# Topic 7: Pointers

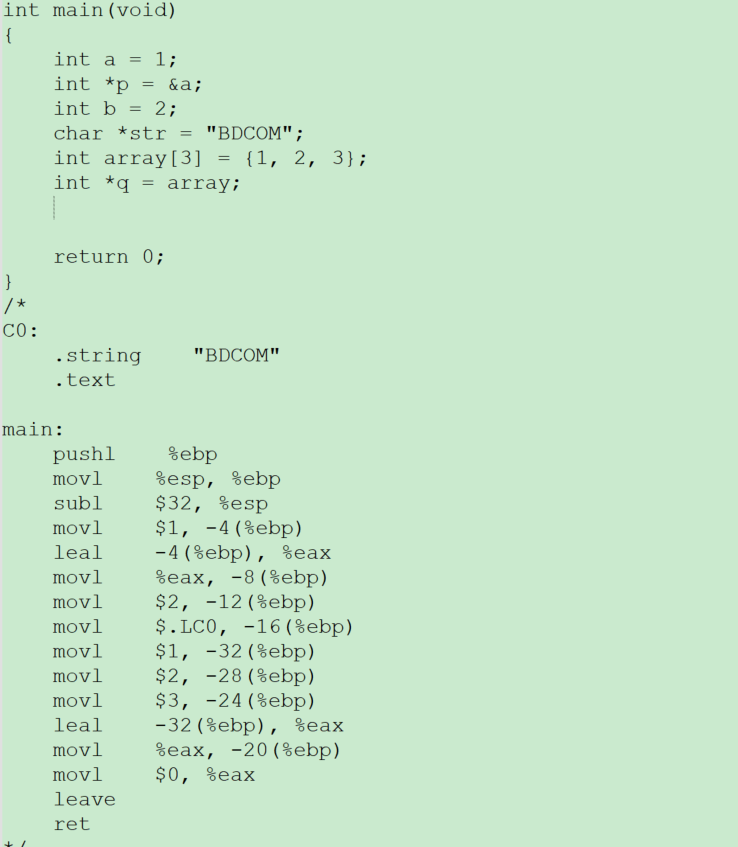
## Basic Knowledge

### Guidance

Pointers On C.pdf chapter 6.1-6.8

### Practice

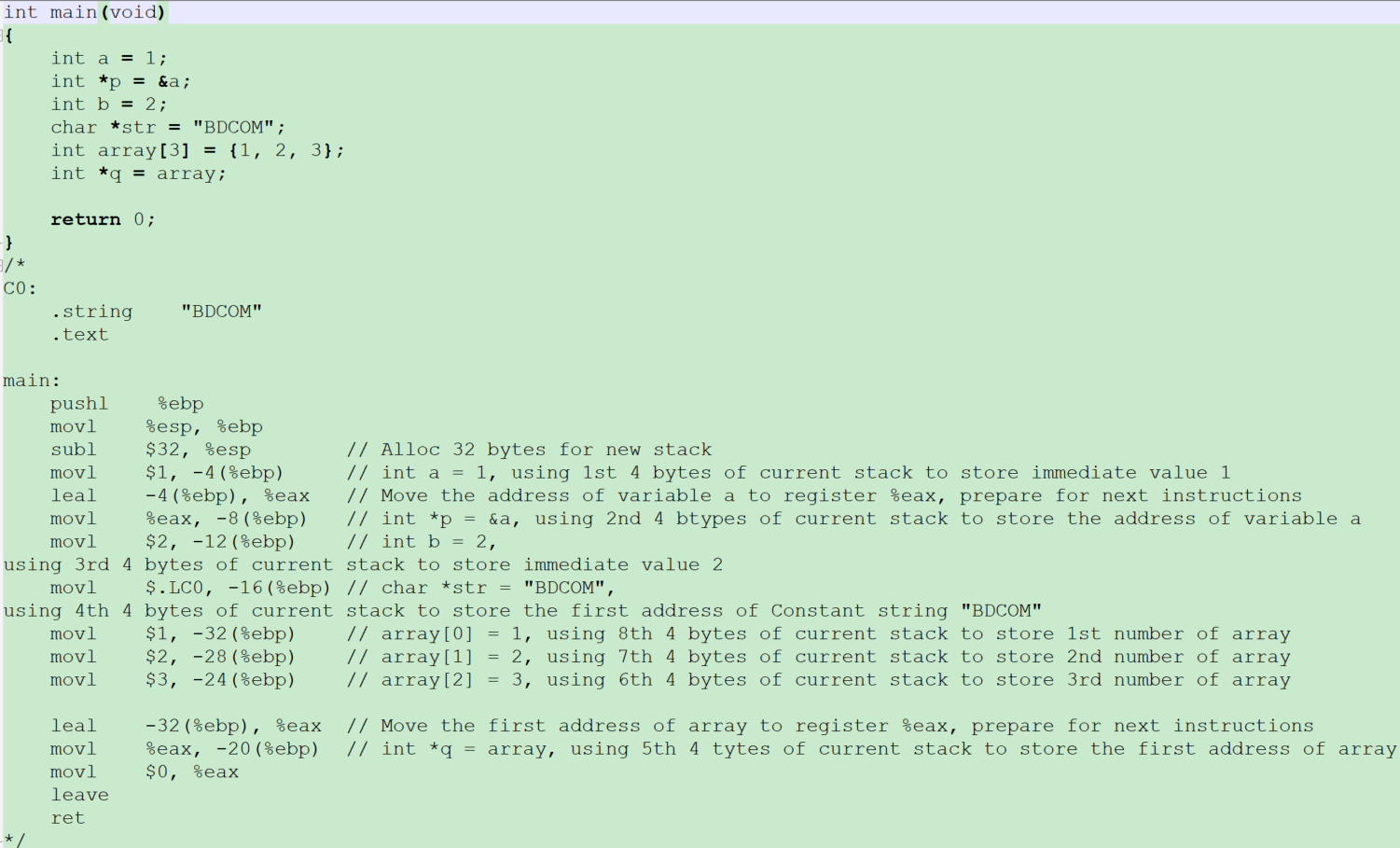
1. What can you get from the following program? (From the perspective of variable content). Write the corresponding C code next to the assembly instruction



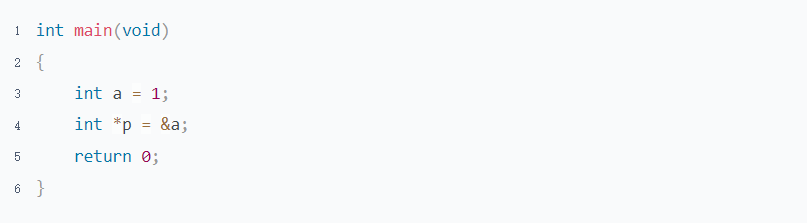
The content of normal variable are numbers that resemble immediate value

The content of pointer variable are address of normal variable or other address

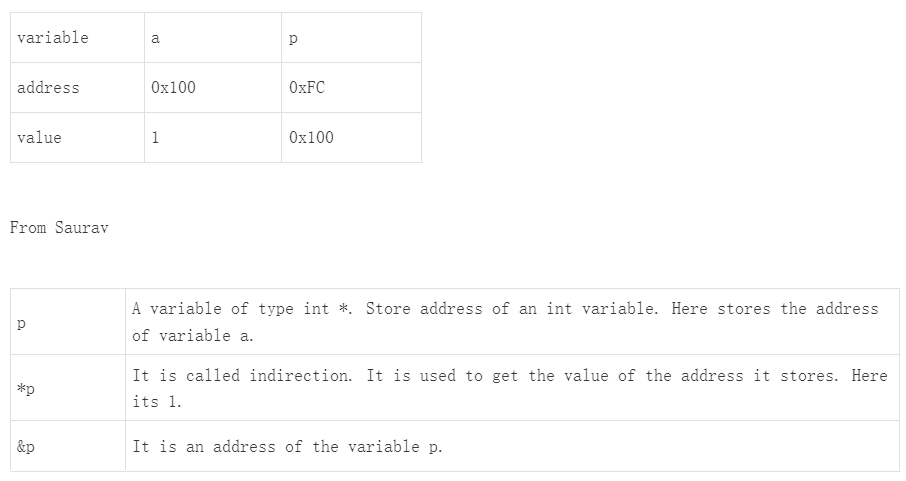
An array is an ungapped space whose elements are pushed onto the stack one by one. The pointer only needs to hold the first address of the array if you want to manipulate the array through the pointer



1. There is a program, for p, \*p and &p, Answer following questions



* 1. What are the meaning of them? Draw a picture to illustrate



* 1. Which one can be used as the l-value?

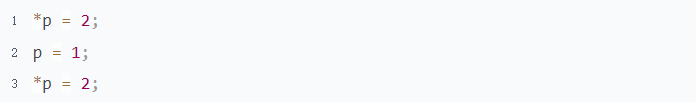
As i said above,

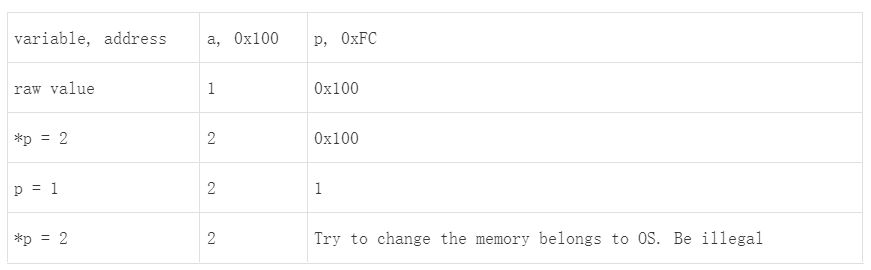
p = , means store a value to the address of p. The value of p changed from 0x100 to

\*p = ,means store a value to the address of p point to. The value of a changed from 1 to

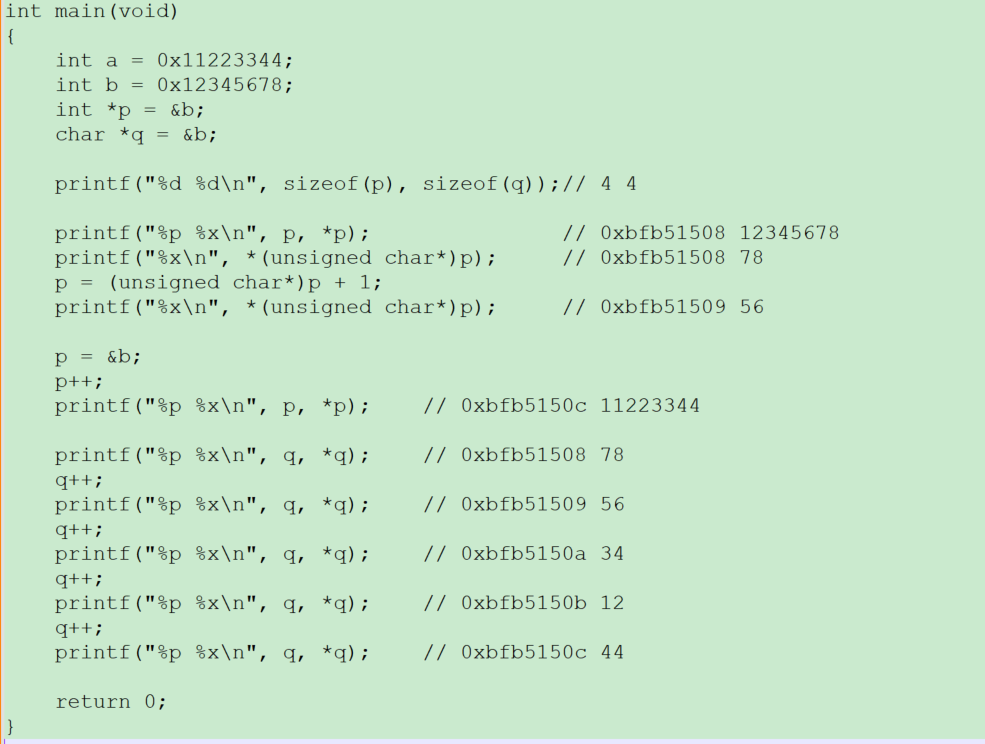
&p is an address. It is a constant and can not be move.

* 1. What does each line in the following program mean? Based on last program





1. What can you get from the following program? (From the perspective of pointer data type)

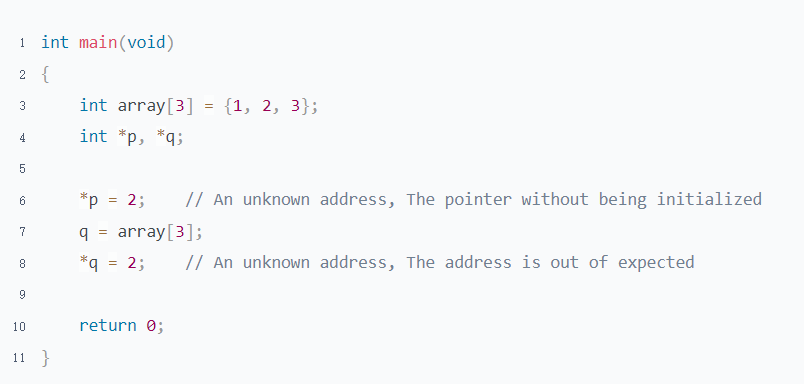


The step size of a pointer depends on its type.

The operation of a pointer is not limited by its type and can be converted to any type at any time if desired.

1. What is a wild pointer? Illustration

Wild pointer is a pointer that without initialized before using.



1. What is dangling pointer? Illustration

Dangling pointer is a pointer that has been released or no longer valid but is still in use.



## Pointer & Struct

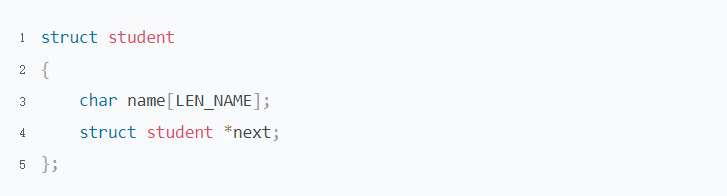
### Guidance

Pointers On C.pdf chapter 11 12

### Practice

1. What are the benefits of using pointer for struct? Make a list of all the scenarios you can think of

* Self referential, Pointers On C.pdf->10.1.5



* Passing by address when call function. Which is more efficient than passsing by value.
* Data structuring. Refer to next question

1. Redo the question 7 of topic "Struct Bit Fields & Alignment" using struct pointer.(Do not need draw any picture, only program)

Refer to “Receive & Send packe”

1. Seat Management & Reservation System

Requirement

* There is no platform restriction, but you need to give comment at the beginning of your program
* Arrays are not allowed to be used
* Default empty seat is 3. Which did not booked by any customer.
* The number of seat is limited to 10.
* Do not need to distinguish between first name and last name. All we need is a name
* A customer can only reserve 1 seat. The identification of a customer is the person id. Different person can have the same name





## Pointer & Array

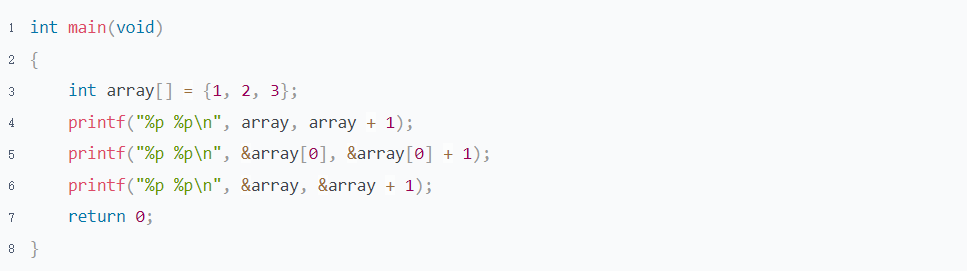
### Guidance

Pointers On C.pdf 6.13.1 8.1

1. What is the difference between array, &array[0] and &array
2. array initialization
3. What happens when an array is the argument to a function?

### Practice

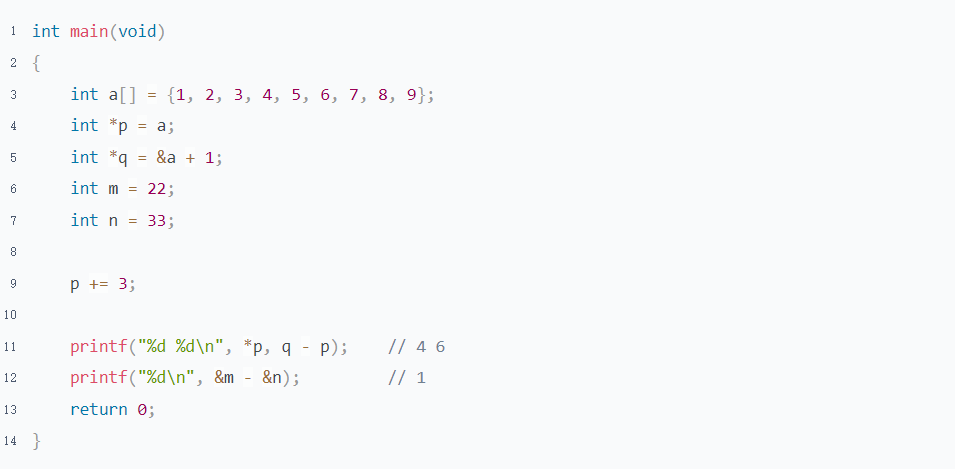
1. What can you get from the following program? Think about it. Why is it designed that way



array and &array[0] have the same meaning, they both represent the first element address of array. Its type is "int \*". Which size is 4 bytes on 32 bit platform.

&array is equivalent to int (\*array)[3]. Which is a pointer to array. Its step size is an array.

1. Why does the following program have such an output? Does it make sense?



Pointer operations should only occur within array ranges or between two pointers pointing to the same array.

Pointer perations outside the scope of the array are meaningless.

1. Why does the following program have such an output?



The array will auto convert to a pointer to the first element of the array when it as a r-value

1. Why does the following program have such an output?

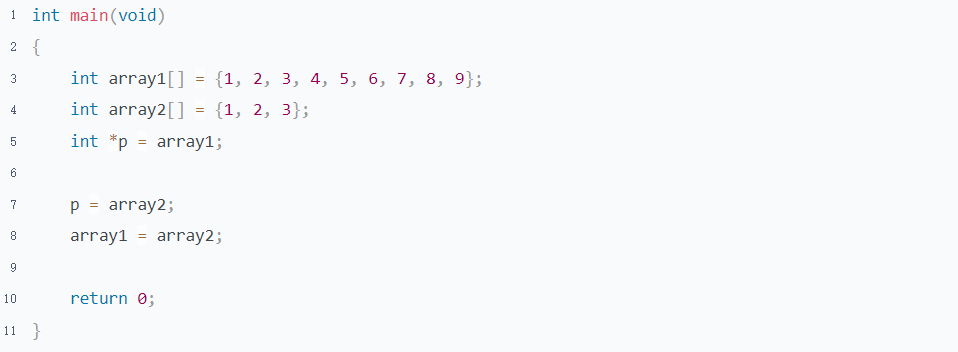


The output of &p and p are different. But The output of &array and array are same.

There are 2 reasons for this

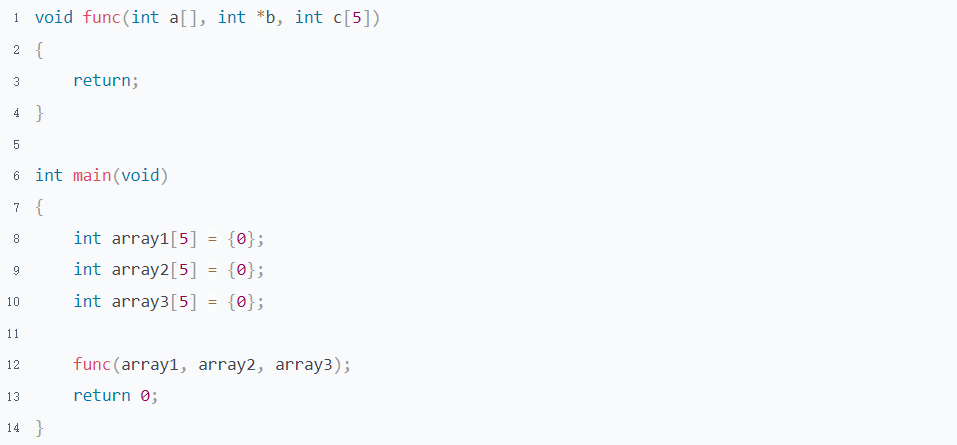
* The value or address of pointer is different memory. But The array has only one space. So we use the array name to represent the first element address
* p and array have the same meaning, they both represent the address they store. Array is a sequential memory. Represented by the address of the first element(&array)

1. Why can not array be assigned just like pointer?



The array name cannot be used as an l-value, otherwise the array will be accessed improperly and cannot be released

1. What is the difference between a, b and c？



The formal parameter a,b, and c are all pointers to the compiler; The difference is that a[] and c[5] code are more readable and help programmer understand the meaning of parameters more quickly

1. Do experiment on Linux for each step of 8.1.4 on “Pointers On C.pdf”

## Pointer & Array Enhancement

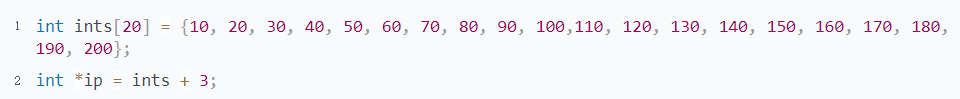
### Guidance

Expert C Programming.pdf 4

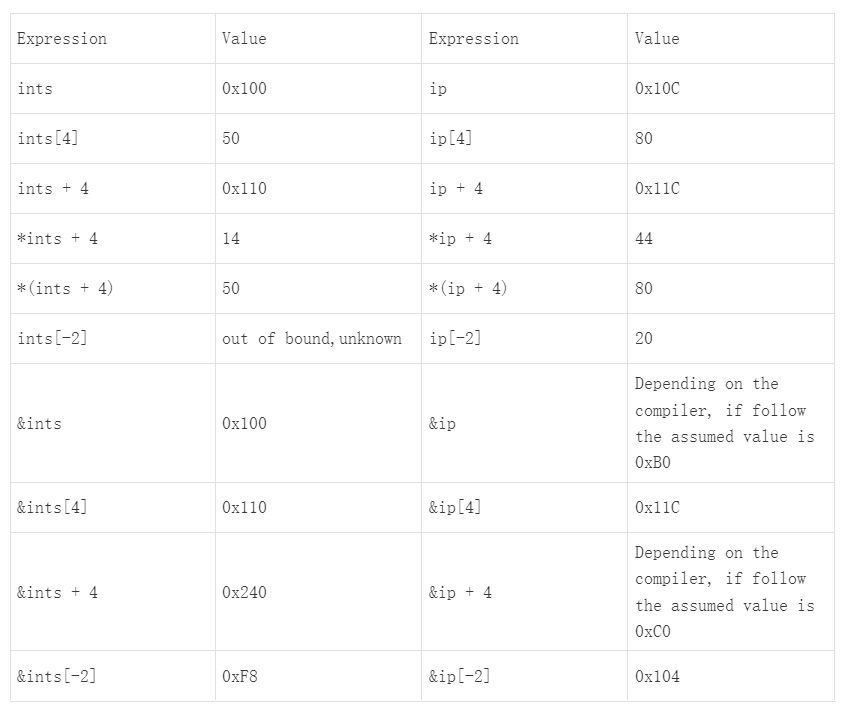
C Primer Plus.pdf 10

### Practice

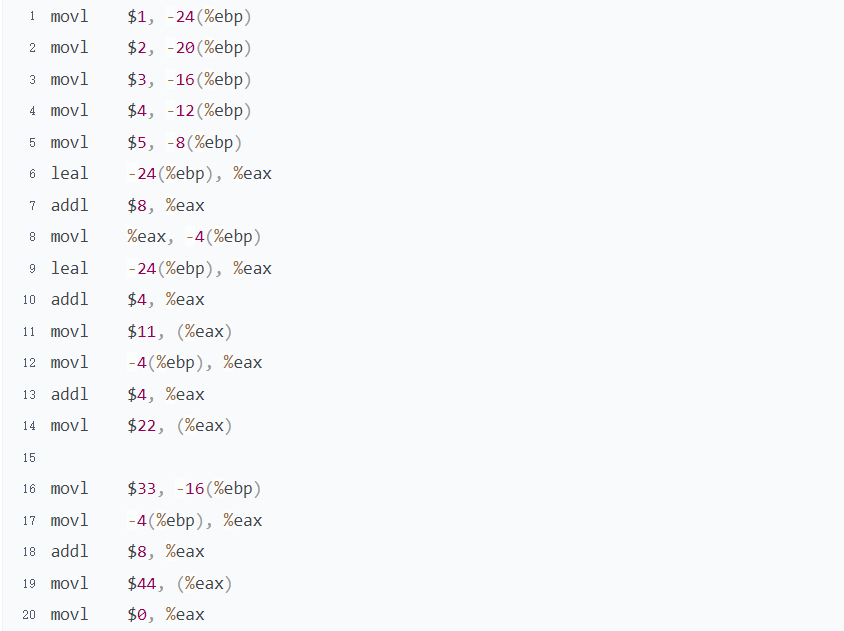
1. Pointers On C.pdf 8.7 1



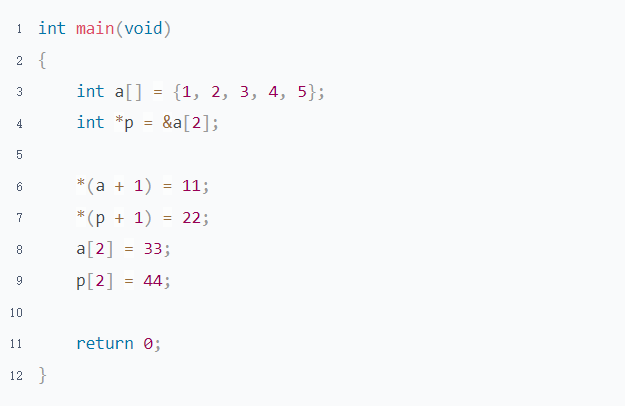
Assume that the ints array begins at location 100, the location of ip is next to array. and that integers and pointers both occupy four bytes. Addresses are sorted from highest to lowest



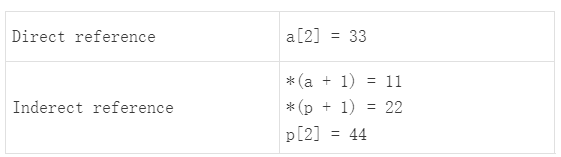
1. Pointers On C.pdf 8.7 5
2. Here is a program, Answer some questions



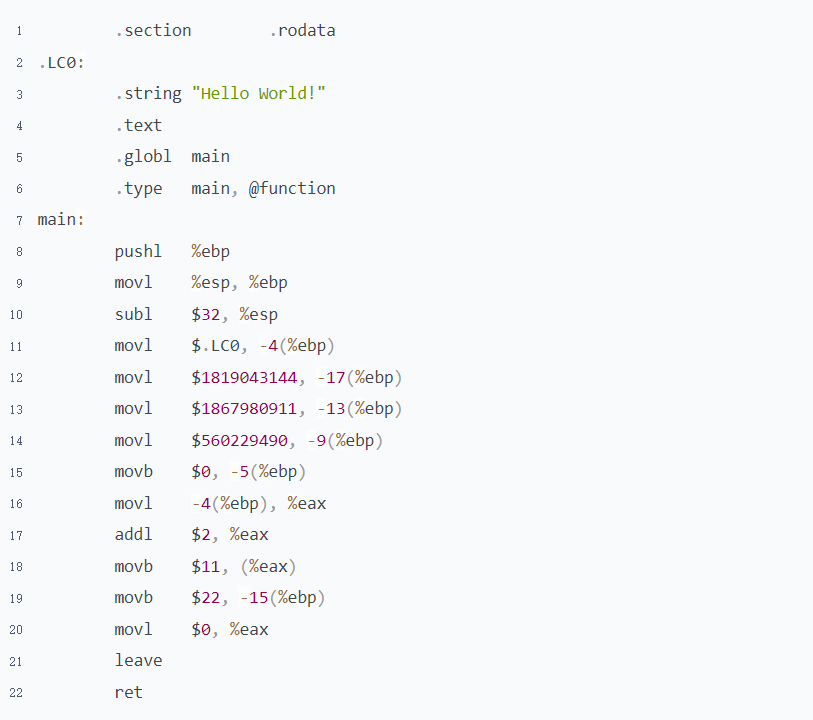
* 1. Convert the following assembly language to C language



* 1. What can you get from the following program?
  2. In the following program, Which are the direct references? What are indirect references?



1. Convert the following assembly language to C language, What can you get from the following program?





## Pointer and Function

### Guidance

The C Programming Language 2nd.pdf 5.2

Computer Systems A Programmers Perspective (3rd).pdf 3.4.2 3.5.1

### Practice

1. What can you get from the following program?



Pointer usage is not affected by its type. Any pointer type can be converted to other types at any time according to requirements

1. There is a program in file " Pointer and Function-Practice", Answer following questions
   1. Write the corresponding C code next to the assembly statement

Refer to "Pointer and Function-Pracitce.c", Obtained from Liakot and Dipto

* 1. Draw some pictures to show the memory changes after each assembly instruction is executed

Refer to "Pointer and Function-Practice.xlsx", Obtained from Liakot

* 1. You may have heard that there are two ways to pass parameters when a function is called. Pass values and references. What do you think about it in C?

Passing values and passing addresses are value copies, there is no difference in essence

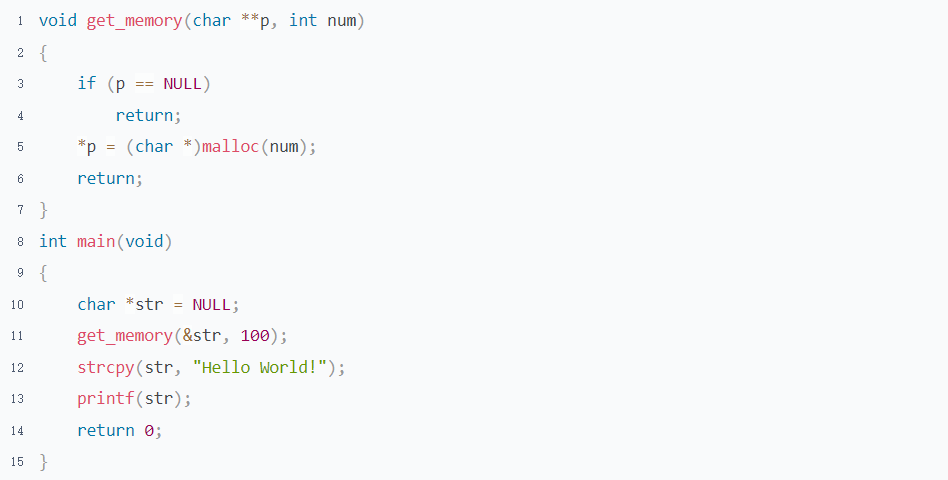
1. What is the problem of the following program?
   1. The start of new space be put into the copy address of str. So the address of str is not changed.



* 1. The space be allocated on the stack. When stack released, It can no longer be used



* 1. The Standard library function malloc and free must be used in pairs



# Topic 8: Advanced Pointers

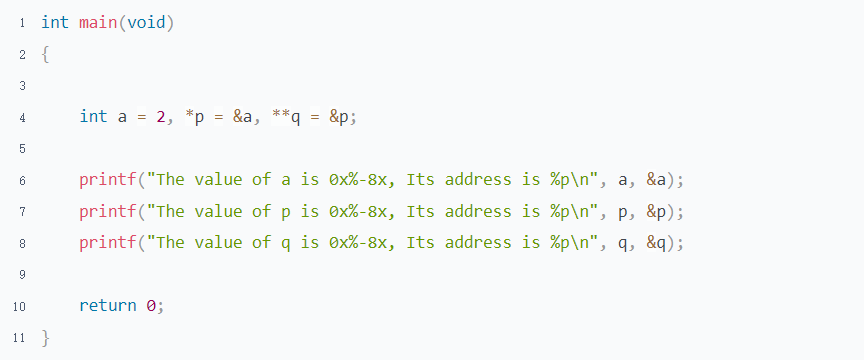
## Pointer to Pointer & Array of Pointer

### Guidance

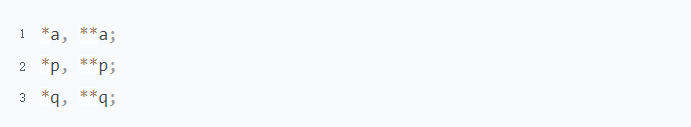
Pointers On C.pdf chapter 6.10 8.3 13.1 13.4

### Practice

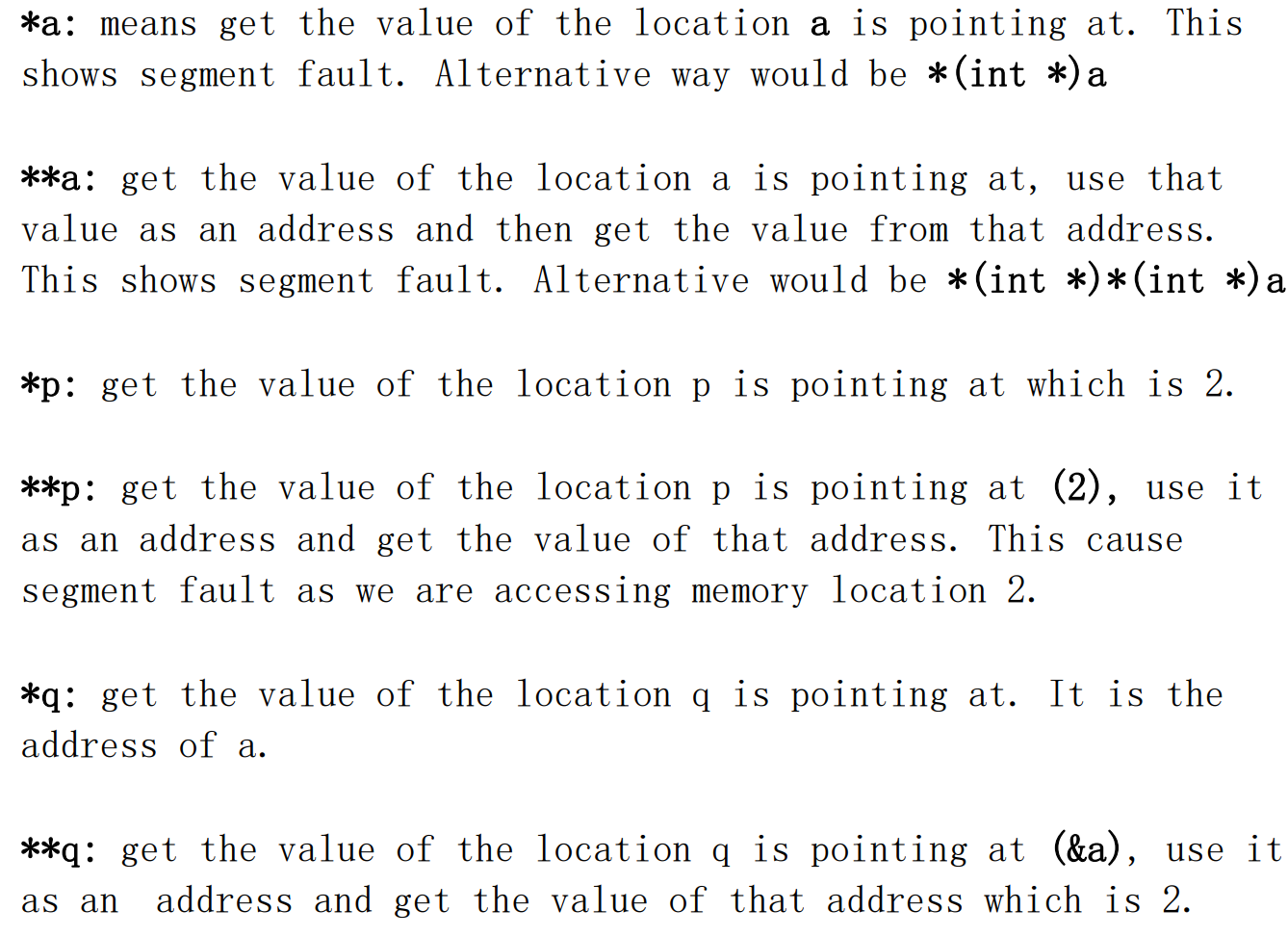
1. Here is a program, Answer some questions



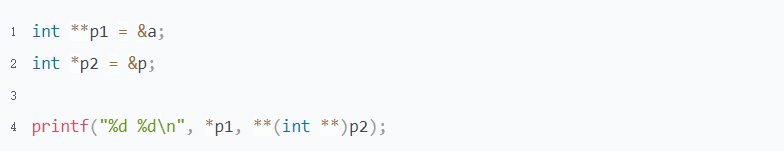
* 1. Draw a picture to show the relationship between these 3 variables
  2. What do the following expressions mean



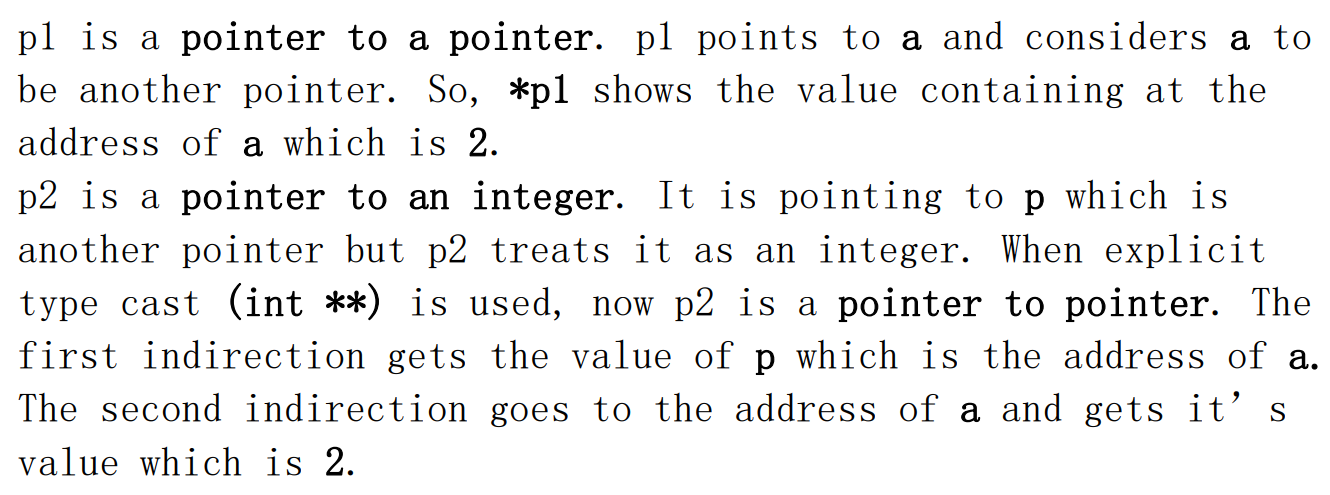
From Dipto



* 1. Based on the previous program. What can you get from the following program?

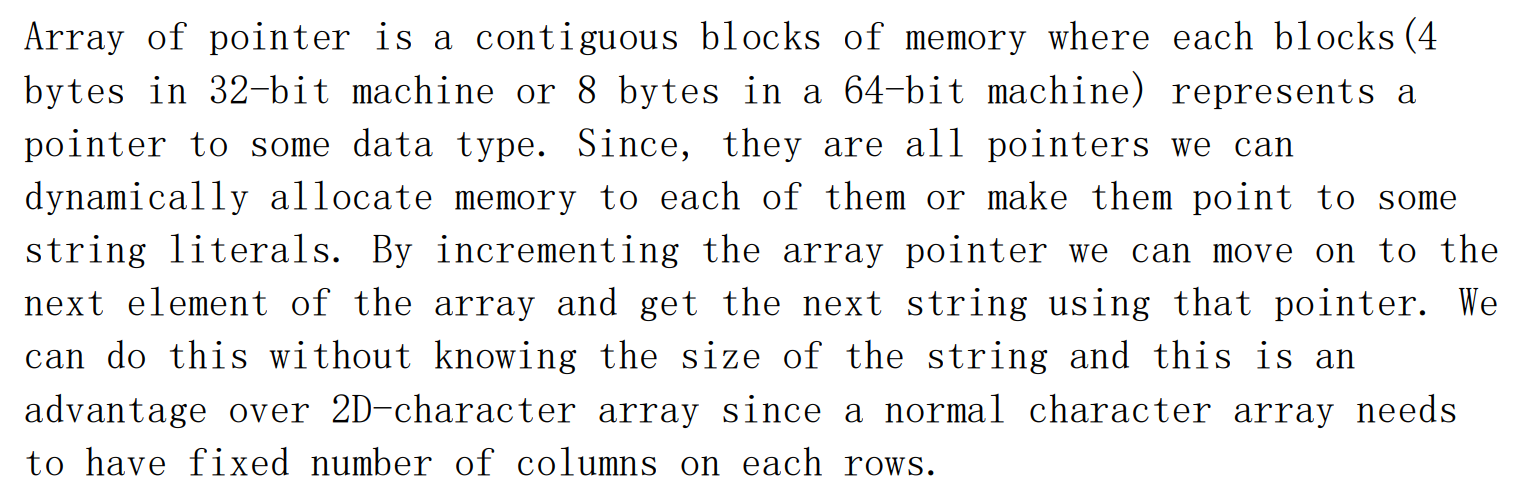


From Dipto

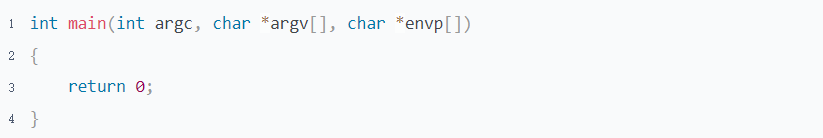


1. What is array of pointer? Why do we need it? What is the difference between array of pointer and normal array?

From Dipto



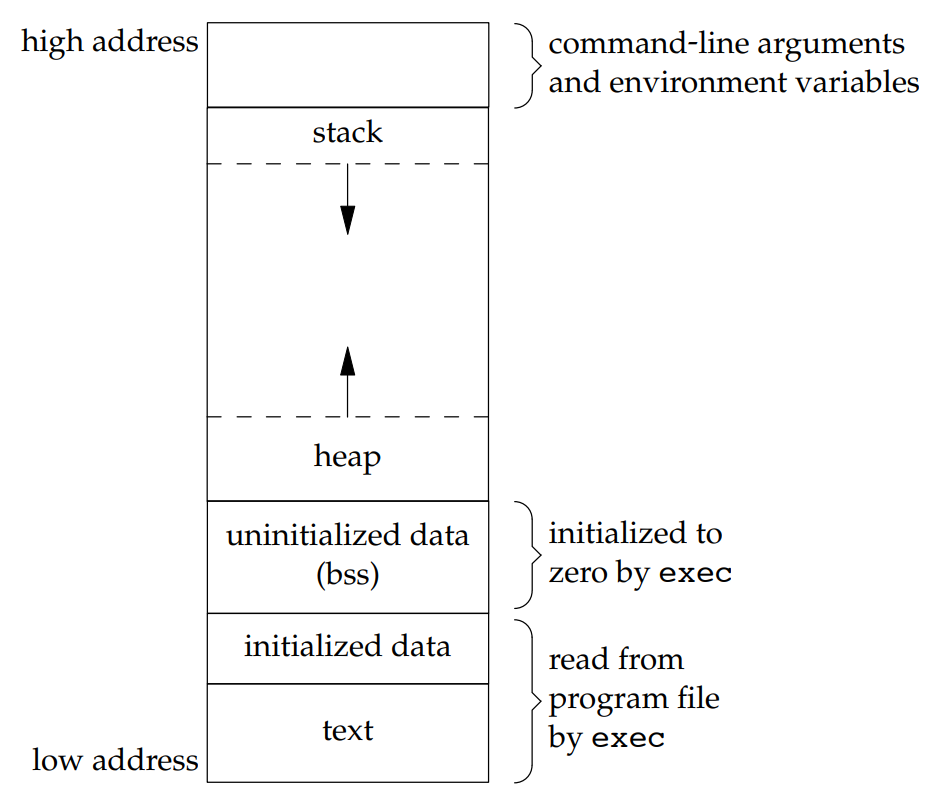
1. Read the program below, Answer some questions



* 1. What kind of data is stored in argv and envp? Where are they stored in memory?

argv and envp are both array of pointer. They store different kind of string. Get from parent process.

They are stored between stack and kernel memory.



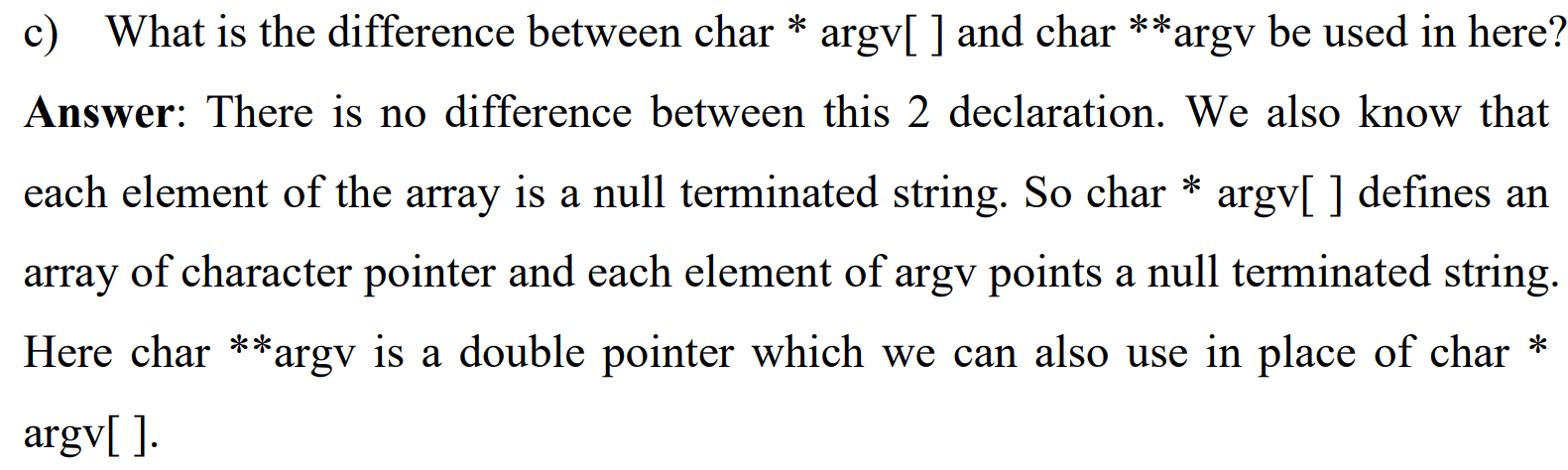
* 1. How many ways to get the content of argv?

From Liakot,We can get the content of the argv using 2 ways



* 1. What is the difference between char \*argv[] and char \*\*argv be used in here?

From Rakin

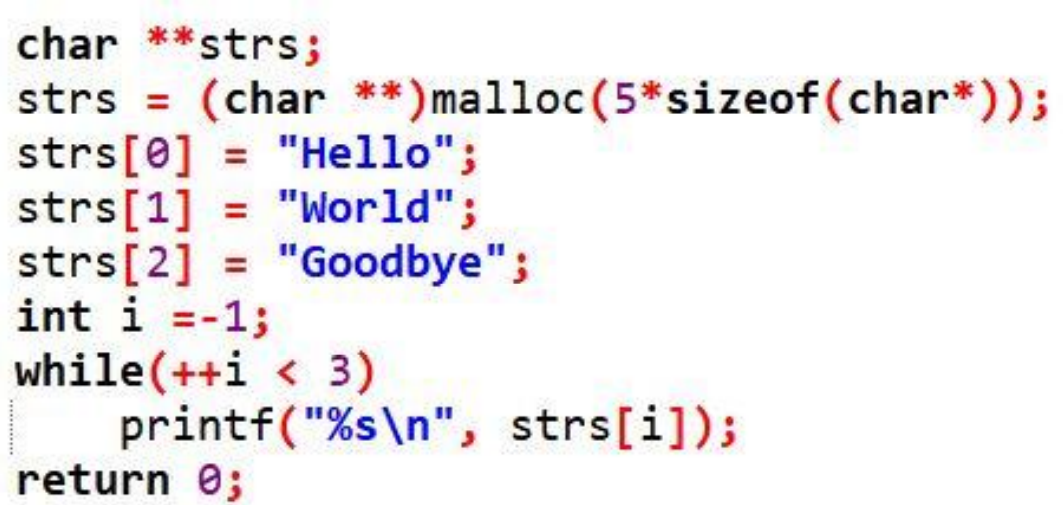


1. Which Scenario the pointer to pointer may be used?

From Liakot



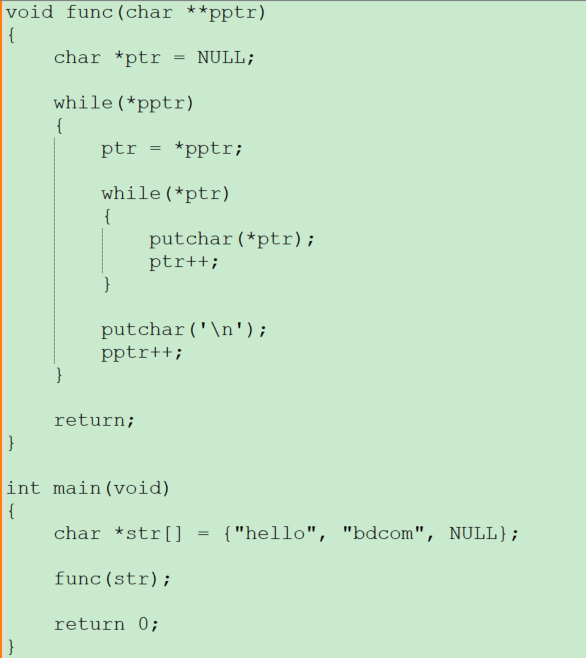




1. What is the relationship between array of pointer and pointer to pointer?

There are two associations

* We can indirect each of element of array of pointer by pointer to pointer
* They are same in some situation, ex. As formal parameter of a function



## Pointer to Array & Multi-Dimensional Array

### Guidance

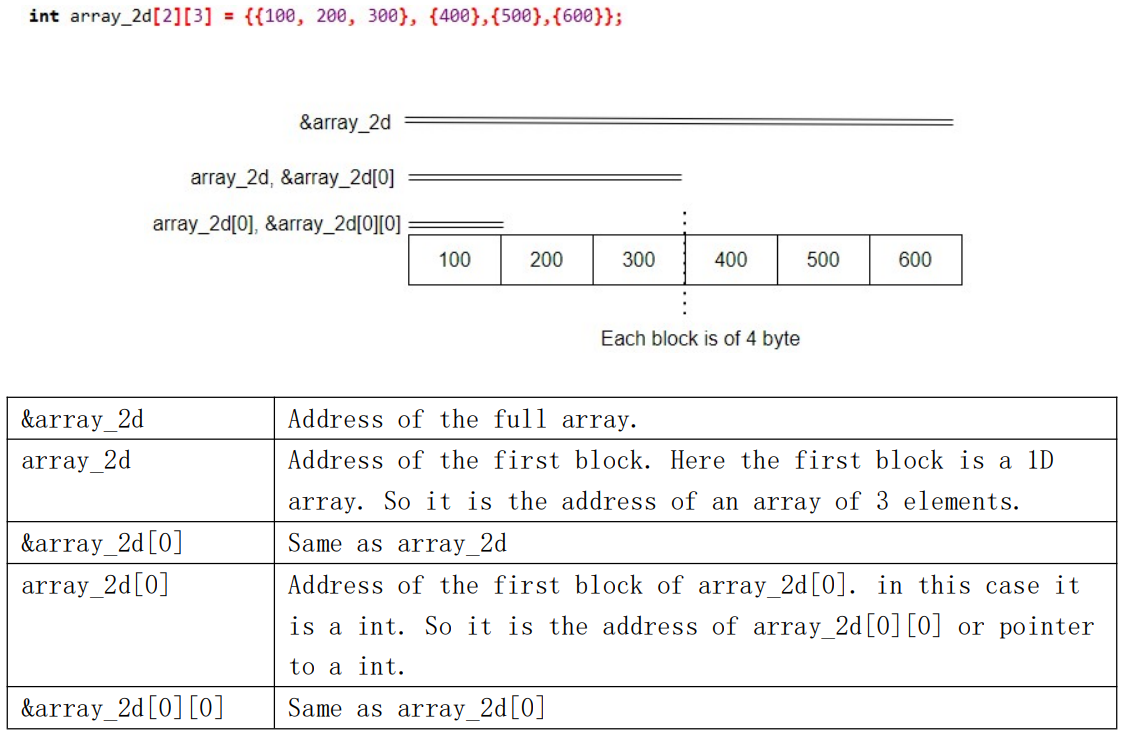
Pointers On C.pdf 8.2

Understanding And Using C Pointers.pdf chapter 4

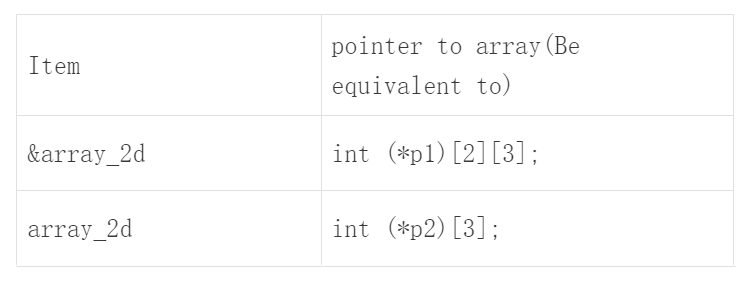
### Practice

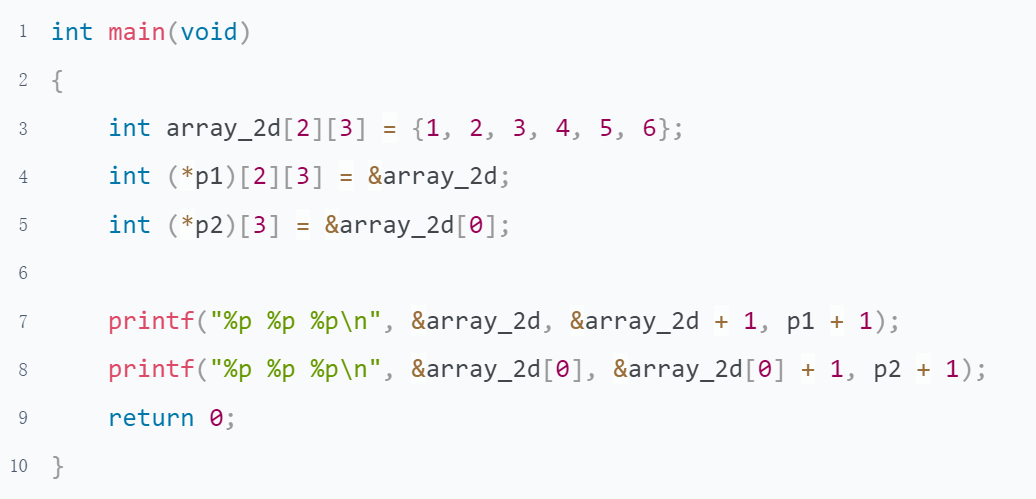
1. What are the meaning of different name of multi-dimensional
   1. There is a two-dimensional array, int array\_2d[2][3], What does the mean of &array\_2d、array\_2d，&array\_2d[0]，array\_2d[0]，&array\_2d[0][0].

From Saurav

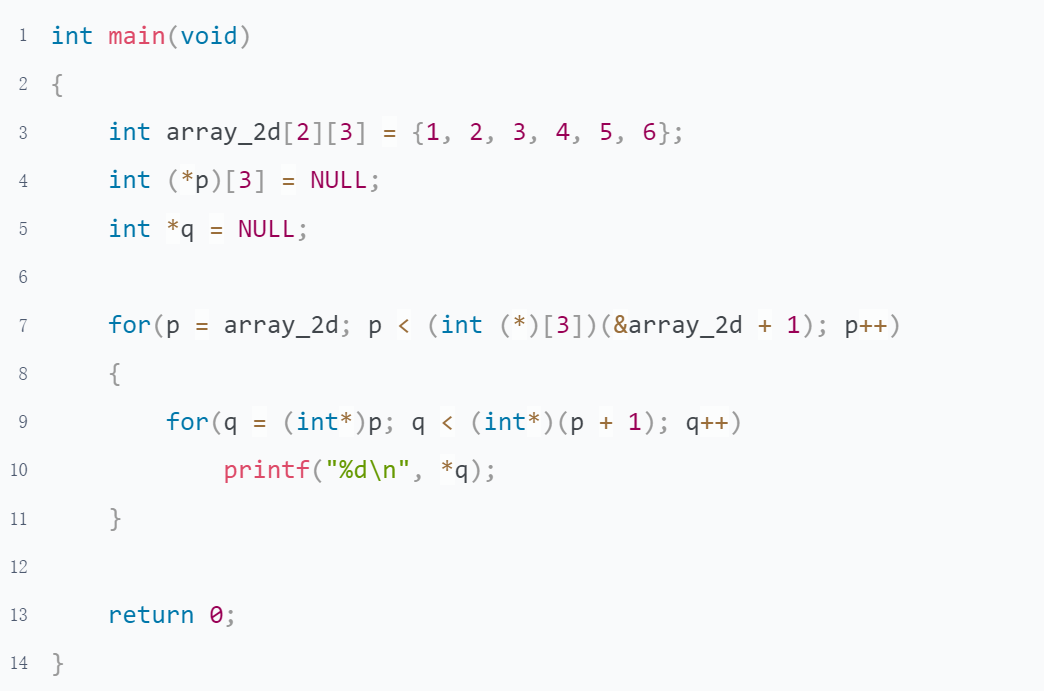


* 1. What are the relationship with pointer to array? List a table





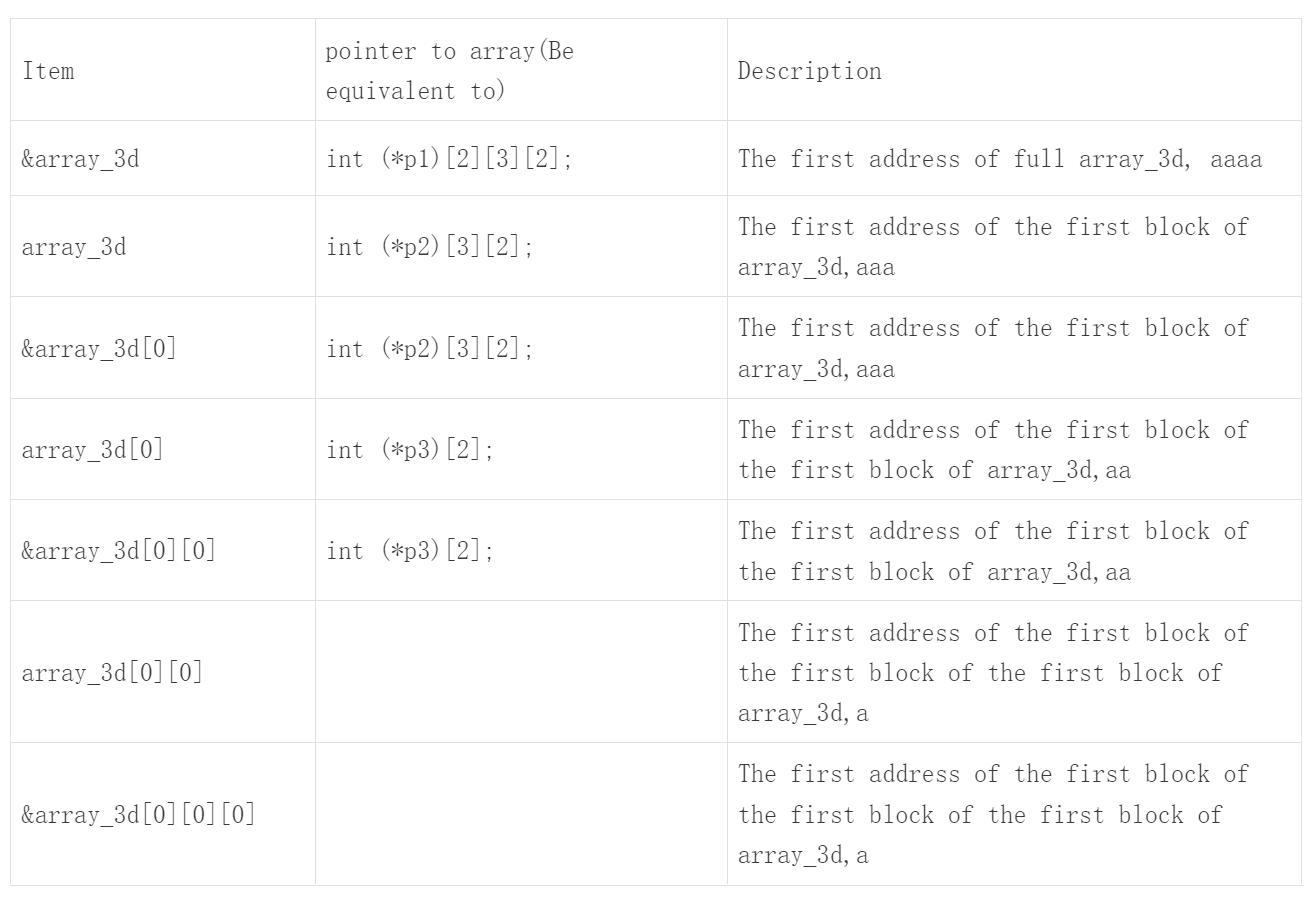
* 1. How to access each element of two-dimensional array through pointer to array?



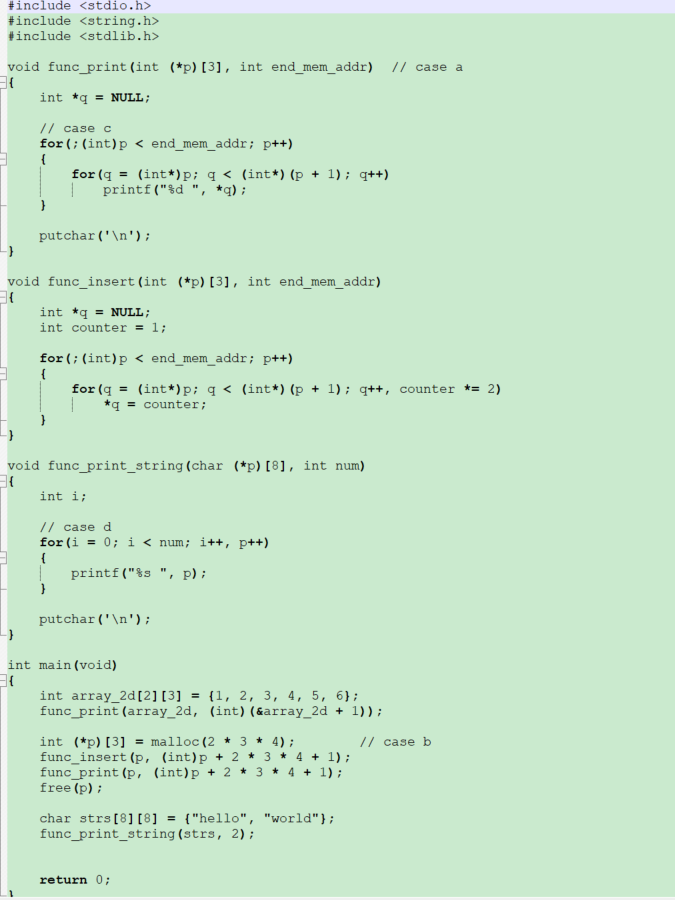
* 1. What about three dimensional array? List all the meaning of first address of array\_3d

int array\_3d[2][3][2];



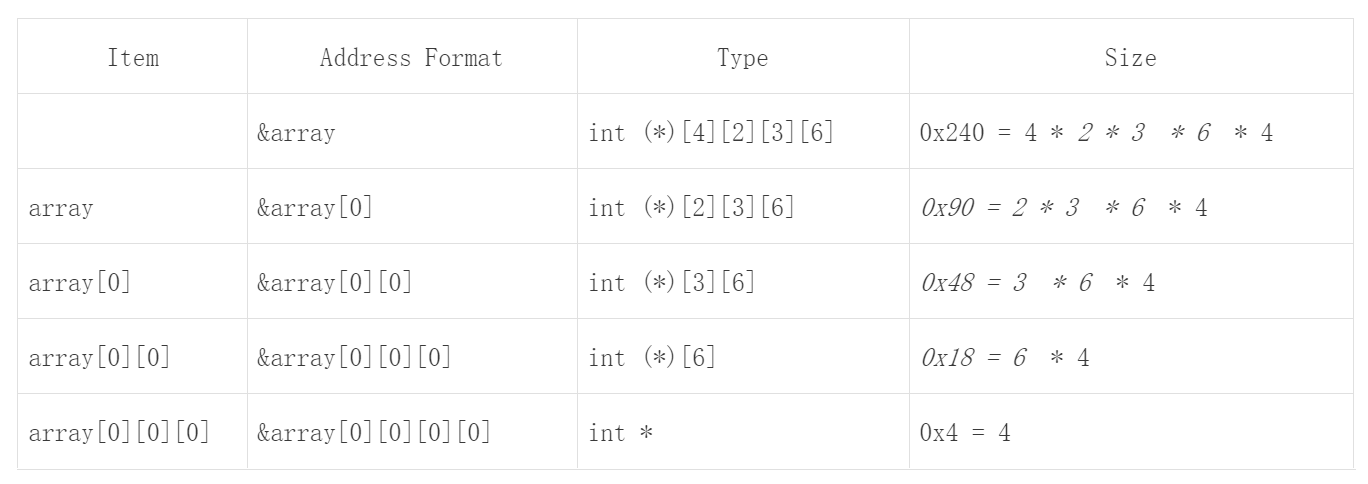


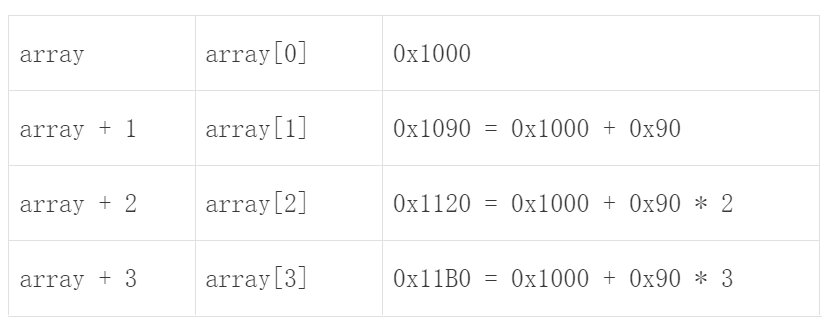
1. Why do we need pointer to array? what is it used for?
   1. For passing multidimentional array to a function.
   2. Creating dynamic array.
   3. Pointer arithmatic improves code efficiency
   4. Easy string manipulation

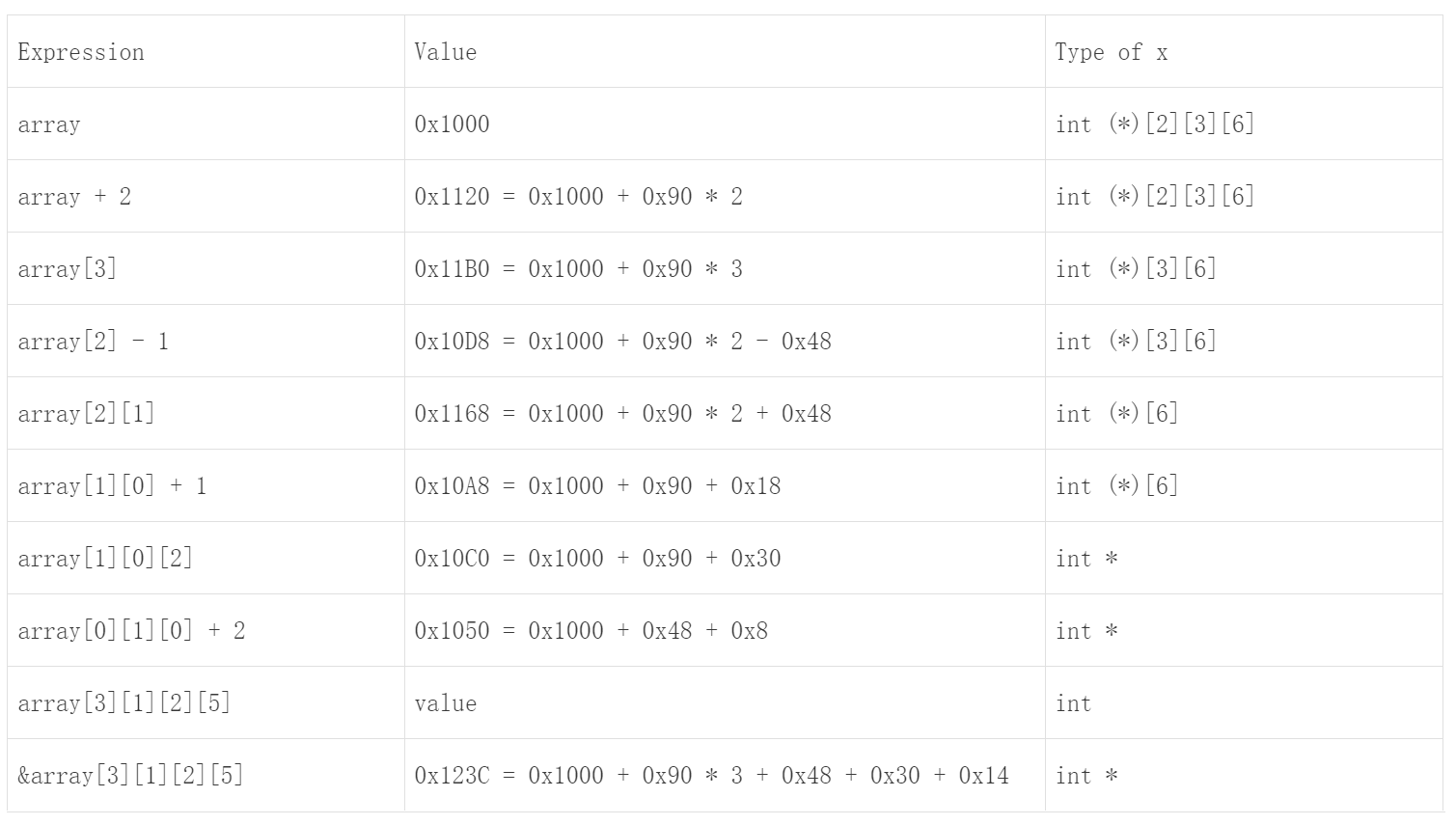


1. Pointers On C.pdf 8.7 11

int array[4][2][3][6];

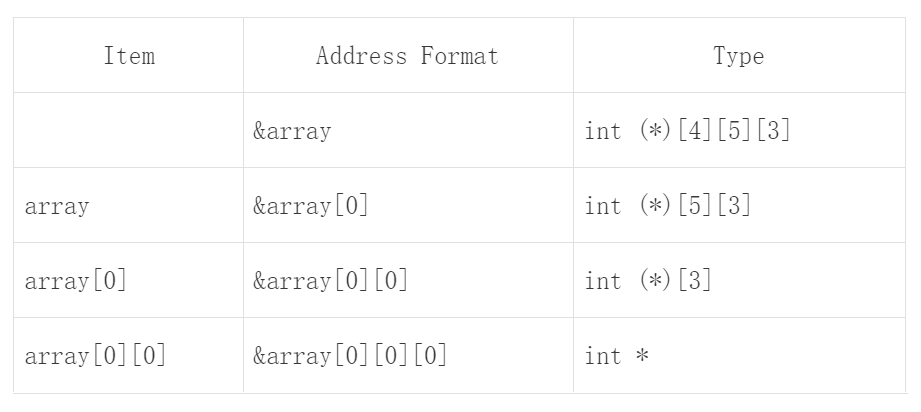


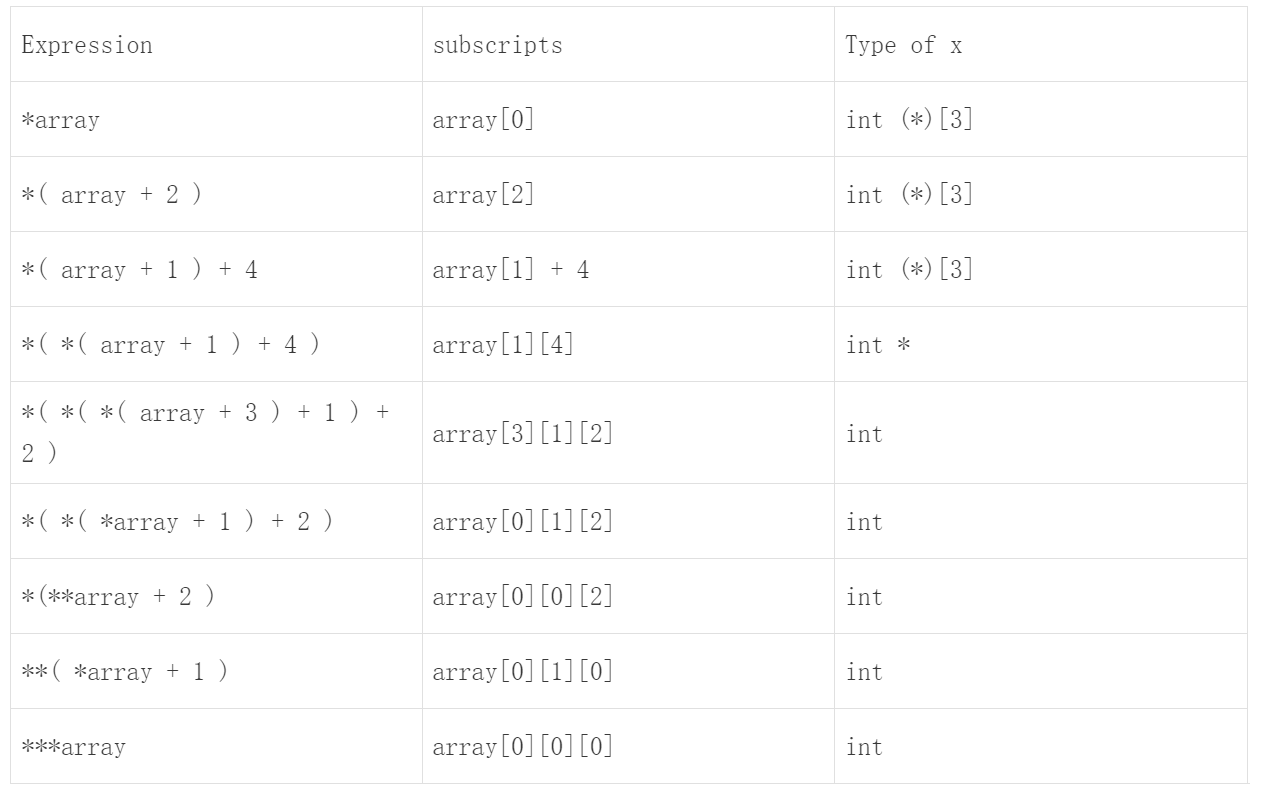




1. Pointers On C.pdf 8.7 13

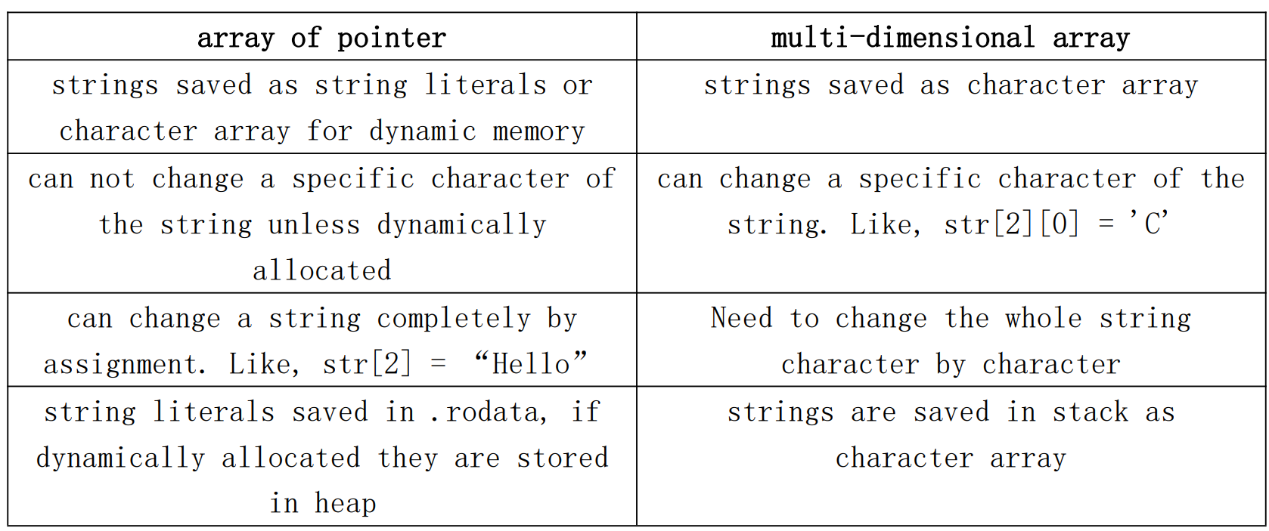
int array[4][5][3];



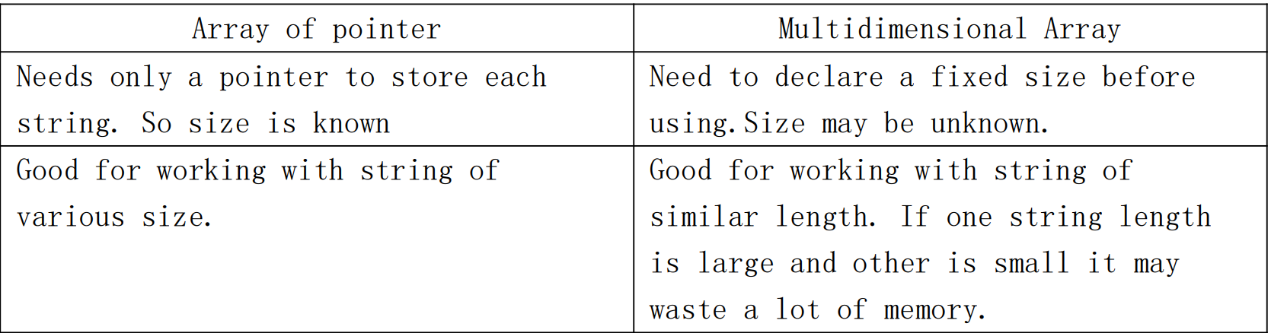


1. What is the difference between array of pointer and multi-dimensional array for string store and process?

From Dipto



From Saurav



## Pointer to Function

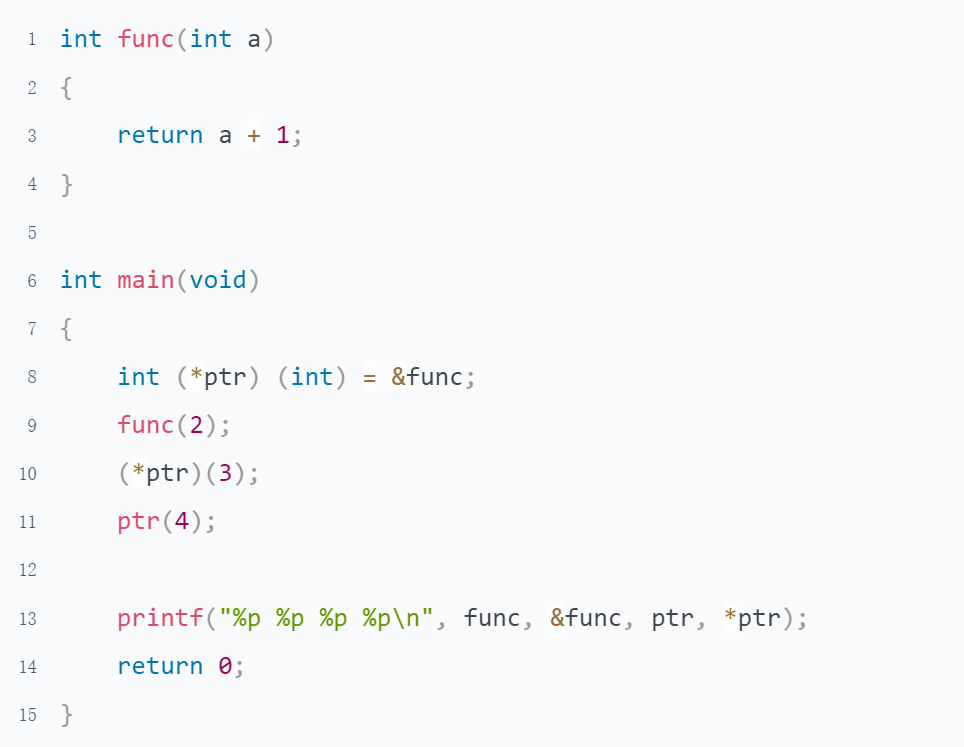
### Guidance

Understanding And Using C Pointers.pdf chapter 3

Pointers On C.pdf 13.3

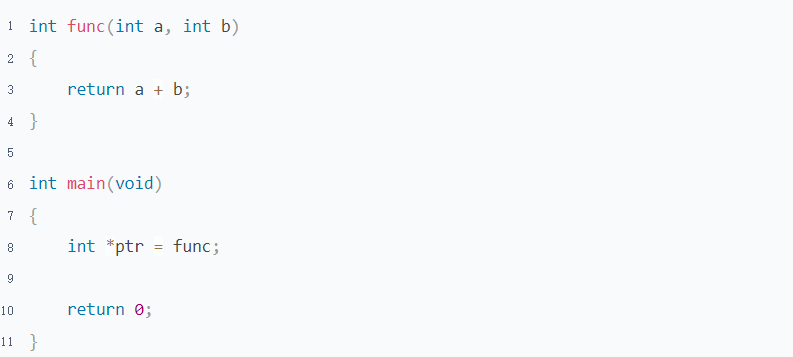
### Practice

1. Refer to P356 on "Pointers On C.pdf", Write a program of your own, through the assembly code to see, What is the difference between these 3 ways



There is a quesion, why are these four addresses the same?

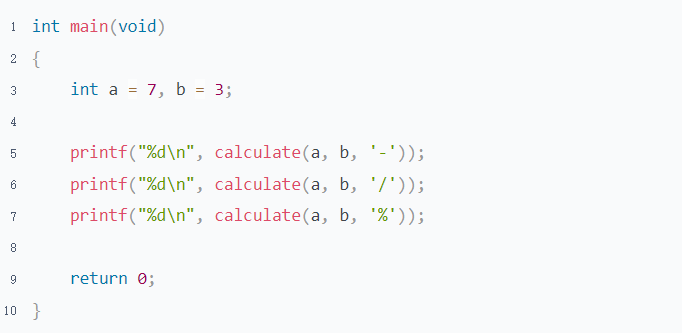
1. How to call func by pointer ptr



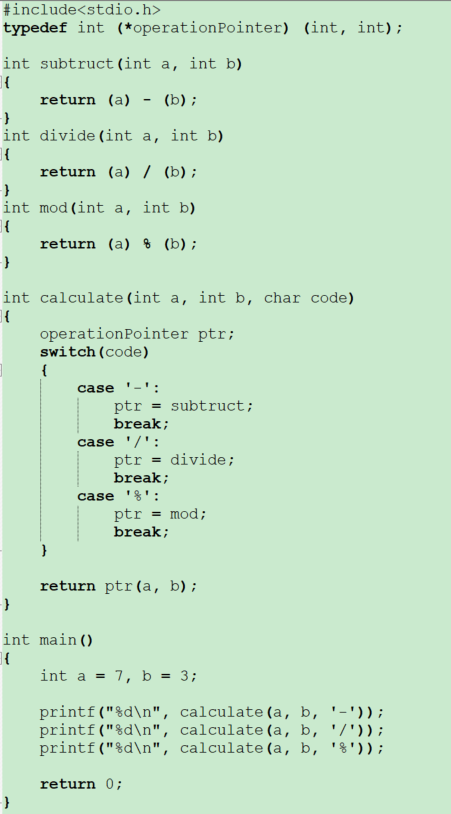
explicit type conversion

printf("%d\n", ((int (\*)(int, int))a)(2, 3));

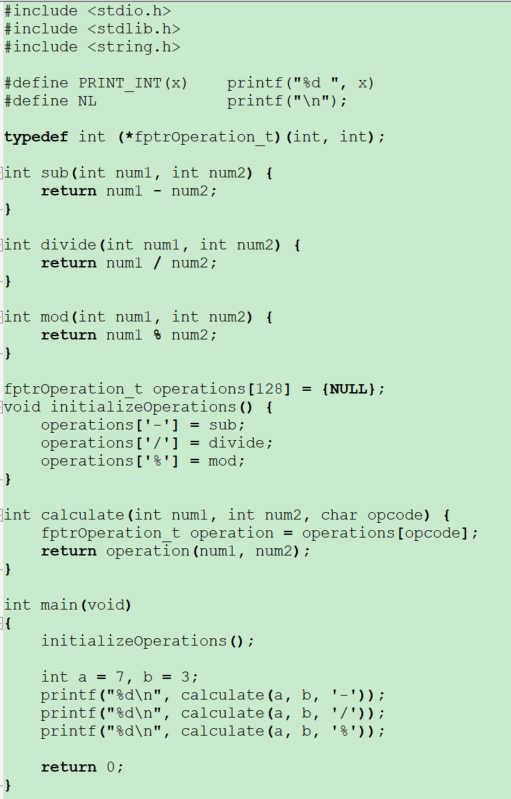
1. Implement the following program with pointer to function



Method 1, From Liakot



Method 2,From Mormo

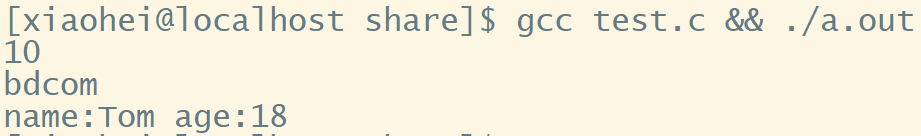


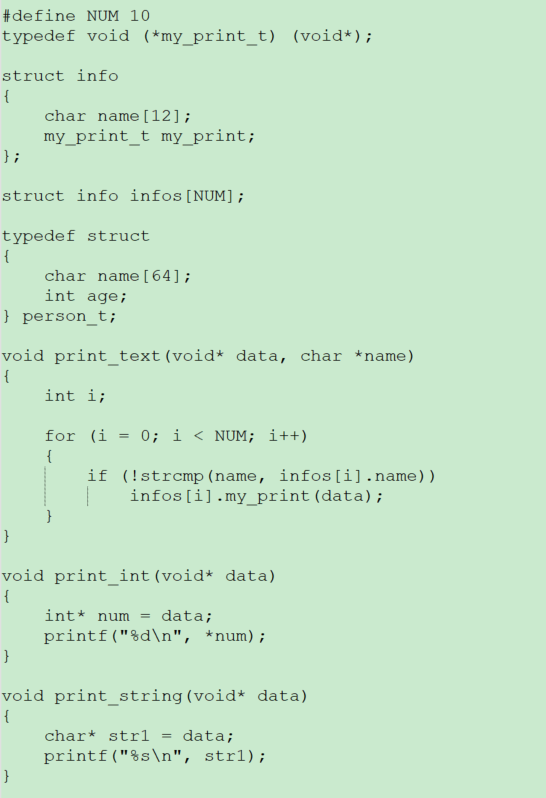
1. Implement the Jump Tables on page 360 of book "Pointers On C.pdf"

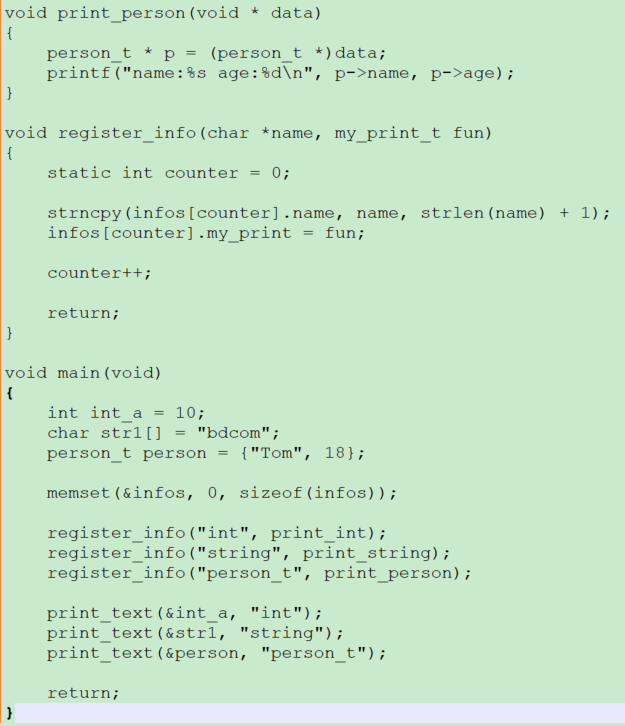


1. Pointers On C.pdf 13.10 3
2. The main function and output have already provided, To complete the remaining parts.









## Pointer Enhancement

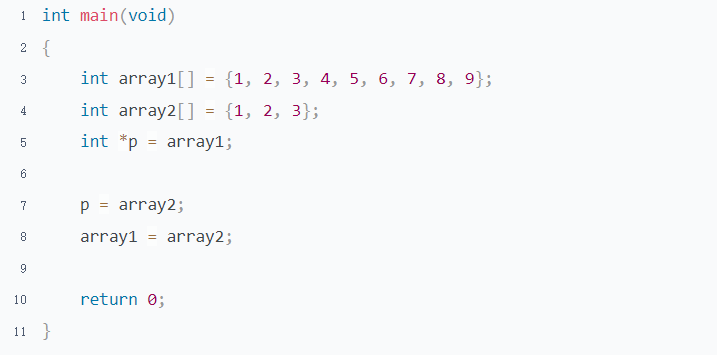
### Guidance

The C Programming Language 2nd.pdf chapter 5

1. constant pointer
2. step size of pointer
3. complicated declarations of pointer

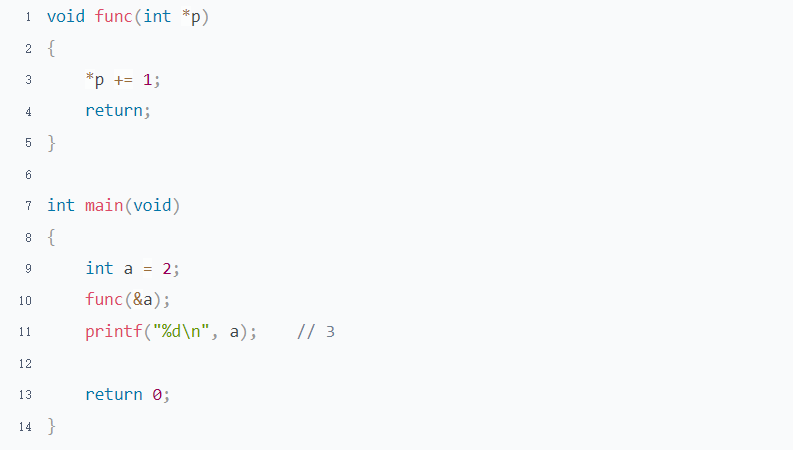
### Practice

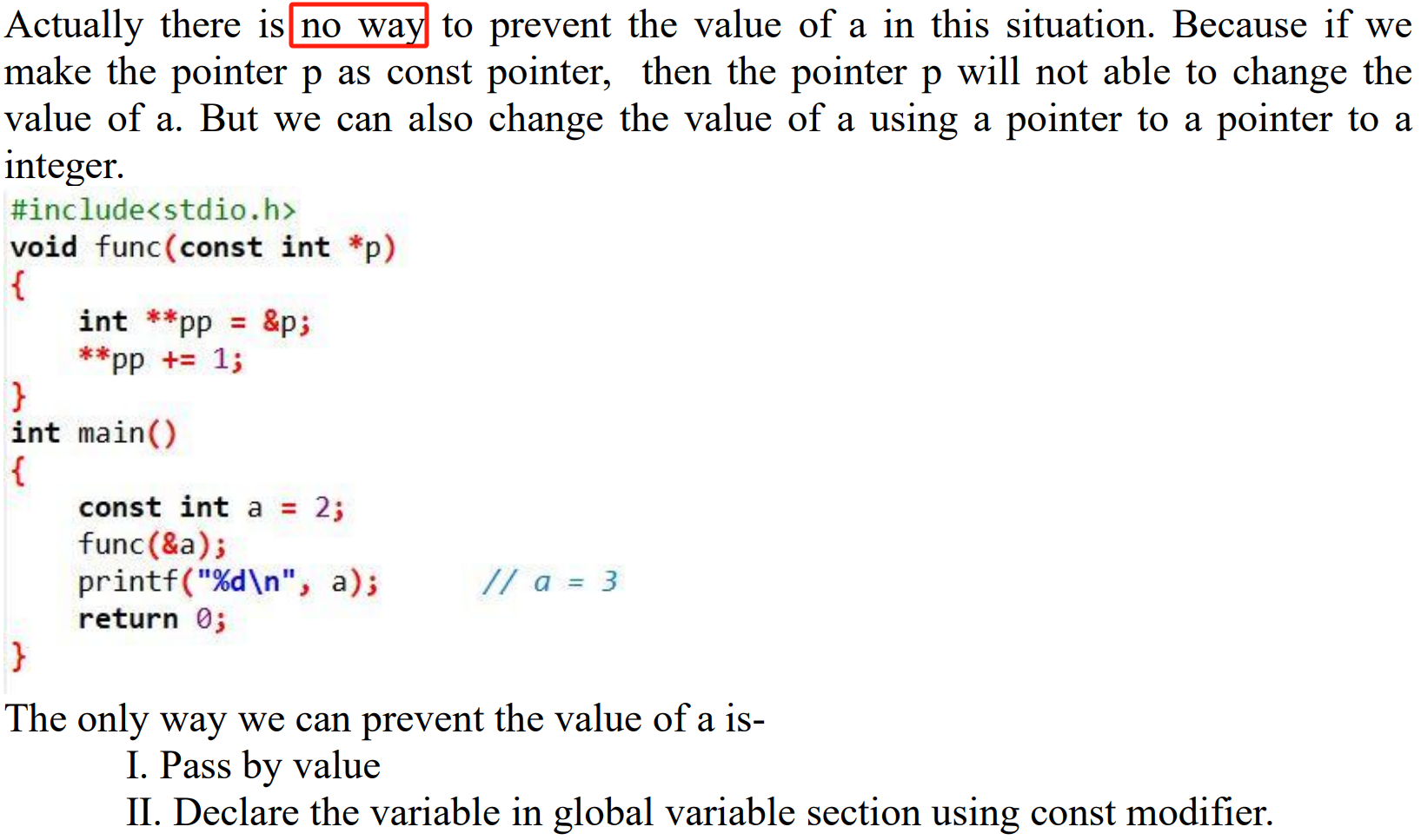
1. How to modify the following code so that the pointer can not be moved?



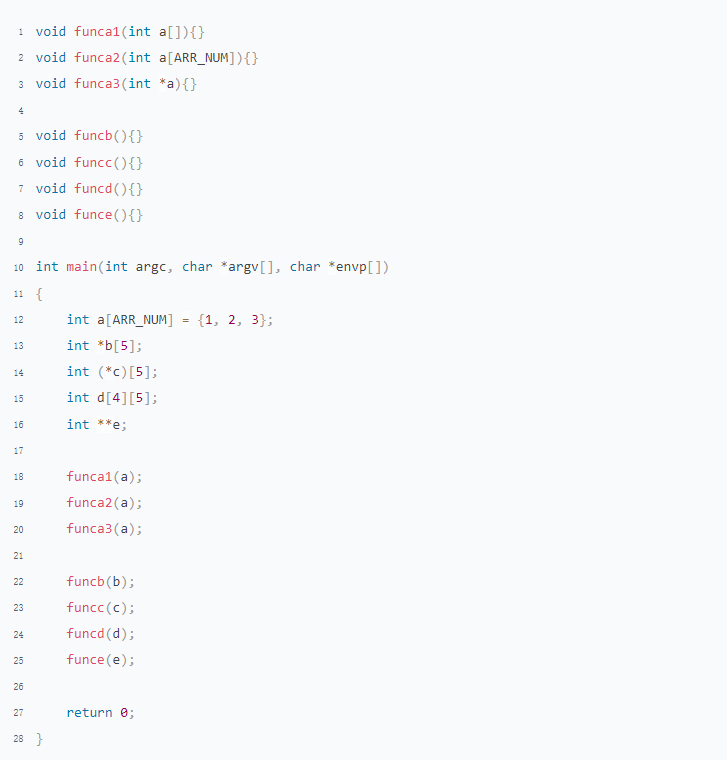
int \* const p = array1;

1. How to prevent the variable a be changed?





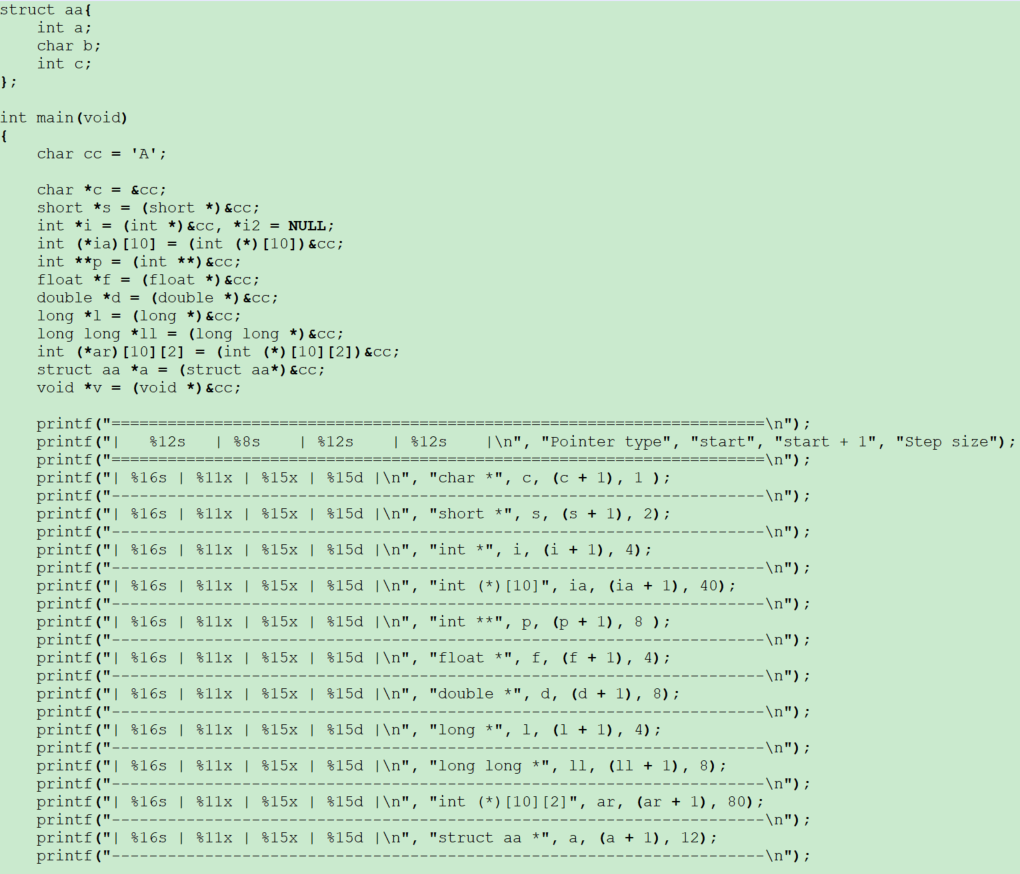
1. Supplement the corresponding formal parameter and add the corresponding formal parameters. Refer to one dimensional array a

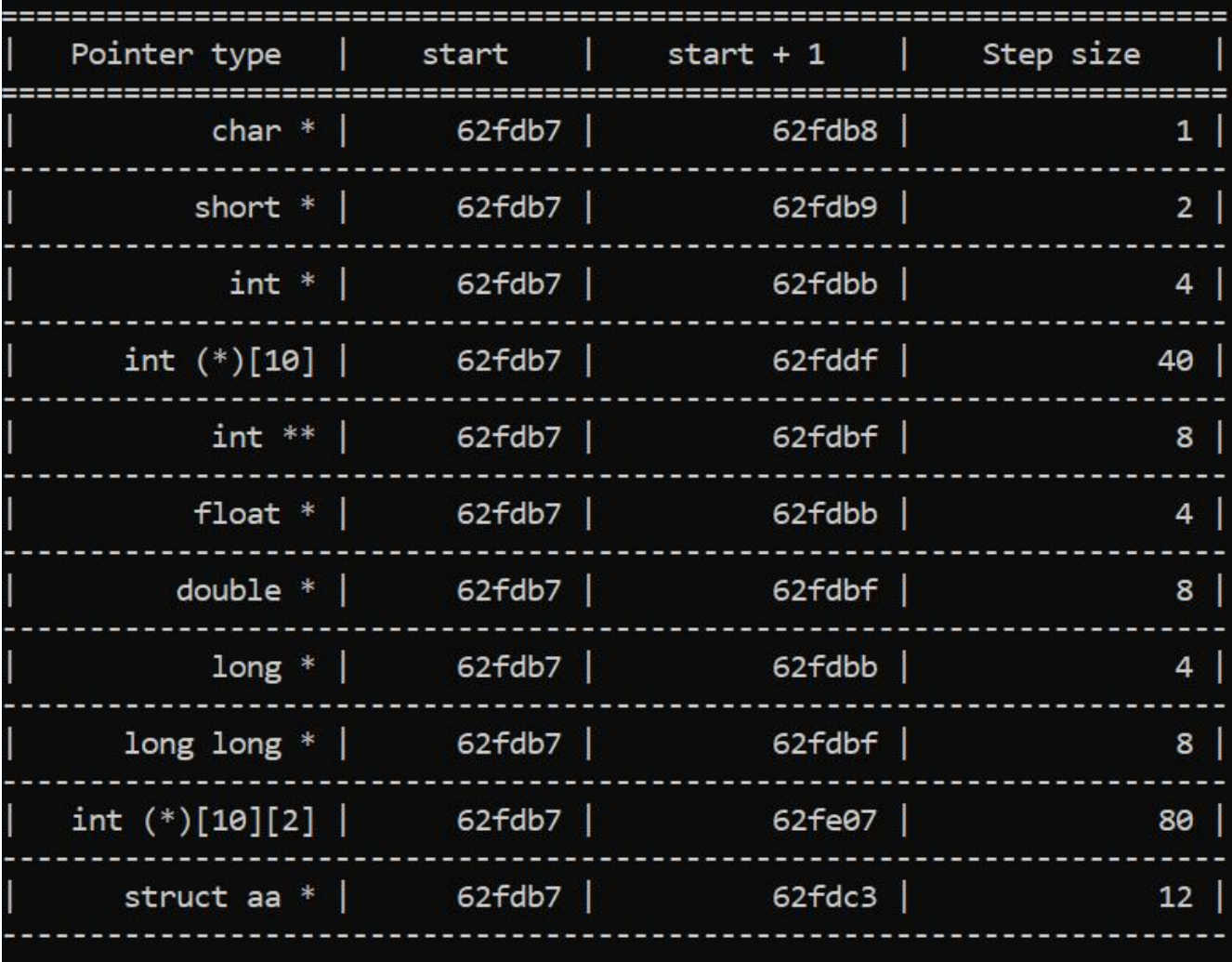


Refer to “The usage of pointer as formal argument.c”

1. Write a program lists out all the types of pointer step size that you know

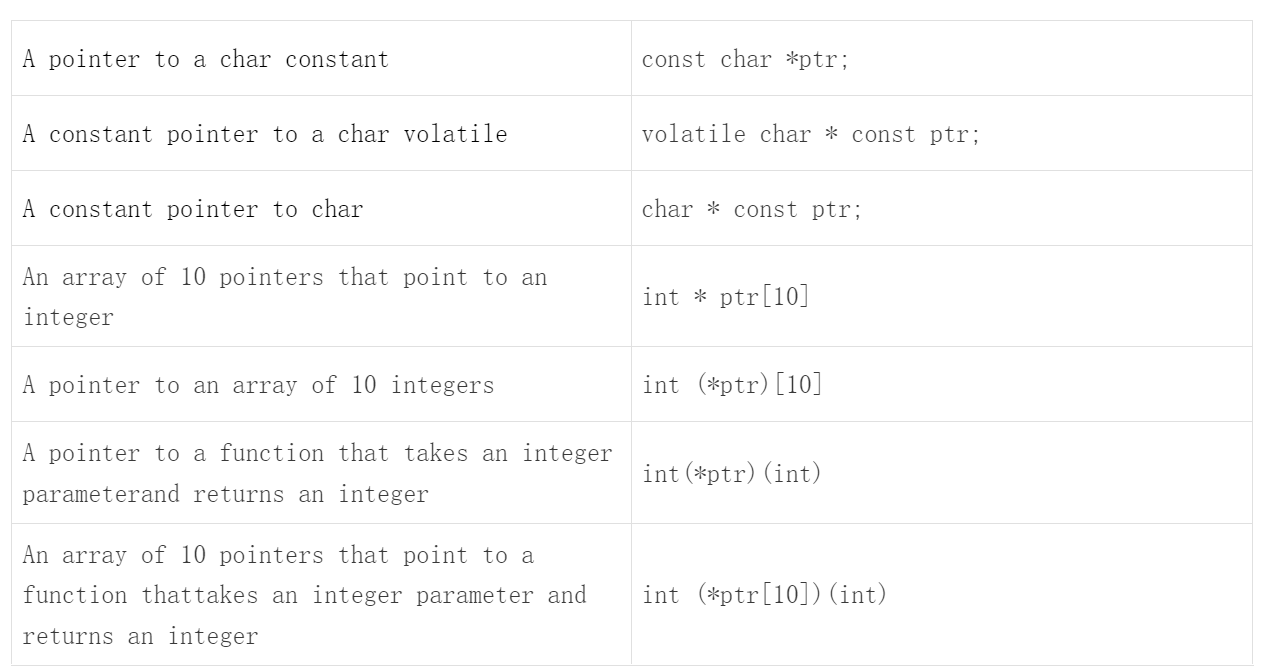
From Dipto





1. Complete the definition according to the description. using variable ptr





1. Use typedef to simplify complicated declarations. Inclusion the steps
   1. int (\*(\*AA)(void \*))[10];
   2. void (\*BB(int, void (\*)(int)))(int);
   3. char (\*(\*CC[3])())[5]

