**PRACTICE – ADVANCED PROGRAMMING**

**Topic: Pointer**

Name: Thái Minh Huy Student ID: 23127379

# Assignment 1

The screen will show following:

* : the address which pointer b stores.
* :3 is the value which pointer b holds in memory.
* :the address of the pointer b.
* 3 is the value of a.
* : the address of variable a.

There is no error in this code.

# Assignment 2

Explain:

* will point to x (x is an integer variable, ip1 is the pointer to int, x has value 100).
* will point to z (z is an integer variable, ip2 is the pointer to int, z has value 50).
* will point to y (y is a float variable, fp is the pointer to float, y has value 20.0).
* will point to ch (ch is a char variable, chp is the pointer to char, ch = ‘Z’).
* pointer will point to x.
* : pointer will point to z.
* (x and z will have value 100).
* z is now 200.
* \*ip2 has value 100, after sum will have 400 (z = 400).
* : y will have value 1.2

Result:

* Print the value of x is 100.
* Print the value of y is 1.2.
* Print the value of z is 400.
* : the address that pointer ip1 holds.
* : 400 is the value which ip1 pointer holds in memory.
* the address of ip1 pointer.
* : the address that pointer ip2 holds.
* : 100 is the value which ip2 pointer holds in memory.
* is the address of ip2 pointer.
* : the address that pointer fp pointer holds.
* : 1.2 is the value which fp pointer holds in memory.
* is the address of fp pointer.
* : the address that pointer chp holds.
* : Z is the value which chp pointer holds in memory.
* is the address of chp pointer.

There are no errors in this code.

# Assignment 3

Explain:

* : Allocate memory for pointer a.
* Allocate memory for pointer b.
* : Assign the value for pointer a.
* : b will now point the same address of a (**Error**) (We need to deallocate the memory first and then assign . Otherwise, the memory that b point to will wander in RAM make computer slower.)
* We change line to “\*b = \*a”. Now \*b has value 2.
* Two lines “cout” print the value memory that a and b pointer point to. The program will print 2 and 2.
* Be sure to use “delete” after we have nothing to do with memory anymore.

# Assignment 4

Explain:

* Assign the value of a (a = 3).
* Initialize pointer p. p points to a.
* Print the value of \*p is 3 (\*p similar to a).
* Allocate the memory for p pointer and assign the value of \*p is 5.
* Print the value of \*p is 5. (p is now pointing the new address).

Error: Not deallocate the memory that pointer p points to. Use “delete” keyword to deallocate the memory after we use “new” keyword.

# Assignment 7

#include <stdio.h>

Int main()

{

Int \*x;

\*x = 100;

}

Answer: A: Invalid Assignment for x.

Explain: x is a pointer but x doesn’t point to anything. ( x is point to a garbage address).

# Assignment 9

Answer: ce (D)

Explain:

* str will point to “peace”.
* s will point to peace.
* ++: this is postflix, s will continue pointing to str[0].
* +3: use pointer arithmetic, s originally points to str[0]. After use arithmetic, s will point to s[3] (“ce”).
* Print “ce” in terminal.

# Assignment 10

Answer: 2, 15, 6, 8, 10. (B)

Explain:

* () function has two arguments: an interger pointer b, size of array (array is basically a const pointer).
* function create a copy of pointer (pass by value). Although it doesn’t change the pointer, memory values which pointer points to will change.
* \*(b + 1) similar to b[1], \*(b + i) is similar to b[i] which means .
* function will go through the loop and change b[1]’s value. After reaching the end of array, it will be which means .

# Assignment 11

Answer: 20, 4, 4 (B)

Explain:

* total bytes of the array. Because arr is an interger variable and has 5 elements .
* : int pointer and it has 4 bytes.
* first element of arr (data type: int), 4 bytes.

# Assignment 12

Answer: 300 (D)

Explain:

* First , str will point to “d”.
* Second str will point to “\n”.
* Printf:
  + : str will point back to “%” (str[0]).
  + is format specifer for interger variable The program will print 300.

# Assignment 13

Answer: The operator used for dereferencing or direction is ”.

Ex: int i = 5, \*ptr = &5;

\*ptr = 10; (i is now 10).

# Assignment 14

string \*x, y;

Anwser: x is a pointer to string, y is a string.

# Assignment 15

Answer: D

Explain:

* Hold the address of the specific object: valid.
* Point one past the end of an object: Valid.
* Zero: Valid (maybe use NULL/nullptr).
* Point to a tye: not a possible state for pointer.

# Assignment 16

Answer: C

Explain:

* i is an interger variable.
* dp is a double pointer.
* A pointer can only point to an object when they have the same data type which means dp can only point to double data type.

# Assignment 17

Answer: B

Explain:

* Init a and b as interger varible and assign value for them (a is 100, b is 200).
* Init two pointers: p points to a, q points to b.
* : p pointer points to b.
* now points to .

# Assignment 18

Answer: D

Explain:

* : Store the address of a, b, c
* print the address of b.
* Return some random number.

# Assignment 19

Answer: A

Explain:

* Init a character array with 20 elements. (arr is a character array).
* In for-loop, assign the ASCII value from 65->74 (A->J) into arr[i] with is from
* When = 10, assign arr[10] to NULL value representing the end of array of characters.
* Print “ABCDEFGHIJ”.

# Assignment 20

Answer: A

Explain:

* Initialize a character pointer (ptr).
* Assign str to “abcdefg”
* : ptr will point to str[0].
* : ptr will point to str[5].
* At this point, ptr will have “fg” string.
* Print “fg” in terminal.

# Assignment 21

Answer: D

Explain:

* A pointer can be initialized with NULL/nullptr.
* Can be initialized with 0.
* Can be initialized with address of an object with the same type.
* All of them are correct.

# Assignment 22

Answer: D

Explain:

* Non-constant pointer to non-constant data: legal.
* Non-constant pointer to constant-data: legal (use when read the data).
* Constant-pointer to non-constant data: legal (don’t want to change the address of pointer points to).
* All of them are legal when write code.

# Assignment 23

Answer: B

Explain: const keyword uses when we don’t want to change the data or modify the value.

# Assignment 24

Answer: C

Explain: the new operator return the address of unallocated blocks in memory.

# Assignment 25

Answer: B

Explain: Referencing (get a value) through a pointer is call indirection.

# Assignment 26

Answer: A

Explain: Use sizeof operator to get the size of the array.

# Assignment 27

Answer: A

Explain: a pointer is a variable that contains the address of a variable.

# Assignment 28

Answer: C

Explain: we can use NULL, nullptr keyword, 0 to initialize a pointer. (These values make pointer point to nothing/garbage memory).

* 3 ways.

# Assignment 29

Answer: C

Explain: Address operator will return the address of variable.

# Assignment 31

Answer: D

Explain:

* Initilize a const interger with value 20.
* Initilize a const pointer point to a const interger, ptr points to .
* Two reasons why the code failed to run.
  + A const pointer can point to one and only address.
  + This pointer points to a const interger so it can’t modify the values
* can only point to I and can’t modify the values of .

# Assignment 32

Answer: B

Explain:

* Inititlize a interger array with 5 elements.
* Initialize a interger pointer name p and point to num (array of interger). (p = num similar to p = &num[0]).
* Assign p[0] = 10.
* Move a pointer to p[1].
* Assign p[1] = 20.
* Move a pointer to p[2].
* Assign p[2] = 30.
* Move a pointer to p[3].
* Assign p[3] = 40.
* Move a pointer to p[4].
* Assign p[4] = 50.
* Use for-loop to print all elements in array.
* Result: 10, 20, 30, 40, 50,

# Assignment 33

Answer: C

Explain:

* Initialize a array with 4 elements (4, 5, 6, 7).
* Initialize a pointer point to the second element of the array.
* Array data structure is technically a pointer so we can use array name as a pointer. (arr[0] similar to \*arr or \*(arr + 0))
* \*arr is the first element of array and its value is 4, 4 + 10 = 14.
* Result: 14.

# Assignment 34

Answer: C

Explain:

* Initialize an interger variable(a), a pointer to int (pa), a reference to interger data type.
* Reference must initialize when declare or we may compile error.
* Fix: int &ra = a;

# Assignment 35

Answer: 2

Explain:

* Initialize an interger array (a).
* Initialize a pointer to int (ptr)
* will point to first element of array.
* \*(ptr + 1) similars to ptr[1] or a[1] which means terminal will print the value of a[1] is 2.

# Assignment 36

Answer: 15

Explain:

* Init an interger variable (a) and a pointer to int (ptr).
* ptr will point to a.
* Use Indirection operator (dereferencing) to get the value of pointer is 5.
* and value of a is 15.

# Assignment 37

Answer: 222

Explain:

* Initialize an interger variable and assign the value for it (int i = 6), a pointer to int (j).
* j points to i.
* \*j similars to I (indirection operator).
* Result: 6 \* 6 \* 6 + 6 = 222.

# Assignment 38

Explain:

* Initialize an interger variable and assign the value for it (x = 20) and two pointers to int.
* y points to x, z points to x because of pointer assignment.
* \*y = \*z = x (Indirection Operator).
* \*y++: \*y = 20, y = 504.
* \*z++: \*z = 20, z = 504.
* x++: x= 21
* Print x = 21, y = 504 (address of x), z = 504 (address of x).

Answer: x = 21, y = 504, z = 504.

# Assignment 39

Answer: x = 10, y = 10, z = 10

Explain:

* Initialize an interger variable and assign value for it (x = 10), a pointer to int (y), a pointer to pointer (z).
* y points to x, z points to y
* Use indirection operator: \*y similars to x, \*z similars to y, \*\*z is similars to x. (\*y =x, \*z = y, \*\*z = x).
* x = \*y = \*\*z = 10.

# Assignment 41

Answer: B

Explain:

* Init a pointer to int, but this pointer does not point to an object, it points to to garbage memory, invalid assignment.
* Result: Invalid Assignment.

# Assignment 42

Answer: C

Explain:

* \*&ptr similars to ptr and the same for &\*ptr.
* Print the address that ptr point to.
* Address and Address

# Assignment 43

Answer: D

Explain:

* First printf line, program will print the value of num variable is 10 and address of num is 2293436.
* Increase num value 1 more digit: 11
* Second printf line will print 11 and address is 2293436.

# Assignment 44

Answer: A

Explain:

* Initialize a int variable i, value of it is 25, init a level 1 pointer j, init a level 2 pointer k.
* j points to I, k points to j (k is the pointer to j while j is the pointer to int).
* printf line will print:
  + k: address k points to j.
  + \*k: address of j points to int.
  + \*\*k: the value of i.
* Result: Address – address – value.

# Assignment 45

Answer: A.

Explain:

* Init a float variable x has value 5.999, two pointers to float (y, z).
* y, z points to x.
* \*(&x) = x, \*y = \*z = x (because y and z point to float).
* Result: 5.999, 5.999, 5.999, 5.999

# Assignment 46

Answer: B

Explain:

* Initialize a array of int with 4 elements, a pointer to int (striker).
* Striker points to the first element of array, track[1] is now 50 (20 + 30).
* First line cout print: “Striker>10 ”. (1).
* Change the value of track[0] = 0 (10 – 10 = 0).
* Move the pointer striker to track[1].
* Second line cout print: “Next@50 ”. (2).
* Move the pointer striker to track[3].
* Third line cout print: “Last@40 ” (3).
* Last: print “Reset To0” (4).

From 1, 2, 3, 4: Final result: “Striker>10 Next@50 Last@40 Reset To0”.

# Assignment 48

Answer: C

Explain:

* Init a constant pointer with I = 20.
* Init a constant pointer to const int and point to i.
* A constant pointer to const int is the pointer to only one object and use to read data.
* Result: Compilation Error.

# Assignment 50

Answer: A.

Explain:

* Init an array to integer with 5 elements, a pointer to int (p).
* P points to the first element of array (x).
* Change the value in array using two pointers method (one at beginning, one at the end of array).

Result: 5 4 3 2 1.