

Answers to 5 values of x

1) x=6, Expected answer: 3

rax	0x6 6	
rbx	0x6 6	
rcx	0x6 6	
rdx	0x0 0	
rsi	0x0 0	
rdi	0x3 3	
rbp	0x0 0x0	
rsp	0x7fffffffdf90	0x7fffffffdf90
r8	0x6 6	
r9	0x0 0	
r10	0x0 0	
r11	0x3 3	
r12	0x0 0	
r13	0x0 0	
r14	0x0 0	
r15	0x0 0	

Value in r11: 3

2) x=10, Expected answer: 5

rax	0xa 10	
rbx	0xa 10	
rcx	0x78	120
rdx	0x0 0	
rsi	0x0 0	
rdi	0x5 5	
rbp	0x0 0x0	
rsp	0x7fffffffdf90	0x7fffffffdf90
r8	0xa 10	
r9	0x0 0	
r10	0x0 0	
r11	0x5 5	
r12	0x0 0	
r13	0x0 0	
r14	0x0 0	
r15	0x0 0	

Value in r11: 5

3) x=19, Expected answer: 19

rax	0x13	19	
rbx	0x13	19	
rcx	0x1b02b9306890000		121645100408832000
rdx	0x0 0		
rsi	0x0 0		
rdi	0x13	19	
rbp	0x0 0x0		
rsp	0x7ffffffdf90		0x7ffffffdf90
r8	0x13	19	
r9	0x0 0		
r10	0x0 0		
r11	0x13	19	
r12	0x0 0		
r13	0x0 0		
r14	0x0 0		
r15	0x0 0		

Value in r11: 19

4) x=1, Expected answer: 1

rax	0x1 1		
rbx	0x1 1		
rcx	0x1 1		
rdx	0x0 0		
rsi	0x0 0		
rdi	0x1 1		
rbp	0x0 0x0		
rsp	0x7ffffffdf90		0x7ffffffdf90
r8	0x1 1		
r9	0x0 0		
r10	0x0 0		
r11	0x1 1		
r12	0x0 0		
r13	0x0 0		
r14	0x0 0		
r15	0x0 0		

Value in r11:1

5) x=20, Expected answer: 5

rax	0x14	20
rbx	0x14	20
rcx	0x78	120
rdx	0x0 0	
rsi	0x0 0	
rdi	0x5 5	
rbp	0x0 0x0	
rsp	0x7fffffffdf90	0x7fffffffdf90
r8	0x14	20
r9	0x0 0	
r10	0x0 0	
r11	0x5 5	
r12	0x0 0	
r13	0x0 0	
r14	0x0 0	
r15	0x0 0	

Value in r11:5

To find out the first number for which the factorial overflowed, I checked the value of r11 for prime numbers, and the first mismatch between the value of x and the value of the answer would show the first inconsistent answer.

This works as for prime numbers the least natural number whose factorial would get no remainder on division by it is the same number itself.

For 64 bit

The program works fine for the first 22 Natural Numbers (1-22).

For x=23, the factorial overflows and the code fails to generate the correct answer which would have been 23 in this case, but rather shows the answer as 39 which is incorrect.

After finding out that 23 is the first natural number for which the code didn't work, I tried 22, 21 and 20 which worked as they didn't have to compute the factorial up to that point as 22 has 11!, 21 has 7! and 20 has 5!. 19 (prime number) gave correct answer 19, hence it can be deduced that the first natural number for which the code fails is 23

For 32 bit

Overflow starts at 13, answer shown is 29 while the correct answer should be 13 itself

For 16 bit

Overflow starts at 11, (works for 7,8,9,10)

Code runs for 10 as it stops at 5!, otherwise even 10! would have caused overflow in the program as $2^{16} < 10!$

For 8 bit

Overflow starts at 7