Báo cáo thực hành nhóm 11

Pha 1:

Ta dùng phím F5 để chuyển sang xem mã giả

```
int __cdecl phase1(int a1)
{
  int result; // eax@4
  int v2[10]; // [sp+Ch] [bp-2Ch]@1

  v2[7] = __isoc99_sscanf(a1, "%d %d %d %d %d %d", v2, &v2[1], &v2[2], &v2[3], &v2[4], &v2[5]);
  if ( v2[7] != 6 )
    explode_bomb();
  v2[6] = 5;
  if ( v2[0] || (result = v2[1], v2[1] != v2[6]) )
    explode_bomb();
  for ( v2[8] = 2; v2[8] <= 5; ++v2[8] )
  {
    result = v2[v2[8]];
    if ( result != v2[v2[8] - 2] + v2[v2[8] - 1] )
      explode_bomb();
  }
  return result;
}
```

Ta có một mảng 7 phần tử và có v2[6] = 5, input là 6 phần tử

Với điều kiện if ta thấy nếu v2[0]!= 0 thì chương trình sẽ nổ vậy suy ra v2[0] = 0. v2[1] != v2[6] thì cũng sẽ nổ vậy nên v2[1] = 5.

Ta chạy còng lặp for và tính các kết quả gán vào biến result, result chính là kết quả của phần tử cần tìm ở mỗi vòng lặp

Sau cùng ta có kết quả của 6 input: **0 5 5 10 15 25**

Ta thử nhập và xem kết quả

```
[*] Phase 1
- Hint: Numbers are always magical!
0 5 5 10 15 25
Good job! You've cleared the first phase!
```

Pha 2:

Dùng f5 chuyển sang mã giả

```
1 char __cdecl phase2(int a1)
2 {
3   char *v1; // ST28_4@1
4   int v2; // eax@2
5   char *s1; // [sp+Ch] [bp-1Ch]@1
6   char *s2; // [sp+10h] [bp-18h]@1
7
8   v1 = QUESTIONS[27];
9   s2 = ANSWERS[*(&QA_MAP + 27)];
0   s1 = (char *)transfer(a1);
1   if ( !*s2 || (LOBYTE(v2) = is_equal(s1, s2), !v2) )
2     explode_bomb();
3   return v2;
4 }
```

```
.data:0804B060 QUESTIONS        dd offset aMyVehicleRegis ; DATA XREF: phase2+10↑r
.data:0804B060                                            ; "My vehicle registration plate starts wi"...
.data:0804B064                  dd offset aWhatIsTheCapit ; "What is the capital of Thailand?"
.data:0804B068                  dd offset aWhatIsYourMajo ; "What is your major in English? (Capital"...
.data:0804B06C                  dd offset aWhichSeasonHas ; "Which season has cherry blossoms?"
.data:0804B070                  dd offset aThanksToMeYouC ; "Thanks to me, you can see straight thro"...
.data:0804B074                  dd offset aWhichCountryIs ; "Which country is the Lion city in South"...
.data:0804B078                  dd offset aWhatIsTheMainL ; "What is the main language used in this "...
.data:0804B07C                  dd offset aEnterTheCurren ; "Enter the current date using the format"...
.data:0804B080                  dd offset aWhichProvinceI ; "Which province in Vietnam has the most "...
.data:0804B084                  dd offset aIAmAnOddNumber ; "I am an odd number. Take away one lette"...
.data:0804B088                  dd offset aWhatWordIsSpel ; "What word is spelled incorrectly in eve"...
.data:0804B08C                  dd offset aWhatIsYourNati ; "What is your nationality?"
.data:0804B090                  dd offset aWhatIsThePhone ; "What is the phone number of our univers"...
.data:0804B094                  dd offset aWhatIsTheName0 ; "What is the name of the analyzed execut"...
.data:0804B098                  dd offset aWhichCityHas37 ; "Which city has 3/7 of a chicken and 2/3"...
.data:0804B09C                  dd offset aInUitWhichMajo ; "In UIT, which major tops the list alpha"...
.data:0804B0A0                  dd offset aIsabellaSParen ; "Isabella's parents have four children. "...
.data:0804B0A4                  dd offset aWhatIsTheFullE ; "What is the full English name of our un"...
.data:0804B0A8                  dd offset aWhichFastFoodB ; "Which fast food brand features a bee in"...
.data:0804B0AC                  dd offset aCompleteTheDom ; "Complete the domain of our faculty: htt"...
.data:0804B0B0                  dd offset aWhatIsTheLarge ; "What is the largest country in the worl"...
.data:0804B0B4                  dd offset aWhatIsTheEstab ; "What is the establishment date of UIT ("...
.data:0804B0B8                  dd offset aWhichContinent ; "Which continent has the least populatio"...
.data:0804B0BC                  dd offset aWhatIsTheLonge ; "What is the longest wall in the world? "...
.data:0804B0C0                  dd offset aWhatAreEaxEbxE ; "What are eax, ebx, eip, esi on a comput"...
.data:0804B0C4                  dd offset aWhatIsTheSmall ; "What is the smallest planet in the Sola"...
.data:0804B0C8                  dd offset aWhatIsTheDomai ; "What is the domain of UIT Students' ema"...
.data:0804B0CC                  dd offset aWhatIsTheVietn ; "What is the Vietnamese name (without ac"...
.data:0804B0D0                  dd offset aWhichIsTheLarg ; "Which is the largest net in the world?"
.data:0804B0D4                  align 10h
.data:0804B0E0                  public QA_MAP

.data:0804B160 ANSWERS          dd offset aJqupLcwuo     ; DATA XREF: phase2+2A↑r
.data:0804B160                                           ; "Jqup Lcwuo"
.data:0804B164                  dd offset aJiuoswsw      ; "Jiuoswsw"
.data:0804B168                  dd offset aQunwzuibqwuAmk ; "Qunwzuibqwu Amkczqbg"
.data:0804B16C                  dd offset aAxzquo        ; "Axzquo"
.data:0804B170                  dd offset aEqulwe        ; "Equlwe"
.data:0804B174                  dd offset aAquoixwzm     ; "Aquoixwzm"
.data:0804B178                  dd offset aDqmbuiumam    ; "Dqmbuiumam"
.data:0804B17C                  dd offset a830802        ; "83/0802"
.data:0804B180                  dd offset aJqupLcwuo     ; "Jqup Lcwuo"
.data:0804B184                  dd offset aAmdmu         ; "Amdmu"
.data:0804B188                  dd offset aQukwzzmkbtg   ; "Qukwzzmkbtg"
.data:0804B18C                  dd offset aDqmbuiumam    ; "Dqmbuiumam"
.data:0804B190                  dd offset a80615030880   ; "80615030880"
.data:0804B194                  dd offset aUb087CqbJwuj  ; "ub087-cqb-jwuj"
.data:0804B198                  dd offset aKpqkiow       ; "Kpqkiow"
.data:0804B19C                  dd offset aIzbqnqkqitQubm ; "Izbqnqkqit Qubmttqomukm"
.data:0804B1A0                  dd offset aQaijmtti      ; "Qaijmtti"
.data:0804B1A4                  dd offset aCuqdmzaqbgWnQu ; "Cuqdmzaqbg Wn Qunwzuibqwu Bmkpuwtwog"
.data:0804B1A8                  dd offset aRwttqjmm      ; "Rwttqjmm"
.data:0804B1AC                  dd offset aUk_cqb_mlc_du ; "uk.cqb.mlc.du"
.data:0804B1B0                  dd offset aZcaaqi        ; "Zcaaqi"
.data:0804B1B4                  dd offset a86840884      ; "86/84/0884"
.data:0804B1B8                  dd offset aIubizkbqki    ; "Iubizkbqki"
.data:0804B1BC                  dd offset aBpmOzmibEittWn ; "Bpm Ozmib Eitt Wn Kpqui"
.data:0804B1C0                  dd offset aZmoqabmza     ; "Zmoqabmza"
.data:0804B1C4                  dd offset aUmzkczg       ; "Umzkczg"
.data:0804B1C8                  dd offset aOu_cqb_mlc_du ; "ou.cqb.mlc.du"
.data:0804B1CC                  dd offset aDiuUqmcYcwkBcO ; "Diu Uqmc Ycwk Bc Oqiu"
.data:0804B1D0                  dd offset aQubmzomb      ; "Qubmzomb"
.data:0804B1D4                  public PHASE_MSG
```

```
int __cdecl transfer(int a1)
{
  char v2; // [sp+Ah] [bp-6h]@8
  char v3; // [sp+Bh] [bp-5h]@2
  int i; // [sp+Ch] [bp-4h]@1

  for ( i = 0; *(_BYTE *)(i + a1); ++i )
  {
    v3 = *(_BYTE *)(i + a1);
    if ( (v3 <= 96 || v3 > 122) && (v3 <= 64 || v3 > 90) )
    {
      if ( v3 > 47 && v3 <= 57 )
        v3 = (v3 - 48 + 8) % 10 + 48;
    }
    else
    {
      if ( v3 <= 96 || v3 > 122 )
        v2 = 65;
      else
        v2 = 97;
      v3 = (v3 - v2 + 8) % 26 + v2;
    }
    *(_BYTE *)(a1 + i) = v3;
  }
  return a1;
}
```

Chuyển sang code C để thuận tiện trong việc tìm kết quả (Decrypt)

```c
#include <stdio.h>

void decrypt(char* str) {
    char v2;
    char v3;
    int i;

    for (i = 0; str[i] != '\0'; ++i) {
        v3 = str[i];
        if ((v3 <= 96 || v3 > 122) && (v3 <= 64 || v3 > 90)) {
            if (v3 > 47 && v3 <= 57)
                v3 = (v3 - 48 - 8 + 10) % 10 + 48;  // Dịch ngược cho số
        } else {
            if (v3 <= 96 || v3 > 122)
                v2 = 65;
            else
                v2 = 97;
            v3 = (v3 - v2 - 8 + 26) % 26 + v2;  // Dịch ngược cho chữ cái
        }
        str[i] = v3;
    }
}

int main() {
    char str[] = "Div Uqmc Ycwk Bc Oqiu";
    decrypt(str);
    printf("Decrypted string: %s\n", str);
    return 0;
}
```

Question ở vị trí 27 vậy ta sẽ đến từ 0 đến 27 thì câu hỏi sẽ ở vị trí được khoanh đỏ

Câu trả lời cũng ở vị trí 27 từ 0 ta cũng sẽ được câutrả lời ở vị trí khoanh đỏ
Ta thử mã hóa chuỗi theo code C, ta sẽ được một chuỗi: **Van Mieu Quoc Tu Giam**

```
Decrypted string: Van Mieu Quoc Tu Giam

Process returned 0 (0x0)   execution time : 0.069 s
Press any key to continue.
```

Ta thử kết quả:

```
[*] Phase 2
- Hint: You must answer your secret question!
Van Mieu Quoc Tu Giam
Two phases have been solved. Keep going!
```

Pha 3:

Ta có mã giả

```
int v2; // [sp+0h] [bp-18h]@1
int v3; // [sp+4h] [bp-14h]@1
int v4; // [sp+8h] [bp-10h]@1
int v5; // [sp+Ch] [bp-Ch]@1

v5 = 0;
v4 = 0;
v4 = __isoc99_sscanf(a1, "%d %d", &v3, &v2);
if ( v4 <= 1 )
  explode_bomb();
switch ( v3 )
{
  case 0:
    v5 = 408;
    break;
  case 1:
    v5 = 936;
    break;
  case 2:
    v5 = 208;
    break;
  case 3:
    v5 = 966;
    break;
  case 4:
    v5 = 624;
    break;
  case 5:
    v5 = 373;
    break;
  case 6:
    v5 = 363;
    break;
  case 7:
    v5 = 262;
    break;
  default:
    explode_bomb();
    return result;
}
```

```
}
result = v2;
if ( v5 != v2 )
  explode_bomb();
return result;
```

Ta có input là 2 số nguyên v3, v2. Trong Switch ta thấy v3 có nhiều giá trị, và nếu v5 khác v2 thì sẽ nổ, vậy giá trị của v5 cũng sẽ là giá trị của v2. Vậy với một giá trị v3 sẽ có một giá trị v2.

Vậy đây là pha có nhiều kết quả tìm được

Kết quả tìm được:

Ta thử kết quả:

```
[*] Phase 3
- Hint: Many cases make everything so confusing.
0 408
You've beaten another phase, that's great. What about the fourth one?
```

Pha 4

Ta chuyển sang mã giả

```
int __cdecl phase4(int a1)
{
  int result; // eax@5
  int v2; // [sp+Ch] [bp-1Ch]@1
  int v3; // [sp+10h] [bp-18h]@1
  int v4; // [sp+14h] [bp-14h]@5
  int v5; // [sp+18h] [bp-10h]@5
  int v6; // [sp+1Ch] [bp-Ch]@1

  v6 = __isoc99_sscanf(a1, "%d %d", &v2, &v3);
  if ( v6 != 2 || v3 <= 1 || v3 > 4 )
    explode_bomb();
  v5 = 9;
  v4 = func4(9, v3);
  result = v2;
  if ( v4 != v2 )
    explode_bomb();
  return result;
}
```

Ta có input là 2 số v2, v3. Với điều kiện thì ta suy ra v3 nằm trong khoảng [2, 3, 4];

Ta chuyển sang code C, để thuận tiện trong việc giải quyết pha

```c
#include <bits/stdc++.h>

int __cdecl func4(int a1, int a2)
{
  int result; // eax@2
  int v3; // ebx@5

  if ( a1 > 0 )
  {
    if ( a1 == 1 )
    {
      result = a2;
    }
    else
    {
      v3 = func4(a1 - 1, a2) + a2;
      result = v3 + func4(a1 - 2, a2);
    }
  }
  else
  {
    result = 0;
  }
  return result;
}

int main()
{
  int result; // eax@5
  int v2; // [sp+Ch] [bp-1Ch]@1
  int v3[] = {2, 3, 4}; // [sp+10h] [bp-18h]@1
  int v4; // [sp+14h] [bp-14h]@5
  int v5; // [sp+18h] [bp-10h]@5
  int v6; // [sp+1Ch] [bp-Ch]@1

  for (int i = 0; i < 3; i++)
  {
    printf("Truong hop %d:\n", i+1);
    printf("v3 = %d\n", v3[i]);

    v4 = func4(9, v3[i]);
    v2 = v4;
    printf("v2 = %d\n\n",v2);
  }
}
```

Ta có kết quả như sau:

```
Truong hop 1:
v3 = 2
v2 = 176

Truong hop 2:
v3 = 3
v2 = 264

Truong hop 3:
v3 = 4
v2 = 352


Process returned 0 (0x0)   execution time : 0.033 s
Press any key to continue.
```

Vậy pha 4 có nhiều kết quả, ta thử kiểm tra 1 kết quả

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

Pha 5

Ta có mã giả

```
size_t __cdecl phase5(char *s)
{
  size_t result; // eax@1
  int v2; // [sp+8h] [bp-10h]@3
  signed int i; // [sp+Ch] [bp-Ch]@3

  result = strlen(s);
  if ( result != 6 )
    explode_bomb();
  v2 = 0;
  for ( i = 0; i <= 5; ++i )
  {
    result = array_3852[s[i] & 0xF];
    v2 += result;
  }
  if ( v2 != 51 )
    explode_bomb();
  return result;
}
```

Ta có input là 1 chuỗi gồm 6 kí tự, với mỗi kí tự là vị trí của các phần tử trong mảng array_3852

Mỗi vòng lặp ta sẽ có một result chính là một giá trị phần tử trong mảng array_3852 , và cứ cộng vào thì kết quả của tổng 6 phần tử đó sẽ là 51. s[i] and với 0xF thì sẽ chính là s[i]. Và s[i] chính là kí tự ở vị ví thứ i trong chuỗi s.

Ta có mảng array_3852 theo mã giả:

```
.data:0804B200 ; int array_3852[]
.data:0804B200 array_3852      dd 2                    ;
.data:0804B204                 db  0Ah
.data:0804B205                 db    0
.data:0804B206                 db    0
.data:0804B207                 db    0
.data:0804B208                 db    6
.data:0804B209                 db    0
.data:0804B20A                 db    0
.data:0804B20B                 db    0
.data:0804B20C                 db    1
.data:0804B20D                 db    0
.data:0804B20E                 db    0
.data:0804B20F                 db    0
.data:0804B210                 db  0Ch
.data:0804B211                 db    0
.data:0804B212                 db    0
.data:0804B213                 db    0
.data:0804B214                 db  10h
.data:0804B215                 db    0
.data:0804B216                 db    0
.data:0804B217                 db    0
.data:0804B218                 db    9
.data:0804B219                 db    0
.data:0804B21A                 db    0
.data:0804B21B                 db    0
.data:0804B21C                 db    3
.data:0804B21D                 db    0
.data:0804B21E                 db    0
.data:0804B21F                 db    0
.data:0804B220                 db    4
.data:0804B221                 db    0
.data:0804B222                 db    0
.data:0804B223                 db    0
.data:0804B224                 db    7
.data:0804B225                 db    0
.data:0804B226                 db    0
.data:0804B227                 db    0
.data:0804B228                 db  0Eh
data:0804B228                  db  0Eh
data:0804B229                  db    0
data:0804B22A                  db    0
data:0804B22B                  db    0
data:0804B22C                  db    5
data:0804B22D                  db    0
data:0804B22E                  db    0
data:0804B22F                  db    0
data:0804B230                  db  0Bh
data:0804B231                  db    0
data:0804B232                  db    0
data:0804B233                  db    0
data:0804B234                  db    8
data:0804B235                  db    0
data:0804B236                  db    0
data:0804B237                  db    0
data:0804B238                  db  0Fh
data:0804B239                  db    0
data:0804B23A                  db    0
data:0804B23B                  db    0
data:0804B23C                  db  0Dh
data:0804B23D                  db    0
data:0804B23E                  db    0
data:0804B23F                  db    0
data:0804B23F _data            ends
```

Từ đó ta có thể có một mảng như sau

Int Array_3852[] = {2, 10, 6, 1, 12, 16, 9, 3, 4, 7, 14, 5, 11,15, 13 }

Vì index của mảng chính là mỗi kí tự của s, cho nên s[i] thuộc [0;9]

Vậy Array_3852[] = {2, 10, 6, 1, 12, 16, 9, 3, 4, 7}

Ta chuyển sang code C

```cpp
#include <iostream>
#include <vector>

using namespace std;

// Hàm đệ quy để tìm các tổ hợp có tổng bằng 51
void findCombinations(vector<int>& array, vector<int>& combination, vector<int>& locate, int sum, int index, int& count) {
    // Nếu tổ hợp có 6 số và tổng bằng 51, in ra tổ hợp
    if (combination.size() == 6) {
        if (sum == 51) {
            for (int num : combination) {
                cout << num << " ";
            }
            cout << " ---> ";
            for (int i : locate)
                cout << i;
            cout << endl;
            count++;
        }
        return;
    }

    // Duyệt các số từ index đến cuối mảng
    for (int i = index; i < array.size(); ++i) {
        if (sum + array[i] <= 51) { // Chỉ thêm số vào tổ hợp nếu tổng không vượt quá 51
            combination.push_back(array[i]);
            locate.push_back(i);
            findCombinations(array, combination, locate, sum + array[i], i + 1, count);
            combination.pop_back();
            locate.pop_back();
        }
    }
}

int main() {
    vector<int> array = {2, 10, 6, 1, 12, 16, 9, 3, 4, 7};
    vector<int> locate;
    vector<int> combination;
    int count = 0;

    findCombinations(array, combination, locate, 0, 0, count);

    cout << "Total combinations: " << count << endl;

    return 0;
}
```

Kết quả chạy code:

```
2 10 12 16 4 7  ---> 014589
10 6 12 16 3 4  ---> 124578
10 6 16 9 3 7  ---> 125679
10 1 12 16 9 3  ---> 134567
6 1 12 16 9 7  ---> 234569
12 16 9 3 4 7  ---> 456789
Total combinations: 6

Process returned 0 (0x0)   execution time : 0.106 s
Press any key to continue.
```

Với trường hợp 2 10 12 16 4 7 ta có tương ứng với các index là từng kí tự của s như sau:

2 -> 0

10 ->1

12 ->4

16 -> 5

4 -> 8

7 -> 9

Vậy với trường hợp 1 ta có chuỗi s = 014589.

Tương tự với các trường hợp còn lại

Ta chạy thử kết quả:

```
[*] Phase 5
-Hint: No hint is also a hint :)
014589
Amazing bomb solvers, the bomb has been deactived. Enjoy your day :))
```