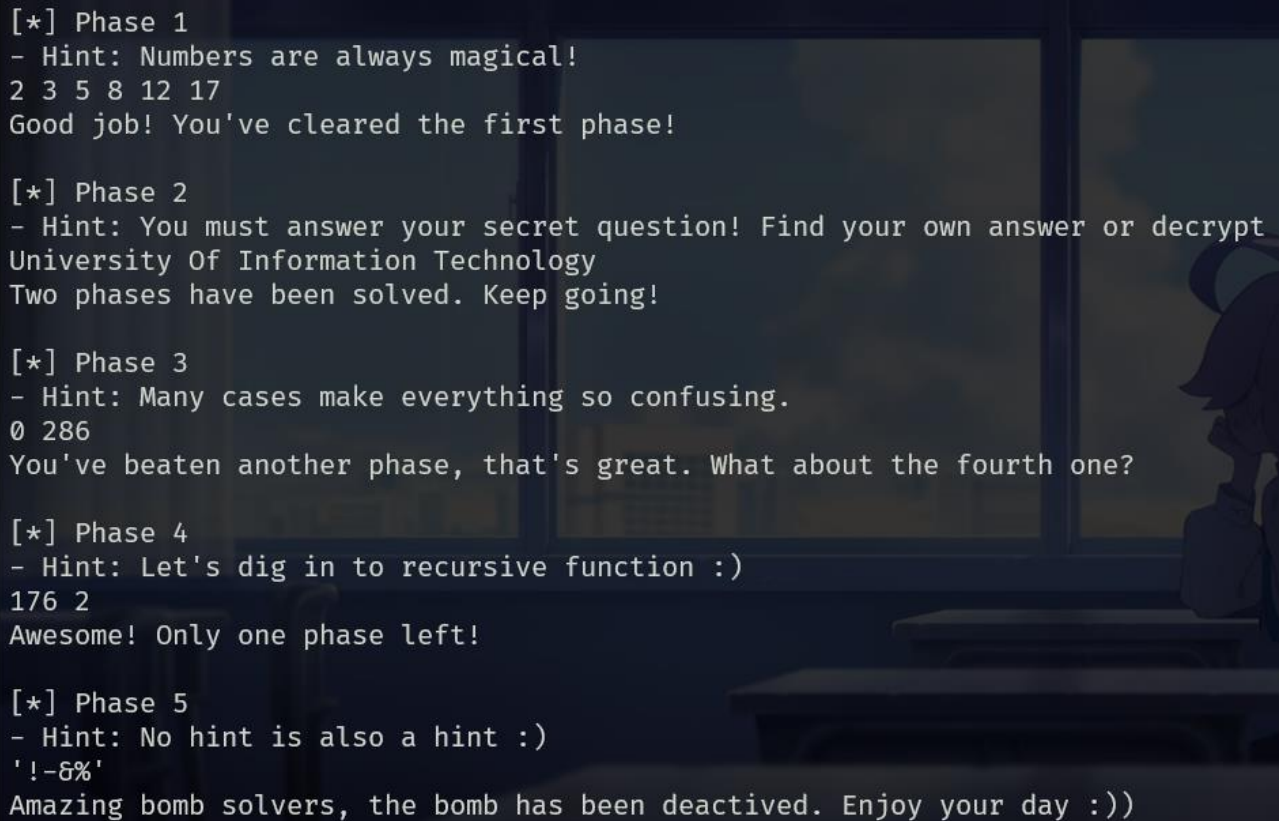
Môn học: Lập trình hệ thống

Giảng viên hướng dẫn	ThS. Đỗ Thị Hương Lan
Sinh viên thực hiện	Nguyễn Tuấn Phát (22521076) Đặng Chí Thịnh (22521399) Trần Hoàng Tuấn Kiệt (22520724)
Mức độ hoàn thành	Hoàn thành (5/5)

Kết quả đạt được

- Số phase hoàn thành: 5/5
- Các phase có nhiều đáp án: 3, 4, 5
- Kết quả tham khảo:
 - + Phase 1: 2 3 5 8 12 17
 - + Phase 2: University Of Information Technology
 - + Phase 3: 0 286
 - + Phase 4: 176 2
 - + Phase 5: '!-&%'

* Minh chứng:



```
[*] Phase 1
- Hint: Numbers are always magical!
2 3 5 8 12 17
Good job! You've cleared the first phase!

[*] Phase 2
- Hint: You must answer your secret question! Find your own answer or decrypt
University Of Information Technology
Two phases have been solved. Keep going!

[*] Phase 3
- Hint: Many cases make everything so confusing.
0 286
You've beaten another phase, that's great. What about the fourth one?

[*] Phase 4
- Hint: Let's dig in to recursive function :)
176 2
Awesome! Only one phase left!

[*] Phase 5
- Hint: No hint is also a hint :)
'!-&%'
Amazing bomb solvers, the bomb has been deactivated. Enjoy your day :))
```

PHASE 1

```
puts("\n[*] Phase 1\n- Hint: Numbers are always magical!");  
line = read_line();  
phase1(line);  
defuse_bomb();
```

```
1 int __cdecl phase1(int a1)  
2 {  
3     int result; // eax  
4     int v2[6]; // [esp+Ch] [ebp-2Ch] BYREF  
5     int v3[6]; // [esp+10h] [ebp-28h] BYREF  
6     int v4[6]; // [esp+14h] [ebp-24h] BYREF  
7     int v5[6]; // [esp+18h] [ebp-20h] BYREF  
8     int v6[6]; // [esp+1Ch] [ebp-1Ch] BYREF  
9     _DWORD v7[2]; // [esp+20h] [ebp-18h] BYREF  
10    int v8; // [esp+28h] [ebp-10h]  
11    int i; // [esp+2Ch] [ebp-Ch]  
12  
13    v8 = __isoc99_sscanf(a1, "%d %d %d %d %d %d", v2, v3, v4, v5, v6, v7);  
14    if ( v8 != 6 )  
15        explode_bomb();  
16    v7[1] = 2;  
17    result = v2[0];  
18    if ( v2[0] != 2 )  
19        explode_bomb();  
20    for ( i = 1; i <= 5; ++i )  
21    {  
22        result = v2[i];  
23        if ( result != v2[i - 1] + i )  
24            explode_bomb();  
25    }  
26    return result;  
27 }
```

- input: chuỗi kí tự các số cách nhau bởi dấu cách, số lượng 6 số (line 13) và bị ràng buộc phải nhập đủ 6 số và số đầu tiên nhập vào phải bằng 2.

```
if ( v8 != 6 )  
    explode_bomb();
```

```
if ( v2[0] != 2 )  
    explode_bomb();
```

- line 20-25 kiểm input nhập vào với điều kiện số có index thứ i phải bằng số có index $(i - 1) + i$

- Ta viết code python để tìm chuỗi có số đầu tiên là 2 và thỏa điều kiện trên, chạy với index từ 1->5

```

1  number = 2
2  res = "" + str(number)
3
4  for i in range(1,6):
5      newNum = number + i # v2[i - 1] + i
6      res += ' ' + str(newNum)
7      number = newNum
8
9  print(res)

```

```

● PS D:\Mon hoc\HK4\LTHT\SystemProgramming\Lab5> python -u "d:\
2 3 5 8 12 17

```

- Vậy chuỗi cần nhập vào là "2 3 5 8 12 17".

```

[*] Phase 1
- Hint: Numbers are always magical!
2 3 5 8 12 17
Good job! You've cleared the first phase!

```

- Do số đầu tiên bị fixed chỉ có thể = 2 nên sẽ chỉ có 1 input hợp lệ.

PHASE 2

```
puts(
    "\n"
    "[*] Phase 2\n"
    "- Hint: You must answer your secret question! Find your own answer or decrypt the answer saved in this file.");
v4 = read_line();
phase2(v4);
defuse_bomb();
```

```
1 int __cdecl phase2(int a1)
2 {
3     int result; // eax
4     char *s1; // [esp+Ch] [ebp-1Ch]
5     char *s2; // [esp+10h] [ebp-18h]
6
7     s2 = ANSWERS[QA_MAP[17]];
8     s1 = (char *)transfer(a1);
9     if ( !*s2 || (result = is_equal(s1, s2)) == 0 )
10         explode_bomb();
11     return result;
12 }
```

- Input: chuỗi kí tự, không giới hạn độ dài, các kí tự. Input là câu trả lời cho các câu hỏi lưu ở QUESTIONS

```
.data:004B0000 public QUESTIONS
.data:004B0060 QUESTIONS dd offset aMyVehicleRegis
.data:004B0060 ; DATA XREF: phase2+10↑r
.data:004B0060 ; "My vehicle registration plate starts wi"...
.data:004B0064 dd offset aWhatIsTheCapit ; "What is the capital of Thailand?"
.data:004B0068 dd offset aWhatIsYourMajo ; "What is your major in English? (Capital"..."
.data:004B006C dd offset aWhichSeasonHas ; "Which season has cherry blossoms?"
.data:004B0070 dd offset aThanksToMeYouC ; "Thanks to me, you can see straight thro"...
.data:004B0074 dd offset aWhichCountryIs ; "Which country is the Lion city in South"..."
.data:004B0078 dd offset aWhatIsTheMainL ; "What is the main language used in this "..."
.data:004B007C dd offset aEnterTheCurren ; "Enter the current date using the format"..."
.data:004B0080 dd offset aWhichProvinceI ; "Which province in Vietnam has the most "..."
.data:004B0084 dd offset aIAmAnOddNumber ; "I am an odd number. Take away one lette"..."
.data:004B0088 dd offset aWhatWordIsSpel ; "What word is spelled incorrectly in eve"..."
.data:004B008C dd offset aWhatIsYourNati ; "What is your nationality?"
.data:004B0090 dd offset aWhatIsThePhone ; "What is the phone number of our univers"..."
.data:004B0094 dd offset aWhatIsTheNameO ; "What is the name of the analyzed execut"..."
.data:004B0098 dd offset aWhichCityHas37 ; "Which city has 3/7 of a chicken and 2/3"..."
.data:004B009C dd offset aInUitWhichMajo ; "In UIT, which major tops the list alpha"..."
.data:004B00A0 dd offset aIsabellaSParen ; "Isabella's parents have four children. "..."
.data:004B00A4 dd offset aWhatIsTheFulle ; "What is the full English name of our un"..."
.data:004B00A8 dd offset aWhichFastFoodB ; "Which fast food brand features a bee in"..."
.data:004B00AC dd offset aCompleteTheDom ; "Complete the domain of our faculty: htt"..."
.data:004B00B0 dd offset aWhatIsTheLarge ; "What is the largest country in the worl"..."
.data:004B00B4 dd offset aWhatIsTheEstab ; "What is the establishment date of UIT ("..."
.data:004B00B8 dd offset aWhichContinent ; "Which continent has the least populatio"..."
.data:004B00BC dd offset aWhatIsTheLonge ; "What is the longest wall in the world? "..."
.data:004B00C0 dd offset aWhatAreFaxEbx ; "What are eax, ebx, eip, esi on a comput"..."
.data:004B00C4 dd offset aWhatIsTheSmall ; "What is the smallest planet in the Sola"..."
.data:004B00C8 dd offset aWhatIsTheDomai ; "What is the domain of UIT Students' ema"..."
.data:004B00CC dd offset aWhatIsTheVietn ; "What is the Vietnamese name (without ac"..."
.data:004B00D0 dd offset aWhichIsTheLarg ; "Which is the largest net in the world?"
```

- Phân tích `s2 = ANSWERS[QA_MAP[17]]`, với câu hỏi ở vị trí 17 trong `QUESTIONS`, ta có

```
set aWhatIsTheFullE ; "What is the full English name of our un ...  
fset aWhichFastFoodR ; "Which fast food brand features a bee in"  
fset aCompleteTh aWhatIsTheFullE db 'What is the full English name of our university? (Capitalize the '  
fset aWhatIsTheL ; DATA XREF: .data:0804B0A4+0  
fset aWhatIsTheE db 'first char of each word)',0  
fset aWhichContinent ; "Which continent has the least populatio"...
```

- Đọc qua câu hỏi ta sẽ biết được đáp án là: “University Of Information Technology”, nhưng giả sử chúng ta không biết đáp án.

- Xem qua `ANSWERS` – nơi lưu đáp án cho các câu hỏi, ở index 17 ta có được đáp án nhưng đã bị mã hóa

```
db 0  
public ANSWERS  
dd offset aFmrlHysrk ; DATA XREF: phase2+2A↑r  
; "Fmrl Hysrk"  
dd offset aFerkoso ; "Ferkoso"  
dd offset aMrjsvqexmsrWig ; "Mrjsvqexmsr Wigyvmmc"  
dd offset aWtvrmrk ; "Wtvrmrk"  
dd offset aAmrhsa ; "Amrhsa"  
dd offset aWmrketsvi ; "Wmrketsvi"  
dd offset aZmixreqiwi ; "Zmixreqiwi"  
dd offset a496468 ; "49/6468"  
dd offset aFmrlHysrk ; "Fmrl Hysrk"  
dd offset aWizir ; "Wizir"  
dd offset aMrgsvvigxpc ; "Mrgsvvigxpc"  
dd offset aZmixreqiwi ; "Zmixreqiwi"  
dd offset a46271696446 ; "46271696446"  
dd offset aRx643Ymxfsqf ; "rx643-ymx-fsqf"  
dd offset aGlmgeks ; "Glmgeks"  
dd offset aEvxmjmgmepMrxi ; "Evxmjmgmep Mrxippmkingi"  
dd offset aMwefippe ; "Mwefippe"  
dd offset aYrmzivvmxcSjMr ; "Yrmzivvmxc Sj Mrjsvqexmsr Xiglrspskc"  
dd offset aNsppmfii ; "Nsppmfii"  
dd offset aRgYmxIhyZr aYrmzivvmxcSjMr db 'Yrmzivvmxc Sj Mrjsvqexmsr Xiglrspskc',0  
dd offset aVywwme ; DATA XREF: .data:0804B1A4+0  
dd offset a42406440 ; "42406440"
```

- Trong hàm `phase2()`, ta có thể nhận thấy `s2` lấy đáp án đã bị mã hóa, `s1` thì lấy giá trị của input chúng ta nhập vào sau khi biến đổi qua hàm `transfer()`.

```

1 int __cdecl transfer(int a1)
2 {
3     char v2; // [esp+Ah] [ebp-6h]
4     char v3; // [esp+Bh] [ebp-5h]
5     int i; // [esp+Ch] [ebp-4h]
6
7     for ( i = 0; *(_BYTE *)(i + a1); ++i )
8     {
9         v3 = *(_BYTE *)(i + a1);
10        if ( (v3 <= 96 || v3 > 122) && (v3 <= 64 || v3 > 90) )
11        {
12            if ( v3 > 47 && v3 <= 57 )
13                v3 = (v3 - 48 + 4) % 10 + 48;
14        }
15        else
16        {
17            if ( v3 <= 96 )
18                v2 = 65;
19            else
20                v2 = 97;
21            v3 = (v3 - v2 + 4) % 26 + v2;
22        }
23        *(_BYTE *)(a1 + i) = v3;
24    }
25    return a1;
26 }

```

- Để tìm được string đã được decoded ta cần đảo ngược một vài phép
 - + line 13: $v3 = (v3 - 48 - 4) \% 10 + 48$
 - + line 21: $v3 = (v3 - v2 - 4) \% 26 + v2$
- Viết file python để thực hiện decrypt:

```
1  encoded = "Yrmzivwmxs Sj Mrjsvqexmsr Xiglrspskc"
2
3  origin_str = ""
4
5  for i in range(0, len(encoded)):
6      v3 = ord(encoded[i])
7      if (v3 <= 96 or v3 > 122) and (v3 <= 64 or v3 > 90):
8          # Check if v3 is a digit
9          if 47 < v3 <= 57:
10             v3 = (v3 - 48 - 4) % 10 + 48
11         else:
12             # v3 is a letter
13             if v3 <= 96:
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL

● PS D:\Mon hoc\HK4\LTHT\SystemProgramming\Lab5> python -u "d:\Mon hoc\HK4\LTHT\University Of Information Technology

- Vậy chuỗi cần nhập: "University Of Information Technology"

```
[*] Phase 2
- Hint: You must answer your secret question! Find your own answer
University Of Information Technology
Two phases have been solved. Keep going!
```

- Tương tự bài 1, bài 2 này chỉ có 1 đáp án hợp lệ do chỉ có 1 câu hỏi cùng với 1 câu trả lời tương ứng.

PHASE 3

```
1 int __cdecl phase3(int a1)
2 {
3     int result; // eax
4     int v2; // [esp+0h] [ebp-18h] BYREF
5     int v3; // [esp+4h] [ebp-14h] BYREF
6     int v4; // [esp+8h] [ebp-10h]
7     int v5; // [esp+Ch] [ebp-Ch]
8
9     v5 = 0;
10    v4 = 0;
11    v4 = __isoc99_sscanf(a1, "%d %d", &v3, &v2);
12    if ( v4 <= 1 )
13        explode_bomb();
14    switch ( v3 )
15    {
16        case 0:
17            v5 = 286;
18            break;
19        case 1:
20            v5 = 516;
21            break;
22        case 2:
23            v5 = 266;
24            break;
25        case 3:
26            v5 = 860;
27            break;
28        case 4:
29            v5 = 988;
30            break;
31        case 5:
32            v5 = 178;
33            break;
34        case 6:
35            v5 = 948;
36            break;
37        case 7:
38            v5 = 203;
39            break;
40        default:
41            explode_bomb();
42    }
43    result = v2;
44    if ( v5 != v2 )
45        explode_bomb();
46    return result;
47 }
```

- input: chuỗi kí tự các số cách nhau bởi dấu cách, số lượng 2 số (line 11) và bị ràng buộc phải nhập đủ 2 số (line 12). Số đầu tiên nhập vào được lưu vào v3, số thứ 2 lưu ở v2.

- Dựa vào các case trong switch ta có thể biết được v3 (số đầu tiên nhập vào) chỉ có thể có giá trị từ 0->7, tùy vào v3 nhập vào mà v2 phải có giá trị tương ứng

- Theo Switch case ta có thể có **8 trường hợp input** nhập vào hợp lệ:
{0, 286}, {1, 516}, {2, 266}, {3, 860}, {4, 988}, {5, 178}, {6, 948}, {7, 203}

{0, 286}

```
[*] Phase 3
- Hint: Many cases make everything so confusing.
0 286
You've beaten another phase, that's great. What about the fourth one?
```

{7, 203}

```
[*] Phase 3
- Hint: Many cases make everything so confusing.
7 203
You've beaten another phase, that's great. What about the fourth one?
```

{4, 988}

```
[*] Phase 3
- Hint: Many cases make everything so confusing.
4 988
You've beaten another phase, that's great. What about the fourth one?
```

PHASE 4

```
1 int __cdecl phase4(int a1)
2 {
3     int result; // eax
4     int v2; // [esp+Ch] [ebp-1Ch] BYREF
5     int v3; // [esp+10h] [ebp-18h] BYREF
6     int v4; // [esp+14h] [ebp-14h]
7     int v5; // [esp+18h] [ebp-10h]
8     int v6; // [esp+1Ch] [ebp-Ch]
9
10    v6 = __isoc99_sscanf(a1, "%d %d", &v2, &v3);
11    if ( v6 != 2 || v3 <= 1 || v3 > 4 )
12        explode_bomb();
13    v5 = 9;
14    v4 = func4(9, v3);
15    result = v2;
16    if ( v4 != v2 )
17        explode_bomb();
18    return result;
19 }
```

- Input: Nhập vào 2 số, lưu lần lượt ở v2, v3, cách nhau bởi dấu cách. Bị ràng buộc phải nhập đủ 2 số và số thứ 2 nhập vào phải > 1 hoặc <= 4 (line 11) hay số thứ 2 chỉ có thể có giá trị là {2, 3}

- Hàm phase4() sẽ tính v4 qua function func4(9, v3) và so sánh với v2 (số thứ nhất nhập vào) để xét xem bom có nổ hay không.

```
1 int __cdecl func4(int a1, int a2)
2 {
3     int v3; // ebx
4
5     if ( a1 <= 0 )
6         return 0;
7     if ( a1 == 1 )
8         return a2;
9     v3 = func4(a1 - 1, a2) + a2;
10    return v3 + func4(a1 - 2, a2);
11 }
```

- Nhìn sơ qua hàm và dựa vào hint của đề, ta có thể nhận thấy được đây là hàm tính đệ quy với tham số $a1 = 9$ và $a2 = v3$.
- Ta viết code python dựa theo pseudo code để tính giá trị trả về của hàm khi $v3 = 2$ và $v3 = 3$

```

4.py > func
1  def func(a1, a2):
2      if (a1 <= 0):
3          return 0
4      if (a2 == 1):
5          return a2
6      v3 = func(a1 - 1, a2) + a2
7      return v3 + func(a1 - 2, a2)
8
9
10 for i in [2, 3]:
11     print(f"i = {i} => func() = {func(9,i)}")

```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL

```

● PS D:\Mon hoc\HK4\LTHT\SystemProgramming\Lab5> python -u "d:\Mon hoc\HK4\LTHT\SystemF
i = 2 => func() = 176
i = 3 => func() = 264
○ PS D:\Mon hoc\HK4\LTHT\SystemProgramming\Lab5> 

```

- Vậy có **2 trường hợp input hợp lệ**: {176, 2} và {264, 3}

```

[*] Phase 4
- Hint: Let's dig in to recursive function :)
176 2
Awesome! Only one phase left!

```

```

[*] Phase 4
- Hint: Let's dig in to recursive function :)
264 3
Awesome! Only one phase left!

```

PHASE 5

```
1 int __cdecl phase5(char *s)
2 {
3     int result; // eax
4     char s1[7]; // [esp+1h] [ebp-17h] BYREF
5     size_t v3; // [esp+8h] [ebp-10h]
6     int i; // [esp+Ch] [ebp-Ch]
7
8     v3 = strlen(s);
9     if ( v3 != 6 )
10         explode_bomb();
11     for ( i = 0; i <= 5; ++i )
12         s1[i] = array_3852[s[i] & 0xF];
13     s1[6] = 0;
14     result = is_equal(s1, "sabres");
15     if ( !result )
16         explode_bomb();
17     return result;
18 }
```

- Input: Nhập vào một chuỗi kí tự, ràng buộc phải có đủ 6 kí tự (line 9)

- Hàm phase5() sẽ tính một chuỗi mới s1 với các giá trị lưu trong array_3852 với các index [s[i] & 0xF] với i từ 0->5. Sau đó so sánh với chuỗi “sabres” để xét xem bom nổ hay không.

```
.data:0804B1E8 ; char array_3852[16]
.data:0804B1E8 array_3852      db 6Dh                ; DATA XREF: phase5+3C↑r
.data:0804B1E9                db  61h ; a
.data:0804B1EA                db  64h ; d
.data:0804B1EB                db  75h ; u
.data:0804B1EC                db  69h ; i
.data:0804B1ED                db  65h ; e
.data:0804B1EE                db  72h ; r
.data:0804B1EF                db  73h ; s
.data:0804B1F0                db  6Eh ; n
.data:0804B1F1                db  66h ; f
.data:0804B1F2                db  6Fh ; o
.data:0804B1F3                db  74h ; t
.data:0804B1F4                db  76h ; v
.data:0804B1F5                db  62h ; b
.data:0804B1F6                db  79h ; y
.data:0804B1F7                db  6Ch ; l
.data:0804B1F7 _data          ends
```

- Phân tích char array_3852[16] ta được string sau: “**maduiersnfotvb**yl”

- Do index của array_3852 ở line 12 được tính bằng $(s[i] \& 0xF)$, nên sẽ tồn tại rất nhiều $s[i]$ cho ra cùng một kết quả index.

VD: 'A' với 'Q' có mã ascii là 65 và 81, đều cho ra index là 1

- Vậy nên sẽ tồn tại rất nhiều tổ hợp $s[0] \rightarrow s[6]$ có thể thỏa đáp án.

- Dựa vào pseudo code, array_3852 và string cuối cần so sánh "sabres", ta viết code python để tìm những trường hợp của input của thể thỏa mãn yêu cầu:

```
1 str_map = "maduiersnfotvbyl"
2 s1 = "sabres"
3
4 def find_possible_s_values(str_map, s1):
5     # Convert str_map to array_3852
6     array_3852 = list(str_map)
7
8     possible_s_values = []
9
10    for char in s1:
11        # Find the indices in array_3852 that match the character
12        indices = [i for i, x in enumerate(array_3852) if x == char]
13
14        s_char_values = []
15        for base in indices:
16            for higher_bits in range(16): # Higher bits can be 0-15 for 4 bits
17                s_char_values.append((higher_bits << 4) | base)
18
19        possible_s_values.append(s_char_values)
20
21    return possible_s_values
22
23 possible_s_values = find_possible_s_values(str_map, s1)
24 for i, values in enumerate(possible_s_values):
25     print(f"Possible values for s[{i}] corresponding to s1[{i}] = '{s1[i]}': {values}")
```



```
Possible values for s[0] corresponding to s1[0] = 's': [7, 23, 39, 55, 71, 87, 103, 119, 135, 151, 167, 183, 199, 215, 231, 247]
Possible values for s[1] corresponding to s1[1] = 'a': [1, 17, 33, 49, 65, 81, 97, 113, 129, 145, 161, 177, 193, 209, 225, 241]
Possible values for s[2] corresponding to s1[2] = 'b': [13, 29, 45, 61, 77, 93, 109, 125, 141, 157, 173, 189, 205, 221, 237, 253]
Possible values for s[3] corresponding to s1[3] = 'r': [6, 22, 38, 54, 70, 86, 102, 118, 134, 150, 166, 182, 198, 214, 230, 246]
Possible values for s[4] corresponding to s1[4] = 'e': [5, 21, 37, 53, 69, 85, 101, 117, 133, 149, 165, 181, 197, 213, 229, 245]
Possible values for s[5] corresponding to s1[5] = 's': [7, 23, 39, 55, 71, 87, 103, 119, 135, 151, 167, 183, 199, 215, 231, 247]
```

$s1[0] = 's': [7, 23, 39, 55, 71, 87, 103, 119, 135, 151, 167, 183, 199, 215, 231, 247]$

$s1[1] = 'a': [1, 17, 33, 49, 65, 81, 97, 113, 129, 145, 161, 177, 193, 209, 225, 241]$

$s1[2] = 'b': [13, 29, 45, 61, 77, 93, 109, 125, 141, 157, 173, 189, 205, 221, 237, 253]$

$s1[3] = 'r': [6, 22, 38, 54, 70, 86, 102, 118, 134, 150, 166, 182, 198, 214, 230, 246]$

$s1[4] = 'e': [5, 21, 37, 53, 69, 85, 101, 117, 133, 149, 165, 181, 197, 213, 229, 245]$

$s1[5] = 's': [7, 23, 39, 55, 71, 87, 103, 119, 135, 151, 167, 183, 199, 215, 231, 247]$

- Chạy code python xong ta sẽ có được mã ascii của từng ký tự có thể thỏa mãn yêu cầu, mình sẽ thử với 3 đáp án khác nhau.

1.

Ascii codes: 39 113 141 150 155

String from ascii: '!-&%'

```
[*] Phase 5
- Hint: No hint is also a hint :)
'!-&%'
Amazing bomb solvers, the bomb has been deactivated. Enjoy your day :))
```

2.

Ascii codes: 119 33 109 118 117 119

String from ascii: w!mvuw

```
[*] Phase 5
- Hint: No hint is also a hint :)
w!mvuw
Amazing bomb solvers, the bomb has been deactivated. Enjoy your day :))
```

3.

Ascii codes: 55 49 45 54 53 55

String from ascii: 71-657

```
[*] Phase 5
- Hint: No hint is also a hint :)
71-657
Amazing bomb solvers, the bomb has been deactivated. Enjoy your day :))
```

Code Python

Phase 1:

```
number = 2
res = "" + str(number)

for i in range(1,6):
    newNum = number + i # v2[i - 1] + i
    res += ' ' + str(newNum)
    number = newNum

print(res)
```

Phase 2:

```
encoded = "Yrmzivvmxc Sj Mrjsvqexmsr Xiglrspskc"

origin_str = ""

for i in range(0, len(encoded)):
    v3 = ord(encoded[i])
    if (v3 <= 96 or v3 > 122) and (v3 <= 64 or v3 > 90):
        # Check if v3 is a digit
        if 47 < v3 <= 57:
            v3 = (v3 - 48 + 4) % 10 + 48
        else:
            # v3 is a Letter
            if v3 <= 96:
                v2 = 65 # 'A'
            else:
                v2 = 97 # 'a'
            v3 = (v3 - v2 + 4) % 26 + v2
    origin_str += chr(v3)

print (origin_str)
```


Phase 4:

```
def func(a1, a2):
    if (a1 <= 0):
        return 0
    if (a2 == 1):
        return a2
    v3 = func(a1 - 1, a2) + a2
    return v3 + func(a1 - 2, a2)

for i in [2, 3]:
    print(f"i = {i} => func() = {func(9,i)}")
```

Phase 5:

```
str_map = "maduiersnfotvbyl"
s1 = "sabres"

def find_possible_s_values(str_map, s1):
    # Convert str_map to array_3852
    array_3852 = list(str_map)

    possible_s_values = []

    for char in s1:
        # Find the indices in array_3852 that match the character
        indices = [i for i, x in enumerate(array_3852) if x == char]

        s_char_values = []
        for base in indices:
            for higher_bits in range(16): # Higher bits can be 0-15 for 4 bits
                s_char_values.append((higher_bits << 4) | base)

        possible_s_values.append(s_char_values)

    return possible_s_values

possible_s_values = find_possible_s_values(str_map, s1)
for i, values in enumerate(possible_s_values):
    print(f"Possible values for s[{i}] corresponding to s1[{i}] = '{s1[i]}': {values}")
```