# 3.1 Assignment 1

Dòng cout << b << endl; -> in ra địa chỉ của biến a (**0x6efe1c**)

Dòng cout << \*b << endl; -> in ra giá trị của biến a (**3**).

Dòng cout << &b << endl; -> in ra địa chỉ của biến b **(0x6efe10**)

Dòng cout << a << endl; -> in ra giá trị của biến a **(3**)

Dòng cout << &a << endl; -> in ra địa chỉ của biến a **(0x6efe1c**)

# 3.2 Assignment 2

int x,z;

float y;

char ch, \*chp;

int \*ip1, \*ip2;

float \*fp;

x = 100;

y = 20.0;

z = 50;

ch = 'Z';

ip1 = &x; // gán địa chỉ của biến x cho ip1

ip2 = &z; // gán địa chỉ của biến z cho ip2

fp = &y; // gán địa chỉ của biến y cho fp

chp = &ch; // gán địa chỉ của biến ch cho chp

ip2 = ip1; // gán giá trị của biến ip1 cho ip2

ip1 = &z; // gán địa chỉ của biến z cho ip1

\*ip1 = \*ip2; // gán giá trị của x cho z

\*ip1 = 200; // gán giá trị của z thành 200

\*ip1 = \*ip2 + 300; // gán giá trị của z thành x + 300

\*fp = 1.2; // gán giá trị của y thành 1.2

cout << x << endl; // in ra giá trị của biến x **(100)**

cout << y << endl; // in ra giá trị của biến y **(1.2)**

cout << z << endl; // in ra giá trị của biến z **(400)**

cout << ip1 << endl; // in ra địa chỉ của biến z **(0x6efe18)**

cout << \*ip1 << endl; // in ra giá trị của biến z **(400)**

cout << &ip1 << endl; // in ra địa chỉ của biến ip1 **(0x6efe00)**

cout << ip2 << endl; // in ra địa chỉ của biến x **(0x6efe1c)**

cout << \*ip2 << endl; // in ra giá trị của biến x **(100)**

cout << &ip2 << endl; // in ra địa chỉ của biến ip2 **(0x6efdf8)**

cout << fp << endl; // in ra địa chỉ của biến y **(0x6efe14)**

cout << \*fp << endl; // in ra giá trị của biến y **(1.2)**

cout << &fp << endl; // in ra địa chỉ của biến fp **(0x6efdf0)**

cout << chp << endl; // in ra địa chỉ của biến ch **(ZÜÖÖ?É☺)**

cout << \*chp << endl; // in ra giá trị của biến ch **(Z)**

cout << &chp << endl; // in ra địa chỉ của biến chp **(0x6efe08)**

# 3.3 Assignment 3

int \*a = new int;

int \*b = new int;

\*a = 2;

b = a; **// b và a trỏ vào cùng một ô địa chỉ**

cout << \*a << endl; **// in ra 2**

cout << \*b << endl; **// in ra 2**

delete a;

delete b; **// vì ô địa chỉ b trỏ vào cũng là địa chỉ a trỏ vào đã được xóa ở dòng trên nên sẽ xảy ra lỗi ở đây.**

# 3.4 Assignment 4

int a = 3;

int \*p = &a; **// gán địa chỉ biến a cho p**

cout << \*p << endl; **// in ra giá trị của biến a ( = 3 )**

p = new int(5); **// tạo một ô địa chỉ mới có giá trị = 5 và gán địa chỉ cho p**

cout << \*p << endl;

**//Đoạn code này quên xóa ô địa chỉ đã tạo cho p nên sẽ xảy ra tràn bộ nhớ.**

# 3.5 Assignment 5

1. int v = 8, \*r, \*s; // v = 8

2. int \*p;

3. int q = 100; // q = 100

4. p = &q; // \*p = 100;

5. r = p; // \*r = 100

6. \*p = 20; //\*p=20 q = 20, \*r = 20

7. p = new int;

8. \*r = 30; //q = 30, \*r = 30

9. q = v; // q = 8, \*r = 8

10. s = p; //\*s = \*p

11. \*s = 50; // \*s = 50, \*p = 50

The last value of:

\*p = 50

q = 8

\*r = 8

v = 8

\*s = 50

# 3.6 Assignment 6

1. int \*p ,\*q , v , nom[5];

2. p = &v;

3. \*p = 12; // \*p = 12, v = 12

4. q = p; // \*q = 12

5. nom[0] = \*q; // nom = [12,,,,]

6. p = nom; // \*p = 12

7. p++; // \*p = nom[1]

8. nom[2] = 12; // nom[2] = 12

9. \*p = 13; nom[1] = 13

10. \*q = 10; //\*q = 10, v = 10

11. v = 11;// \*q = 11, v = 11

12. \*(p+3) = 16; //nom[4] = 16

13. p = &nom[3];

14. \*p = 10; // nom[3] = 10

15. p--; //\*p = nom[2]

The last value of:

* \*p = 12
* \*q = 11
* v = 11
* nom[0] = 12
* nom[1] = 13
* nom[2] = 12
* nom[3] = 10
* nom[4] = 16

# 3.7 Assignment 7

Graphical user interface

Description automatically generatedB. Error: suspicious pointer conversion at line “\*x=100”.

# Graphical user interface Description automatically generated with low confidence3.9 Assignment 9

D. ce

# 3.10 Assignment 10

Graphical user interface, text, application

Description automatically generated

B. 2, 15, 6, 8, 10

# 3.11 Assignment 11

Graphical user interface, text

Description automatically generatedB. 20, 4, 4

# 3.12 Assignment 12

D. 300

# 3.13 Assignment 13

Text

Description automatically generated with medium confidencea) \*

# 3.14 Assignment 14

Graphical user interface, text, application

Description automatically generateda) x is a pointer to a string, y is a string

# 3.15 Assignment 15

Text

Description automatically generatedd) point to a type

# 3.16 Assignment 16

Text

Description automatically generatedc) int i; double\* dp = &i;

# 3.17 Assignment 17

Graphical user interface, application, Teams

Description automatically generatedb) p now points to b

# 3.18 Assignment 18

Graphical user interface, text, application

Description automatically generatedd) it will return some random number

# Graphical user interface, text, application Description automatically generated3.19 Assignment 19

a) ABCDEFGHIJ

# 3.20 Assignment 20

Graphical user interface, text, application

Description automatically generated

1. fg

# Graphical user interface, text, application, chat or text message Description automatically generated3.21 Assignment 21

D. All of them

# Text Description automatically generated3.22 Assignment 22

D. All of them

# 3.23 Assignment 23

Text

Description automatically generatedB. const

# 3.24 Assignment 24

Text

Description automatically generated with medium confidenceC. The new operator

# Graphical user interface, text, application, chat or text message Description automatically generated3.25 Assignment 25

B. Indirection

# 3.26 Assignment 26

Text

Description automatically generated with medium confidenceA. sizeof

# 3.27 Assignment 27

Graphical user interface, text, application, chat or text message

Description automatically generatedA. Pointer contains an address of a variable

# 3.28 Assignment 28

Text

Description automatically generatedC. 3

# 3.29 Assignment 29

Graphical user interface, text, application

Description automatically generatedC. Address operator

# 3.30 Assignment 30

Text

Description automatically generatedC. 129, a

# Text Description automatically generated3.31 Assignment 31

D. Compile error