

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

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

 In ra giá trị lớn nhất của usigned 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 	 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: usigned 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?
 - Thực tế là máy tính không tính toán ở dạng Thập phân mà ở dạng Nhị phân. Số hữu tỉ trong hầu hết các ngôn ngữ lập trình được biểu diễn dưới dạng số chấm động, chúng là một con số gần đúng của số tương đương trong Hệ thập phân.

 - Vì vậy với một máy tính, số hữu tỷ như "0.1" là một số gần đúng của số gốc trong Hệ thập phân, "0.1 + 0.1 +0.1" sẽ không có tổng bằng "0.3" mà là một số cực kỳ gần với nó.

4. Vì sao 1.3*0.25 không bằng 0.325?

- Thực tế là máy tính không tính toán ở dạng Thập phân mà ở dạng Nhị phân. Số hữu tỉ trong hầu hết các ngôn ngữ lập trình được biểu diễn dưới dạng số chấm động, chúng là một con số gần đúng của số tương đương trong Hệ thập phân.

"1.3" trở thành

"1.29999995231628420000000000000000000000000000000"

- Vì vậy với phép tính "1.3*0.25", máy tính cho ra kết quả gần đúng với nó chứ không bằng "0.325"
- 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.

```
#include <stdio.h>
  #include <math.h>
  #include <limits.h>
 void main()
- (
     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);
      float x = 0.1 + 0.1 + 0.1;
     printf("%.50f",x);
      float y = 1.3 * 0.25;
     printf("\n%.50f",y);
      float z = 1.0 / 3;
     printf("\n%.50f",z);
     printf("\n");
      return;
```