

Name: Thái Minh Huy:

FP: 23127379

Practice 2.4

Given the following program.

```
#include <stdio.h>
```

```
int main() {
```

```
    int x = 1023;
```

```
    char *p = (char *) &x;
```

```
    printf("%d %d %d %d\n", p[0], p[1], p[2], p[3]);
```

```
}
```

The result on the screen is: -1 3 0 0

Explain

- First, x is declared with value 1023. Because x is an integer variable, it will store the value of x into 4 bytes consecutive cells (One cell for one byte and 1 byte is 8 bit), below is how ~~the~~ program stored.

1023 | 256

255 | 3

3 | 0

$\Rightarrow x \rightarrow$

255	3	0	0
-----	---	---	---

(256 is number of value that one byte can store)

x can be written as:

1111 1111 0000 0011 0000 0000 0000 0000

- Declare p is a pointer character and assigned p with $\text{const int } * \rightarrow \text{char } *$

- $\&x$ get the ~~first~~ address of x , point to the first cell

- $(\text{char } *) \&x$ make a pointer character.

- If we don't cast $\&x$ into character pointer, when we use pointer arithmetic to move the pointer, it will the program will read four bytes and they can point to a other byte address

- Finally, the program will print each cell of x .

Because of format specifier "%d", it will print ~~using~~ stored in memory by one byte:

1111 1111 \rightarrow -1
 0000 0011 \rightarrow 3
 0000 0000 \rightarrow 0
 0000 0000 \rightarrow 0

\rightarrow The result is -1 3 0 0

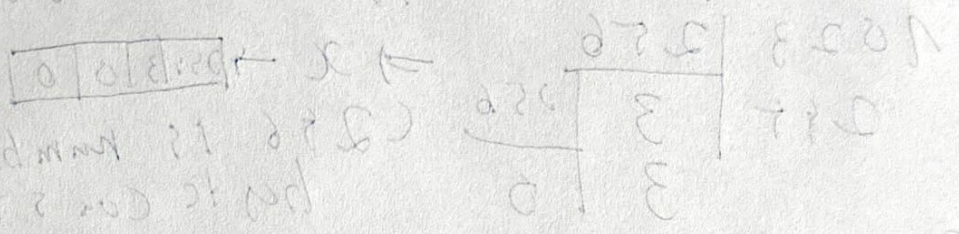
Name: Tawfik Hady
 ID: 2023012301
 Section: 2.1

Given the following program:
 #include <stdio.h>
 int main()
 {
 int x = 10000;
 char *p = (char *) 0;
 printf("x: %d, p: %s", x, p);
 }

Print "x: %d, p: %s", p[0], p[1], p[2], p[3], p[4].
 char *p = (char *) 0;

The result on the screen is: -1 3 0 0
 Explain

- First, x is declared with value 10000. Because x is an integer variable, it will store the value of x into a byte containing cell (0) as cells for the byte are 1 byte (8 bit), below is how the program stores



x can be written as:

1111 1111 0000 0000 0000 0000 0000 0000
 - Because p is a character array and assigned p with '\0' (0) as char.
 - If we get the first address of x, point to the first cell (char) it takes a pointer character.
 - If we don't cast it into character pointer, when we use pointer arithmetic to move the pointer, it will be wrong. The program will read from other cells and that can cause a crash.

- Finally, the program will print some error.
 Because of format specifier "%s", it will print a null character.