



HO CHI MINH UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY
SOFTWARE ENGINEERING DEPARTMENT
ADVANCED PROGRAM IN COMPUTER SCIENCE
COURSE: **INTRODUCTION TO CS II**
LECTURER: Dr. ĐÌNH BÁ TIẾN

WEEK 02

POINTER

✚ TRƯƠNG PHƯỚC LỘC
✚ HỒ TUẤN THANH

HCMC, January 31, 2015

Table of contents

1	Problems	3
1.1	Exercise 01	3
1.1.1	Source code 1.1	3
1.1.2	Source code 1.2	4
1.1.3	Source code 1.3	4
1.1.4	Source code 1.4	4
1.1.5	Source code 1.5	4
1.1.6	Source code 1.6	5
1.2	Exercise 02	6
1.3	Exercise 03	6
1.4	Exercise 04	7
1.5	Exercise 05: Dynamic allocated array of integer numbers	7
1.6	Exercise 06	8
1.6.1	Technical requirements	8
1.6.2	Problem	8
2	A03	8
3	H03	9

1 Problems

1.1 Exercise 01¹

What is printed out?

1.1.1 Source code 1.1

```
int x,z;
float y;
char ch, *chp;
int *ip1, *ip2;
float *fp;

x = 100;
y = 20.0;
z = 50;

ip1 = &x;
ip2 = &z;
fp = &y;
chp = &ch;

ip2 = ip1;

ip1 = &z;

*ip1 = *ip2;

*ip1 = 200;

*ip1 = *ip2 + 300;

*fp = 1.2;

printf("%f\n", *fp);

printf("%d\n", ip1);

printf("%d\n", *ip1);
```

¹ <http://www-ee.eng.hawaii.edu/~tep/EE160/Notes/Pointers/exercise1.html>

1.1.2 Source code 1.2

```
void main()
{
    char* a;
    char b;
    long c;
    cout << sizeof(a) <<endl;
    cout << sizeof(b) <<endl;
    cout << sizeof(c) <<endl;
}
```

1.1.3 Source code 1.3

```
int abc;
int *ptr_abc;
abc = 123456;
ptr_abc = &abc;

cout << abc <<endl;
cout << *ptr_abc <<endl;
cout << *abc <<endl;
cout << &abc <<endl;
```

1.1.4 Source code 1.4

```
void main()
{
    int *a = new int;
    int *b = new int;
    *a = 2;
    b = a;
    delete a;
    delete b;
}
```

1.1.5 Source code 1.5

```
int i = 0;
while (true)
{
    i++;
    int *number = new int;
```

```
*number = i;  
cout << *number << endl;  
}
```

1.1.6 Source code 1.6

```
void main()  
{  
    int a[51]={4,6,7,9,12};  
    int i, j, check=0;  
    int N=5;  
  
    for(i=0; i<N; i++)  
    {  
        if(*(a+i)==N)  
        {  
            check = 1;  
            break;  
        }  
    }  
  
    if(check==1)  
    {  
        cout << "... " << endl;  
    }  
    else  
    {  
        cout << "... " << endl;  
  
        for(i=0; i<N; i++)  
        {  
            if(*(a+i)>N)  
                break;  
        }  
  
        for(j=N; j>i; j--)  
        {  
            *(a+j) = *(a+j-1);  
        }  
    }  
}
```

```
        a[i] = N;

        cout << "... " << endl;
        for(i=0; i<=N; i++)
            cout << " " << *(a+i);

    }
}
```

1.2 Exercise 02²

Introduce **int** variables **x** and **y** and **int*** **pointer** variables **p** and **q**. Set **x** to 2, **y** to 8, **p** to the address of **x**, and **q** to the address of **y**. Then print the following information:

- (1) The address of **x** and the value of **x**.
- (2) The value of **p** and the value of ***p**.
- (3) The address of **y** and the value of **y**.
- (4) The value of **q** and the value of ***q**.
- (5) The address of **p** (not its contents!).
- (6) The address of **q** (not its contents!).

Use the **Hex** function to print all **pointer**/address values and format the output so it is easy to make comparisons.

1.3 Exercise 03

Introduce **int** variables **x**, **y**, **z** and **int*** **pointer** variables **p**, **q**, **r**. Set **x**, **y**, **z** to three distinct values. Set **p**, **q**, **r** to the addresses of **x**, **y**, **z** respectively.

- (1) Print with labels the values of **x**, **y**, **z**, **p**, **q**, **r**, ***p**, ***q**, ***r**.

2

<http://74.125.153.132/search?q=cache:NhzTiCTNLbYJ:www.ccs.neu.edu/teaching/EdGroup/VisualCpp/Labs2/Pointers/PointerEx1.doc+pointer+exercise&cd=1&hl=vi&ct=clnk&gl=vn>

- (2) Print the message: Swapping values.
- (3) Execute the swap code: `z = x; x = y; y = z;`
- (4) Print with labels the values of `x, y, z, p, q, r, *p, *q, *r`.

1.4 Exercise 04

Introduce **int** variables `x, y, z` and **int*** **pointer** variables `p, q, r`. Set `x, y, z` to three distinct values. Set `p, q, r` to the addresses of `x, y, z` respectively.

- (1) Print with labels the values of `x, y, z, p, q, r, *p, *q, *r`.
- (2) Print the message: Swapping pointers.
- (3) Execute the swap code: `r = p; p = q; q = r;`
- (4) Print with labels the values of `x, y, z, p, q, r, *p, *q, *r`.

1.5 Exercise 05: Dynamic allocated array of integer numbers

Write a small program with the following functions:

1. Load an array of integer numbers from file.
2. Find the largest prime number in the array.
3. Sort odd numbers in ascending order, even numbers in descending order.
4. Insert a new integer number to the array that persists the above property. (do not call sorting function ☺).
5. Remove all integer values that appear twice in the array.
6. Find the gcd of all elements in the array.
7. Print out the array to screen.
8. Save the array to file.

Note:

1. In the main function, create a menu to choose the functions from 2 to 7.
2. Use OOP (class).
3. Use dynamic array to store integer numbers.

1.6 Exercise 06

1.6.1 Technical requirements

OOP

C++ File I/O

Pointer (dynamic allocated array)

1.6.2 Problem

You've got array $a[1], a[2], \dots, a[n]$, consisting of n integers. Count the number of ways to split all the elements of the array into three contiguous parts so that the sum of elements in each part is the same.

More formally, you need to find the number of such pairs of indices i, j ($2 \leq i \leq j \leq n - 1$), that $\text{sum}(a[1], a[i-1]) = \text{sum}(a[i], a[j]) = \text{sum}(a[j], a[n])$

Input

The first line contains integer n ($1 \leq n \leq 5 \cdot 10^5$), showing how many numbers are in the array. The second line contains n integers $a[1], a[2], \dots, a[n]$ ($|a[i]| \leq 10^9$) — the elements of array a .

Output

Print a single integer — the number of ways to split the array into three parts with the same sum.

Sample

Input:

5

1 2 3 0 3

Output:

2

2 A03

Problem 1, 2, 3

tploc/htthanh@fit.hcmus.edu.vn

3 H03

Problem 1, 2, 3, 4, 5, 6