一．Question 1:  **蓝色的字体为修改的地方**

Comment1:保存A函数要使用的callee-saved registers的数值

Comment2:%r11是caller-saved register,跳过他是因为暂时不会修改其数值，在调用C函数之前在保存

Comment3:对齐栈顶，分配三个空的空间以供后续的local variables使用

Comment4:保存caller-saved register %r11的数值，防止C函数改变其原始数值,同时使其作为调用C函数之后返回值存储的寄存器(return address)

Question2：

|  |  |  |  |
| --- | --- | --- | --- |
| Var | Location | Var | Location |
| a0 | %rbx | A4 | %r12 |
| A1 | %r15 | A5 | %r11 |
| A2 | %r14 | A6 | (%rsp) |
| A3 | %r13 | A7 | 8(%rsp) |

Question3:

|  |  |  |  |
| --- | --- | --- | --- |
| Var | Location | Var | Location |
| a0 | %rdi | A4 | %r8 |
| A1 | %rsi | A5 | %r9 |
| A2 | %rdx | A6 | 8(%rsp) |
| A3 | %rcx | A7 | 16(%rsp) |

Question4:

sub $16,%rsp

movq %rbx,%rdi

Movq %r15,%rsi

Movq %r14,%rdx

Movq %r13,%rcx

Movq %r12,%r8

Movq %r11,%r9

Movq (%rsp),%rbx

Leaq 8(%rsp),%r11

Pushq %r11

Pushq %rbx

Lea 40(%rsp),8(%rsp) 改为

Movb 32(%rsp),(%rsp)

Question5:

值应该是 %rsp + 40,因为C函数刚开始时%rsp + 8地址存储的是a7的地址(&a7),且地址为8bytes

Question6：

Push %r12

Movl $1 , %eax

Cmpq $1 , %rdi

Jle .L35

Movq %rdi,%r12

Leaq -1(%rdi) , %rdi

Call D

Imulq %r12 , %rax

Pop %r12

.L35 : ret

二．

if(x == 0) return 0;

Else return x \* foo(x - 1);