

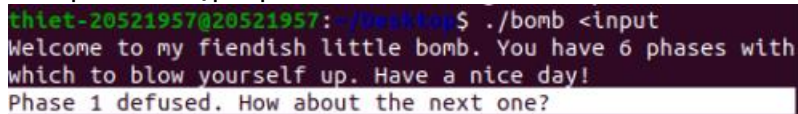
Câu 1:

- Định dạng file input là 1 chuỗi ký tự
- Điều kiện: so sánh chuỗi input với chuỗi đã được set sẵn. Nếu bằng nhau thì đúng không boom.
- Input: There are rumors on the internets.

```
int __cdecl phase_1(int a1)
{
    int result; // eax@1

    result = strings_not_equal(a1, "There are rumors on the internets.");
    if ( result )
        explode_bomb();
    return result;
}
```

- Kết quả khi nhập input:



```
thiet-20521957@20521957:~/Desktop$ ./bomb <input
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
Phase 1 defused. How about the next one?
```

Câu 2:

- Định dạng file input : là 1 chuỗi 6 ký tự số liên tiếp, được lưu trong mảng v3

```
int __cdecl read_six_numbers(int a1, int a2)
{
    int result; // eax@1

    result = __isoc99_sscanf(a1, "%d %d %d %d %d %d", a2, a2 + 4, a2 + 8, a2 + 12, a2 + 16, a2 + 20);
    if ( result <= 5 )
        explode_bomb();
    return result;
}
```

- Điều kiện: ký tự đầu tiên của mảng phải là 1 và sẽ dùng vòng lặp chạy từ 1 cho đến <=5 để kiểm tra ký tự tiếp theo là = số trước *2

```
int __cdecl phase_2(int a1)
{
    signed int i; // [sp+10h] [bp-28h]@3
    int v3[6]; // [sp+14h] [bp-24h]@1
    int v4; // [sp+2Ch] [bp-Ch]@1

    v4 = *MK_FP(__GS__, 20);
    read_six_numbers(a1, (int)v3);
    if ( v3[0] != 1 )
        explode_bomb();
    for ( i = 1; i <= 5; ++i )
    {
        if ( v3[i] != 2 * v3[i - 1] )
            explode_bomb();
    }
    return *MK_FP(__GS__, 20) ^ v4;
}
```

- Input: 1 2 4 8 16 32
- Kết quả khi nhập file input

```
Welcome to my fiendish little bomb. You have 6 phases with  
which to blow yourself up. Have a nice day!  
Phase 1 defused. How about the next one?  
That's number 2. Keep going!
```

Câu 3:

- Định dạng input: 2 gồm 2 số kiểu int

```
int __cdecl phase_3(int a1)  
{  
    int result; // eax@15  
    int v2; // [sp+1Ch] [bp-1Ch]@1  
    int v3; // [sp+20h] [bp-18h]@1  
    int v4; // [sp+24h] [bp-14h]@1  
    int v5; // [sp+28h] [bp-10h]@1  
    int v6; // [sp+2Ch] [bp-Ch]@1  
  
    v6 = *MK_FP(__GS__, 20);  
    v4 = 0;  
    v5 = 0;  
    v5 = __isoc99_sscanf(a1, "%d %d", &v2, &v3);
```

- Điều kiện: chương trình sẽ kiểm tra biến đầu tiên ta nhập vào. Biến đầu tiên sẽ là 1 trong các 8 case phía dưới. Sau đó chương trình sẽ kiểm tra biến thứ 2 ta nhập vào theo từng case ở biến thứ 1 có trùng với biến ở trong case không. Nếu đúng thì boom không nổ

```
if ( v5 <= 1 )  
    explode_bomb();  
switch ( v2 )  
{  
    case 0:  
        v4 = 898;  
        break;  
    case 1:  
        v4 = 498;  
        break;  
    case 2:  
        v4 = 292;  
        break;  
    case 3:  
        v4 = 994;  
        break;  
    case 4:  
        v4 = 773;  
        break;  
    case 5:  
        v4 = 813;  
        break;  
    case 6:  
        v4 = 781;  
        break;  
    case 7:
```

- Input:
TH1: 0 898
TH2: 1 498
TH3: 2 292

TH4: 3 994
TH5: 4 773
TH6: 5 813
TH7: 6 781
TH8: 7 866

- Kết quả khi nhập 1 trong các TH ở trên:

```
Welcome to my fiendish little bomb. You have 6 phases with  
which to blow yourself up. Have a nice day!  
Phase 1 defused. How about the next one?  
That's number 2. Keep going!  
Halfway there!
```

Câu 4:

- Định dạng Input: là 2 số nguyên kiểu int

```
int __cdecl phase_4(int a1)  
{  
    int v2; // [sp+18h] [bp-20h]@1  
    int v3; // [sp+1Ch] [bp-1Ch]@1  
    int v4; // [sp+20h] [bp-18h]@1  
    int v5; // [sp+24h] [bp-14h]@5  
    int v6; // [sp+28h] [bp-10h]@5  
    int v7; // [sp+2Ch] [bp-Ch]@1  
  
    v7 = *MK_FP(__GS__, 20);  
    v4 = __isoc99_sscanf(a1, "%d %d", &v3, &v2);
```

- Điều kiện: Nếu như input không đủ 2 ký tự số hoặc ký tự thứ 2 nhập vào phải nằm trong khoảng từ 2 cho đến 4 thì bom mới không nổ. Và ký tự v3 đầu tiên phải được tính từ hàm func4 với 2 tham số đầu vào là 8 và ký tự thứ 2 mình nhập

```
    v4 = __isoc99_sscanf(a1, "%d %d", &v3, &v2);  
    if ( v4 != 2 || v2 <= 1 || v2 > 4 )  
        explode_bomb();  
  
    v5 = 8;  
    v6 = func4(8, v2);  
    if ( v6 != v3 )  
        explode_bomb();
```

- Hàm func4

```

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;
}

```

- Input: Sau khi tính từ hàm func4 với a2 từ 2 đến 4
TH1: 108 2
TH2: 162 3
TH3: 216 4

- Kết quả khi nhập 1 trong các input trên:

```

Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
Phase 1 defused. How about the next one?
That's number 2. Keep going!
Halfway there!
So you got that one. Try this one.

```

Câu 5:

- Input: không có kiểu dữ liệu quy định, chỉ cần nhập đủ 6 ký tự thì bom sẽ không nổ

```

result = string_length(a1);
if ( result != 6 )
    explode_bomb();

```

- Điều kiện: Ta thấy trong vòng lặp result sẽ được gán bằng index của array_2703 sau khi tính $(_BYTE *) (i + a1) \& 0xF$ sẽ tính ký tự của ta nhập sau đó $\& 0xF$ và ra số ở dạng hex

```

for ( i = 0; i <= 5; ++i )
{
    result = array_2703[(\_BYTE *) (i + a1) & 0xF];
    v3 += result;
}
if ( v3 != 53 )

```

- Các giá trị có trong mảng array_2703

```

array_2703    dd 2, 0Ah, 6, 1, 0Ch, 10h, 9, 3, 4, 7, 0Eh, 5, 0Bh, 8
; DATA XREF: phase_5+43↑r
              dd 0Fh, 0Dh

```

- Sau đó sẽ dùng index để tìm phần tử trong mảng array_2703 rồi cộng với v3
- Giá trị cộng lại phải sao cho $v3 == 53$ thì bom mới không nổ

- Ta thấy trong mảng array_2703 các giá trị từ 1 đến 16 -> mảng có 15 phần tử
- Ta viết 1 chương trình để tính index của 6 số trong mảng array_2703 sao cho giá trị tại index đó tổng 6 số lại = 53, có tổng cộng 153 TH

```

1 // C++ program for naive solution to
2 // print all combination of 4 elements
3 // in A[] with sum equal to X
4 #include <bits/stdc++.h>
5 using namespace std;
6 void findSixElements(int A[], int n, int X){
7     int count = 0;
8     for(int a=0; a<n-5; a++){
9         for(int b=a+1; b<n-4; b++){
10             for(int c=b+1; c<n-3; c++){
11                 for(int d=c+1; d<n-2; d++){
12                     for(int e=d+1; e<n-1; e++){
13                         for(int f=e+1; f<n; f++){
14                             if(A[a]+A[b]+A[c]+A[d]+A[e]+A[f]==X){
15                                 cout<<"Index:a="<<a<<","<<"Index:b="<<b<<","<<"Index:c="<<c<<","<<"Index:d="<<d<<","<<"Index:e="<<e<<","<<"Index:f="<<f<<endl;
16                             }
17                         }
18                     }
19                 }
20             }
21         }
22     }
23 }
24 // Driver Code
25 int main()
26 {
27     int A[] = {0x2, 0x0A, 0x6, 0x1, 0x0C, 0x10, 0x9, 0x3, 0x4, 0x7, 0x0E, 0x5,
28               0x0B, 0x8, 0x0F, 0x0D};
29     int n = sizeof(A) / sizeof(A[0]);
30     int X = 53;
31     findSixElements(A, n, X);
32     return 0;
33 }

```

- Em vẫn chưa viết được cách đổi từ chỉ số index đó sang kiểu char giới hạn từ 33 đến 127 trong bảng mã ASCII
- Ta sẽ dùng Index đó tra trong bảng mã ASCII thì sẽ tìm được pass. Tại vì ta phải nhập vào sao cho ký tự ta nhập &0xF sẽ ra được index của số trong mảng
- Input:
- Array_2703 = {0x2, 0x0A, 0x6, 0x1, 0x0C, 0x10, 0x9, 0x3, 0x4, 0x7, 0x0E, 0x5, 0x0B, 0x8, 0x0F, 0x0D}
- TH1:

Index:a=0,Index:b=1,Index:c=2,Index:d=4,Index:e=10,Index:f=6

Ta tra trong bảng ASCII thì a1 & 0xF = 0 thì ở dạng 0xa0(với a là số bất kỳ) và ký tự ta nhập phải trong khoảng từ 33 đến 127(decimal)

->Ký tự thứ 1(a): @ hoặc `

Tương tự với các Index phía sau

->Ký tự thứ 2(b): ! hoặc A hoặc a (0xa1)

->Ký tự thứ 3(c): " hoặc B hoặc b (0xa2)

->Ký tự thứ 4(d): \$ hoặc D hoặc d (0xa4)

->Ký tự thứ 5(e): * hoặc J hoặc j (0xaA)

->Ký tự thứ 6(f): & hoặc F hoặc f (0xa6)

->Input: @!"\$*&

- Kết quả sau khi nhập

```

Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
Phase 1 defused. How about the next one?
That's number 2. Keep going!
Halfway there!
So you got that one. Try this one.
Good work! On to the next...

```

- Các trường hợp Index còn lại tương tự như trên
 - Index:a=0,Index:b=1,Index:c=2,Index:d=4,Index:e=10,Index:f=6
 - Index:a=0,Index:b=1,Index:c=3,Index:d=10,Index:e=14,Index:f=12

- Index:a=0,Index:b=1,Index:c=4,Index:d=8,Index:e=12,Index:f=10
- Index:a=0,Index:b=1,Index:c=5,Index:d=7,Index:e=14,Index:f=9
- Index:a=0,Index:b=1,Index:c=5,Index:d=8,Index:e=9,Index:f=10
- Index:a=0,Index:b=1,Index:c=5,Index:d=9,Index:e=15,Index:f=11
- Index:a=0,Index:b=1,Index:c=6,Index:d=8,Index:e=10,Index:f=10
- Index:a=0,Index:b=1,Index:c=7,Index:d=10,Index:e=15,Index:f=12
- Index:a=0,Index:b=1,Index:c=9,Index:d=12,Index:e=13,Index:f=14
- Index:a=0,Index:b=1,Index:c=9,Index:d=13,Index:e=15,Index:f=15
- Index:a=0,Index:b=1,Index:c=11,Index:d=13,Index:e=14,Index:f=15
- Index:a=0,Index:b=2,Index:c=4,Index:d=5,Index:e=10,Index:f=7
- Index:a=0,Index:b=2,Index:c=4,Index:d=8,Index:e=14,Index:f=10
- Index:a=0,Index:b=2,Index:c=4,Index:d=10,Index:e=13,Index:f=12
- Index:a=0,Index:b=2,Index:c=5,Index:d=8,Index:e=12,Index:f=10
- Index:a=0,Index:b=2,Index:c=6,Index:d=10,Index:e=12,Index:f=12
- Index:a=0,Index:b=2,Index:c=6,Index:d=13,Index:e=14,Index:f=15
- Index:a=0,Index:b=2,Index:c=8,Index:d=12,Index:e=14,Index:f=14
- Index:a=0,Index:b=2,Index:c=9,Index:d=10,Index:e=15,Index:f=12
- Index:a=0,Index:b=2,Index:c=12,Index:d=13,Index:e=15,Index:f=15
- Index:a=0,Index:b=3,Index:c=4,Index:d=10,Index:e=15,Index:f=12
- Index:a=0,Index:b=3,Index:c=5,Index:d=12,Index:e=13,Index:f=14
- Index:a=0,Index:b=3,Index:c=5,Index:d=13,Index:e=15,Index:f=15
- Index:a=0,Index:b=3,Index:c=6,Index:d=12,Index:e=14,Index:f=14
- Index:a=0,Index:b=3,Index:c=10,Index:d=13,Index:e=14,Index:f=15
- Index:a=0,Index:b=4,Index:c=5,Index:d=7,Index:e=15,Index:f=9
- Index:a=0,Index:b=4,Index:c=5,Index:d=8,Index:e=11,Index:f=10
- Index:a=0,Index:b=4,Index:c=5,Index:d=9,Index:e=12,Index:f=11
- Index:a=0,Index:b=4,Index:c=6,Index:d=10,Index:e=11,Index:f=12
- Index:a=0,Index:b=4,Index:c=7,Index:d=10,Index:e=12,Index:f=12
- Index:a=0,Index:b=4,Index:c=7,Index:d=13,Index:e=14,Index:f=15
- Index:a=0,Index:b=4,Index:c=11,Index:d=12,Index:e=13,Index:f=14
- Index:a=0,Index:b=4,Index:c=11,Index:d=13,Index:e=15,Index:f=15
- Index:a=0,Index:b=5,Index:c=6,Index:d=8,Index:e=13,Index:f=10
- Index:a=0,Index:b=5,Index:c=6,Index:d=9,Index:e=10,Index:f=11
- Index:a=0,Index:b=5,Index:c=6,Index:d=11,Index:e=15,Index:f=13
- Index:a=0,Index:b=5,Index:c=7,Index:d=8,Index:e=10,Index:f=10
- Index:a=0,Index:b=5,Index:c=9,Index:d=11,Index:e=14,Index:f=13
- Index:a=0,Index:b=5,Index:c=10,Index:d=11,Index:e=13,Index:f=13
- Index:a=0,Index:b=6,Index:c=7,Index:d=12,Index:e=15,Index:f=14
- Index:a=0,Index:b=6,Index:c=8,Index:d=10,Index:e=15,Index:f=12
- Index:a=0,Index:b=6,Index:c=10,Index:d=11,Index:e=14,Index:f=13
- Index:a=0,Index:b=7,Index:c=9,Index:d=12,Index:e=14,Index:f=14
- Index:a=0,Index:b=7,Index:c=10,Index:d=12,Index:e=13,Index:f=14
- Index:a=0,Index:b=7,Index:c=10,Index:d=13,Index:e=15,Index:f=15
- Index:a=0,Index:b=8,Index:c=9,Index:d=10,Index:e=14,Index:f=12
- Index:a=0,Index:b=8,Index:c=12,Index:d=13,Index:e=14,Index:f=15
- Index:a=0,Index:b=9,Index:c=11,Index:d=12,Index:e=15,Index:f=14
- Index:a=1,Index:b=2,Index:c=3,Index:d=4,Index:e=14,Index:f=6
- Index:a=1,Index:b=2,Index:c=3,Index:d=10,Index:e=12,Index:f=12
- Index:a=1,Index:b=2,Index:c=3,Index:d=13,Index:e=14,Index:f=15
- Index:a=1,Index:b=2,Index:c=4,Index:d=7,Index:e=14,Index:f=9
- Index:a=1,Index:b=2,Index:c=4,Index:d=8,Index:e=9,Index:f=10

- Index:a=1,Index:b=2,Index:c=4,Index:d=9,Index:e=15,Index:f=11
- Index:a=1,Index:b=2,Index:c=5,Index:d=6,Index:e=13,Index:f=8
- Index:a=1,Index:b=2,Index:c=5,Index:d=7,Index:e=12,Index:f=9
- Index:a=1,Index:b=2,Index:c=5,Index:d=11,Index:e=13,Index:f=13
- Index:a=1,Index:b=2,Index:c=6,Index:d=11,Index:e=14,Index:f=13
- Index:a=1,Index:b=2,Index:c=7,Index:d=12,Index:e=13,Index:f=14
- Index:a=1,Index:b=2,Index:c=7,Index:d=13,Index:e=15,Index:f=15
- Index:a=1,Index:b=2,Index:c=8,Index:d=10,Index:e=13,Index:f=12
- Index:a=1,Index:b=2,Index:c=9,Index:d=10,Index:e=11,Index:f=12
- Index:a=1,Index:b=3,Index:c=4,Index:d=5,Index:e=12,Index:f=7
- Index:a=1,Index:b=3,Index:c=4,Index:d=10,Index:e=11,Index:f=12
- Index:a=1,Index:b=3,Index:c=5,Index:d=6,Index:e=15,Index:f=8
- Index:a=1,Index:b=3,Index:c=5,Index:d=8,Index:e=13,Index:f=10
- Index:a=1,Index:b=3,Index:c=5,Index:d=9,Index:e=10,Index:f=11
- Index:a=1,Index:b=3,Index:c=5,Index:d=11,Index:e=15,Index:f=13
- Index:a=1,Index:b=3,Index:c=6,Index:d=8,Index:e=14,Index:f=10
- Index:a=1,Index:b=3,Index:c=6,Index:d=10,Index:e=13,Index:f=12
- Index:a=1,Index:b=3,Index:c=7,Index:d=12,Index:e=15,Index:f=14
- Index:a=1,Index:b=3,Index:c=8,Index:d=10,Index:e=15,Index:f=12
- Index:a=1,Index:b=3,Index:c=10,Index:d=11,Index:e=14,Index:f=13
- Index:a=1,Index:b=4,Index:c=5,Index:d=7,Index:e=11,Index:f=9
- Index:a=1,Index:b=4,Index:c=7,Index:d=11,Index:e=14,Index:f=13
- Index:a=1,Index:b=4,Index:c=8,Index:d=9,Index:e=14,Index:f=11
- Index:a=1,Index:b=4,Index:c=9,Index:d=11,Index:e=12,Index:f=13
- Index:a=1,Index:b=5,Index:c=6,Index:d=7,Index:e=13,Index:f=9
- Index:a=1,Index:b=5,Index:c=7,Index:d=11,Index:e=12,Index:f=13
- Index:a=1,Index:b=5,Index:c=8,Index:d=9,Index:e=12,Index:f=11
- Index:a=1,Index:b=6,Index:c=7,Index:d=8,Index:e=15,Index:f=10
- Index:a=1,Index:b=6,Index:c=8,Index:d=10,Index:e=11,Index:f=12
- Index:a=1,Index:b=7,Index:c=8,Index:d=10,Index:e=12,Index:f=12
- Index:a=1,Index:b=7,Index:c=8,Index:d=13,Index:e=14,Index:f=15
- Index:a=1,Index:b=7,Index:c=9,Index:d=10,Index:e=13,Index:f=12
- Index:a=1,Index:b=7,Index:c=10,Index:d=11,Index:e=15,Index:f=13
- Index:a=1,Index:b=8,Index:c=11,Index:d=12,Index:e=13,Index:f=14
- Index:a=1,Index:b=8,Index:c=11,Index:d=13,Index:e=15,Index:f=15
- Index:a=2,Index:b=3,Index:c=4,Index:d=5,Index:e=14,Index:f=7
- Index:a=2,Index:b=3,Index:c=4,Index:d=12,Index:e=13,Index:f=14
- Index:a=2,Index:b=3,Index:c=4,Index:d=13,Index:e=15,Index:f=15
- Index:a=2,Index:b=3,Index:c=5,Index:d=10,Index:e=11,Index:f=12
- Index:a=2,Index:b=3,Index:c=9,Index:d=12,Index:e=15,Index:f=14
- Index:a=2,Index:b=3,Index:c=11,Index:d=12,Index:e=14,Index:f=14
- Index:a=2,Index:b=4,Index:c=6,Index:d=8,Index:e=13,Index:f=10
- Index:a=2,Index:b=4,Index:c=6,Index:d=9,Index:e=10,Index:f=11
- Index:a=2,Index:b=4,Index:c=6,Index:d=11,Index:e=15,Index:f=13
- Index:a=2,Index:b=4,Index:c=7,Index:d=8,Index:e=10,Index:f=10
- Index:a=2,Index:b=4,Index:c=9,Index:d=11,Index:e=14,Index:f=13
- Index:a=2,Index:b=4,Index:c=10,Index:d=11,Index:e=13,Index:f=13
- Index:a=2,Index:b=5,Index:c=7,Index:d=11,Index:e=14,Index:f=13
- Index:a=2,Index:b=5,Index:c=8,Index:d=9,Index:e=14,Index:f=11
- Index:a=2,Index:b=5,Index:c=9,Index:d=11,Index:e=12,Index:f=13
- Index:a=2,Index:b=6,Index:c=8,Index:d=12,Index:e=13,Index:f=14

- Index:a=2,Index:b=6,Index:c=8,Index:d=13,Index:e=15,Index:f=15
- Index:a=2,Index:b=6,Index:c=10,Index:d=11,Index:e=12,Index:f=13
- Index:a=2,Index:b=7,Index:c=8,Index:d=10,Index:e=14,Index:f=12
- Index:a=2,Index:b=7,Index:c=11,Index:d=12,Index:e=15,Index:f=14
- Index:a=2,Index:b=8,Index:c=9,Index:d=10,Index:e=12,Index:f=12
- Index:a=2,Index:b=8,Index:c=9,Index:d=13,Index:e=14,Index:f=15
- Index:a=2,Index:b=9,Index:c=10,Index:d=11,Index:e=15,Index:f=13
- Index:a=3,Index:b=4,Index:c=5,Index:d=6,Index:e=12,Index:f=8
- Index:a=3,Index:b=4,Index:c=5,Index:d=7,Index:e=10,Index:f=9
- Index:a=3,Index:b=4,Index:c=5,Index:d=11,Index:e=12,Index:f=13
- Index:a=3,Index:b=4,Index:c=6,Index:d=8,Index:e=15,Index:f=10
- Index:a=3,Index:b=4,Index:c=8,Index:d=10,Index:e=12,Index:f=12
- Index:a=3,Index:b=4,Index:c=8,Index:d=13,Index:e=14,Index:f=15
- Index:a=3,Index:b=4,Index:c=9,Index:d=10,Index:e=13,Index:f=12
- Index:a=3,Index:b=4,Index:c=10,Index:d=11,Index:e=15,Index:f=13
- Index:a=3,Index:b=5,Index:c=6,Index:d=9,Index:e=14,Index:f=11
- Index:a=3,Index:b=5,Index:c=7,Index:d=8,Index:e=14,Index:f=10
- Index:a=3,Index:b=5,Index:c=7,Index:d=10,Index:e=13,Index:f=12
- Index:a=3,Index:b=6,Index:c=7,Index:d=10,Index:e=14,Index:f=12
- Index:a=3,Index:b=6,Index:c=8,Index:d=12,Index:e=15,Index:f=14
- Index:a=3,Index:b=6,Index:c=9,Index:d=10,Index:e=12,Index:f=12
- Index:a=3,Index:b=6,Index:c=9,Index:d=13,Index:e=14,Index:f=15
- Index:a=3,Index:b=8,Index:c=9,Index:d=12,Index:e=14,Index:f=14
- Index:a=3,Index:b=8,Index:c=10,Index:d=12,Index:e=13,Index:f=14
- Index:a=3,Index:b=8,Index:c=10,Index:d=13,Index:e=15,Index:f=15
- Index:a=3,Index:b=9,Index:c=12,Index:d=13,Index:e=15,Index:f=15
- Index:a=3,Index:b=11,Index:c=12,Index:d=13,Index:e=14,Index:f=15
- Index:a=4,Index:b=5,Index:c=8,Index:d=11,Index:e=13,Index:f=13
- Index:a=4,Index:b=6,Index:c=7,Index:d=8,Index:e=12,Index:f=10
- Index:a=4,Index:b=6,Index:c=8,Index:d=11,Index:e=14,Index:f=13
- Index:a=4,Index:b=7,Index:c=8,Index:d=12,Index:e=13,Index:f=14
- Index:a=4,Index:b=7,Index:c=8,Index:d=13,Index:e=15,Index:f=15
- Index:a=4,Index:b=7,Index:c=10,Index:d=11,Index:e=12,Index:f=13
- Index:a=4,Index:b=8,Index:c=9,Index:d=10,Index:e=11,Index:f=12
- Index:a=5,Index:b=6,Index:c=7,Index:d=8,Index:e=9,Index:f=10
- Index:a=5,Index:b=6,Index:c=7,Index:d=9,Index:e=15,Index:f=11
- Index:a=5,Index:b=6,Index:c=8,Index:d=11,Index:e=12,Index:f=13
- Index:a=5,Index:b=6,Index:c=9,Index:d=11,Index:e=13,Index:f=13
- Index:a=5,Index:b=7,Index:c=8,Index:d=10,Index:e=11,Index:f=12
- Index:a=5,Index:b=8,Index:c=9,Index:d=11,Index:e=15,Index:f=13
- Index:a=6,Index:b=7,Index:c=9,Index:d=12,Index:e=13,Index:f=14
- Index:a=6,Index:b=7,Index:c=9,Index:d=13,Index:e=15,Index:f=15
- Index:a=6,Index:b=7,Index:c=11,Index:d=13,Index:e=14,Index:f=15
- Index:a=6,Index:b=8,Index:c=9,Index:d=10,Index:e=13,Index:f=12
- Index:a=6,Index:b=8,Index:c=10,Index:d=11,Index:e=15,Index:f=13
- Index:a=7,Index:b=8,Index:c=9,Index:d=12,Index:e=15,Index:f=14
- Index:a=7,Index:b=8,Index:c=11,Index:d=12,Index:e=14,Index:f=14
- Index:a=7,Index:b=11,Index:c=12,Index:d=13,Index:e=15,Index:f=15
- Index:a=8,Index:b=9,Index:c=10,Index:d=11,Index:e=14,Index:f=13

*File input cho 5 phase:

