**Tên: Võ Quốc Bình**

**MSSV: 21127233**

**Lớp học phần: Kỹ thuật lập trình 22clc01**

**W02\_30**

1. **Assignment 1:**

**Kết quả in ra:**

Địa chỉ của a

3

Địa chỉ của con trỏ b

3

Địa chỉ của a

**Lý do:**

* 2 biến int a và int\* b
* b là con trỏ trỏ tới địa chỉ a nên nó giữa địa chỉ của a nên khi cout ra b thì sẽ cout ra địa chỉ của a
* \*b sẽ in ra giá trị mà địa chỉ con trỏ b đang giữ nên sẽ in ra 3
* &b sẽ in ra địa chỉ của của con trỏ b
* cout ra a ra 3 và &a sẽ ra địa chỉ của a vì nó là biến lưu trữ bình thường

1. **Assignment 2:**

**Kết quả in ra:**

100

1.2

400

Địa chỉ của z

400

Địa chỉ của con trỏ ip1

Địa chỉ của z

100

Địa chỉ của con trỏ ip2

Địa chỉ của y

1.2

Địa chỉ của con trỏ fp

Địa chỉ của ch

Z

Địa chỉ của con trỏ chp

**Lý do:**

* Giá trị ban đầu của x là 100 và con trỏ ip1 trỏ tới x và lúc sau trỏ tới biến z nên sau khi con trỏ ip1 thay đối giá trị của biến tham chiếu thì chỉ chỉ có biến z bị đổi còn biến y giữ nguyên là 100
* Giá trị ban đầu của y là 20.0 nhưng vì con trỏ của fp trỏ tới y và thay đổi giá trị của biến tham chiếu thành 1.2 nên y = 1.2 và in ra 1.2
* Con trỏ ip2 lưu biến tham chiếu của ip1 ban đầu nên ip2 trỏ tới x và giữ giá trị của biến tham chiếu là 100. Khi ip1 thay đổi giá trị biến tham chiếu thành \*ip2 + 300 tức giá trị biến tham chiếu bị đổi thành 400
* ip1, \*ip1, &ip2 tương ứng với địa chỉ biến tham chiếu z, giá trị của biến tham chiếu z, và địa chỉ con trỏ ip1
* ip2, \*ip2, &ip2 tương ứng với địa chỉ biến tham chiếu x, giá trị của biến tham chiếu x, và địa chỉ con trỏ ip2
* fp, \*fp, &fp tương ứng với địa chỉ biến tham chiếu y, giá trị của biến tham chiếu y, và địa chỉ con trỏ fp
* chp, \*chp, &chp tương ứng với địa chỉ biến tham chiếu ch, giá trị của biến tham chiếu ch, và địa chỉ con trỏ chp

1. **Assignment 3**

**Kết quả in ra:**

(Không có)

**Lý do chương trình bị lỗi vì:**

* b = a nên b sẽ giữ vị trí biến a tham chiếu đến.Tiếp đó a đã xóa địa chỉ biến mà nó đã khởi tạo tron heap mà b lại tiếp tục xóa địa chỉ biến đó nữa nên do đó nó sẽ gây ra lỗi khi xóa
* Không chỉ vậy địa chỉ con trỏ b khởi tạo ban đầu vẫn còn gây ra lãng phí tài nguyên trong ram

1. **Assignment 4**

**Kết quả in ra:**

3

5

**Lý do:**

* Ban đầu con trỏ p trỏ tới biến a nên nó sẽ tham chiếu tới biến a và in ra giá trị của biến a 3
* Sau đó con trỏ p khởi tạo vùng nhớ mới với toán tử new cùng với giá trị ban đầu là 5 nên khi in ra sẽ in giá trị là 5

1. **Assignment 5**

**Kết quả cuối cùng của**

\*p=50

q=8

\*r=8

v=8

\*s=50

**Lý do:**

* Con trỏ p và con trỏ s cùng giữ 1 vị trí nên khi con trỏ s thay đổi giá trị của biến tham chiếu thì giá trị của biến tham chiếu của p đương nhiên cũng thay đổi theo
* q = v mà v bằng 8 nên q cũng bằng 8
* Giá trị của biến tham chiếu của con trỏ r là 8 vì ban đầu r giữ địa chỉ của biến tham chiếu mà con trỏ p giữ ban đầu là biến q nên khi q thay đổi giá trị cuối cùng là 8 thì \*r biến thành 8

1. **Assignment 6**

**Kết quả cuối cùng của**

\*p = 12

\*q = 11

v = 11

nom = địa chỉ của nom[0]

**Lý do:**

* \*p = 12 vì con trỏ p hiện đang trỏ tới vùng nhớ nom[3] (dòng 13) nên \*p lúc đó là giá trị của nom[3]. Mà p--(dòng 15) nghĩa là địa chỉ của vùng nhớ mà p đang trỏ tới giảm xuống 4 byte. Tức p đang trỏ tới vùng nhớ của nom[2] mà nom[2] = 12 nên \*p = 12
* \*q = 11 vì ban đầu q = p (dòng 4) tức vùng nhớ mà con trỏ p đang giữ là vùng nhớ của con trỏ q mà p đang trỏ tới vùng nhớ của v (dòng 2) nên q đang trỏ tới vùng nhớ của v mà v = 11 (dòng 11) nên \*q = 11
* v = 11 (dòng 11)
* nom = địa chỉ của nom[0] vì khởi tạo mảng int ban đầu gồm 5 phần tử nên nom sẽ giữ vị trí của phần tử đầu tiên là nom[0]

1. **Assignment 7**

**Đáp án:**

B) Error: Suspicious pointer conversion

**Lý do:**

Không cấp phát địa chỉ cho con trỏ do đó con trỏ trỏ vào NULL mà ta lại đổi giá trị biến tham chiếu nên bị lỗi

1. **Assignment 8**

**Đáp án:**

E) (lỗi)

**Lý do:**

Con trỏ x ban đầu không được cấp phát vùng nhớ

1. **Assignment 9**

**Đáp án:**

D) ce

**Lý do:**

Ban đầu con trỏ s được trỏ tới vùng nhớ của str. Sau đó khi printf con trỏ s dung toán tứ postfix ++ và +3 nên con trỏ s sẽ trỏ tới vùng nhớ của str[3]

1. **Assignment 10**

**Đáp án:**

B) 2, 15, 6, 8, 10

**Lý do:**

Tại dòng change(a, 5). Trong hàm change chỉ thay đổi biến a[1] (vì b tương đương với a mà \*(b+1) tức a[1]) mà vòng for chạy đến \*(b+1) = \*(b+4) + 5 mà \*(b+4) + 5=15 nên a[1] = 15

1. **Assignment 11**

**Kết quả**

B) 20, 4, 4

**Lý do**

* sizeof(arr) tức kích thước của mảng int. Mảng gồm 5 phần tử mỗi phần tử 4 byte tức 20
* sizeof(\*arr) tức kích thước đầu tiên của mảng là arr[0] bằng 4 byte
* sizeof(arr[0]) là 4 byte

1. **Assignment 12**

**Kết quả**

D) 300

**Lý do**

Ta lại di chuyển con trỏ str\* 2 lần (str++) do đó ta phải điều chỉnh lại bằng cách – 2 để nó trả lại đúng vùng nhơ và in ra 200

1. **Assignment 13**

**Kết quả**

1. \*

**Lý do**

Định nghĩa

1. **Assignment 14**

**Kết quả**

A) x is a pointer to string, y is a string

**Lý do**

Định nghĩa

1. **Assignment 15**

**Kết quả**

D) point to a type

**Lý do**

* Pointer chí có thể giữ vị trí của một object cụ thể
* Pointer gán tham chiếu sẽ thay đổi đối tượng mà tham chiếu
* Pointer có thể bằng 0

1. **Assignment 16**

**Kết quả**

C) int i; double\* dp = &i;

**Lý do**

* Sai vì dp phải gán địa trỉ của biến double chứ không phải int

1. **Assignment 17**

**Kết quả**

B) p now points to b

**Lý do**

* Việc gán tham chiếu sẽ thay đổi đối tượng mà tham chiếu được liên kết nên ở đây p sẽ trỏ tới b thay vì a

**18. Assignment 18**

**Kết quả**

D) it will return some random number

**Lý do**

* Phần tử trong mảng không được là địa chỉ. Nó có thể là địa chỉ của các biến tĩnh hoặc biến ngoài.

**19. Assignment 19**

**Kết quả**

A) ABCDEFGHIJ

**Lý do**

* Mỗi lần giá trị của (arr+i) được gán bằng 65 + i. Trong lần lặp đầu tiên i = 0 và 65 được gán bằng ‘\0’. Vì vậy, nó sẽ in từ A đến J

**20. Assignment 20**

**Kết quả**

A) fg

**Lý do**

* Con trỏ ptr trỏ tới string ‘fg’. Do đó nó in ra ‘fg’

**21. Assignment 21**

**Kết quả**

D) All of the above

**Lý do**

* Định nghĩa

**22. Assignment 22**

**Kết quả**

D) All of the above

**Lý do**

* Cách để pass 1 pointer vô function là tham chiếu

**23. Assignment 23**

**Kết quả**

B) const

**Lý do**

* Toán tử const sẽ không để biến con trỏ bị thay đổi giá trị khi sử dụng

**24. Assignment 24**

**Kết quả:**

C. The new operator

**Lý do**

* Nếu pointer chưa có địa chỉ thì bắt buộc phải cấp phát địa chỉ cho vùng nhớ bằng toán tử new

**25. Assignment 25**

**Kết quả**

B) Indirection

**Lý do**

* Tham chiếu 1 giá trị thông qua pointer gọi là tham chiếu gián tiếp

**26. Assignment 26**

**Kết quả**

A) sizeof

Lý do

* Dungf size of để xác định kích thước của mảng

**27. Assignment 27**

**Kết quả**

A) Pointer contains an address of a variable

**Lý do**

* Pointer giữa địa chỉ của một biến

**28. Assignment 28**

**Kết quả**

C) 3

**Lý do**

* Pointer có thể khởi tạo với 0, Null và địa chỉ của 1 biến

**29. Assignment 29**

**Kết quả**

C) Address operator

**Lý do**

* & tương đương với địa chỉ của biến đó

**30. Assignment 30**

**Kết quả**

D) 129, a

**Lý do**

* Đầu tiên khai báo int a = 32
* Tiếp theo pointer ptr sẽ nhận địa chỉ của a (do ptr=&a)
* Khai báo ch=‘A’
* &cho=ch có nghĩa là biến cho và ch sẽ cũng chung địa chỉ nên bất cứ ch nhận giá trị nào thì cho sẽ là giá trị đó và ngược lại
* cho+=a tức ‘A’+32=‘a’nên ch=‘a’
* \*ptr+=ch tức là 32+‘a’=129 nên a cũng biến thành 129