Name: Thái Minh Hưng:

ID: 23127379

Pratice 2.4

Given the following program.

```
#include <stdio.h>
int main(){
    int x = ~~2023~~ 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 store the value of x into 4 bytes consecutive cells (One cells for one byte and 1 byte is 8 bit), below is how ~~the~~ program stored.

```
1023 | 256
 255 | 3    | 256
       3    | 0
```

→ x → | 255 | 3 | 0 | 0 |

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

x can be written as:

11111111 00000011 00000000 00000000

- Declare p is ~~a~~ pointer Character and assigned p with cast int* → char*
- &x get ~~the first~~ address of x, point to the first cell
- (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 garbage address
- Finally, the program will print each cell of x. Because of format specifier "%d", It will print ~~unsig~~ signed integer by one byte:

1111 1111 → -1

0000 0011 → 3

0000 0000 → 0

0000 0000 → 0

⇒ The result is -1 3 0 0