

HW2

1. Kết quả biên dịch đoạn code trong file overflow_rounded_off.c

[illegible]

2. Nhận xét kết quả của 2 đoạn code và giải thích vì sao dòng 2,3 của mỗi đoạn có kết quả khác nhau

```
printf("%u\n", UINT_MAX);
printf("%u\n", UINT_MAX + 1);
printf("%u\n", UINT_MAX+2);
```

```
printf("%d\n", INT_MAX);
printf("%d\n", INT_MAX+1);
printf("%d\n", INT_MAX+2);
```

<ul style="list-style-type: none"> - In ra giá trị lớn nhất của unsigned int - In ra giá trị của UINT_MAX + 1 theo kiểu unsigned int - In ra giá trị của UINT_MAX + 2 theo kiểu unsigned int 	<ul style="list-style-type: none"> - In ra giá trị lớn nhất của int - In ra giá trị của INT_MAX + 1 theo kiểu signed int - In ra giá trị của INT_MAX + 2 theo kiểu signed int
%u: unsigned int	%d: signed int
UINT_MAX: định nghĩa giá trị lớn nhất cho một unsigned int	INT_MAX: định nghĩa giá trị lớn nhất cho một int
Số nguyên không dấu có dạng số học mô-đun. Tăng số nguyên không dấu lớn nhất lên 1 thì kết quả là 0. Tăng lên 2 thì thu được 1.	INT_MAX + 1, INT_MAX + 2 thao tác này gọi hành vi không xác định. Tràn số nguyên có dấu là hành vi không xác định trong C. Nó có thể dẫn đến INT_MIN hoặc chương trình có thể bị lỗi.

3. Vì sao $0.1+0.1+0.1$ không bằng 0.3 ?

- [illegible]

4. Vì sao 1.3×0.25 không bằng 0.325?

- [illegible]

5. Hiệu chỉnh đoạn code trên để in ra giá trị của phép tính float $z = 1/3$?

```
float z = 1.0 / 3;
```

- Nếu lấy “1 / 3” thì kiểu int chia kiểu int kết quả ra kiểu int là “0”.
- Điều chỉnh về “1.0 / 3” thì kết quả ra kiểu float.

