

# 第三章课后习题

1180300220-崔涵

## 3. 59

```
1. // *dest in %rdi, x in %rsi, y in %rdx
2. // x*y = (2^64 * x_63 + x_0) * (2^64 * y_63 + y_0)
3.           = 2^64 * x_63 * 2^64 * y_63 (溢出) + x_0 * y_63 * 2^64 + x_63 * y_0 * 2^64 + x_0 * y_0
4. store_prod:
5.     movq    %rdx, %rax //将 y 存入%rax
6.     cqto    //将 y 的符号位扩展至 rdx 中
7.     movq    %rsi, %rcx //将 x 存入%rcx
8.     sarq    $63, %rcx //将 x 的符号位拓展, 并存入%rcx 中
9.     imulq   %rax, %rcx // 计算 y_0 * x_63, 存入%rcx 中
10.    imulq   %rsi, %rdx // 计算 x_0 * y_63, 存入%rdx 中
11.    addq    %rdx, %rcx // 计算 x_0 * y_63 + y_0 * x_63, 存入%rcx 中
12.    mulq    %rsi      // 计算 x_0 * y_0, 结果的低位存入%rax
13.    addq    %rcx, %rdx // 计算 x_0 * y_63 + y_0 * x_63 + x_0 * y_0, 得到最终结果的高 64 位
14.    movq    %rax, (%rdi) //将最终结果的低 64 位存入%rdi 的位置
15.    movq    %rdx, 8(%rdi) //将最终结果的高 64 位存入%rdi+8 的位置
16.    ret
```

## 3. 63

```
1. long switch_prob(long x, long n){
2.     long result = x;
3.     switch(n){
4.         case 60:
5.             result = 8 * x;
6.             break;
7.         case 61:
8.             break;
9.         case 62:
10.            result = 8 * x;
11.            break;
12.         case 63:
13.            result = x >>= 3;
14.            break;
15.         case 64:
```

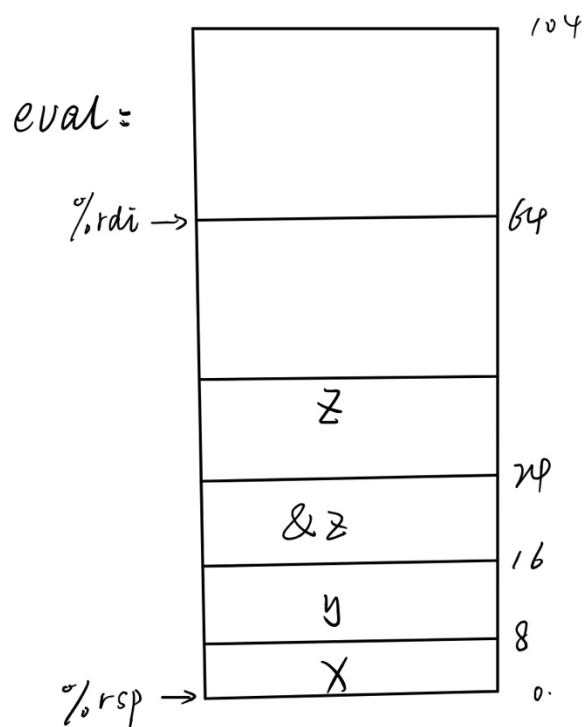
```

16.         result = x;
17.         result <=<= 4;
18.         result -= x;
19.         x = result;
20.     case 65:
21.         x *= x;
22.     default :
23.         result = 75 + x;
24.
25. }
26. return result;
27. }

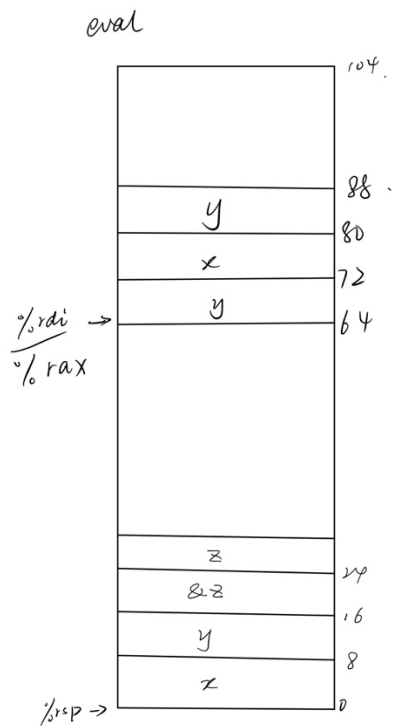
```

### 3.67

A



- B. 传递了 64 (%rsp)
- C. 利用 %rsp+偏移量的方式。
- D. 从 %rdi 出发，按照偏移量保存
- E.



F. 结构体的传递和返回，以指针形式进行，这个指针是结构内容的存储位置，也是此结构作为返回值时的返回值。

### 3.71

```

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <assert.h>
4.
5. void good_echo(){
6.     char tmp[10];
7.     char *a;
8.     while(1){
9.         a = NULL;
10.        a = fgets(tmp,10,stdin);
11.        if(a == NULL){
12.            break;
13.        }
14.        printf("%s",a);
15.        //printf("\n");
16.    }
17.    return;
18. }
19.
20. int main(){
21.     good_echo();

```

```
22.     return 0;
23. }
```

### 3.75

- A. 对于第  $n$  个复数，`%xmmi` 存放其实部，其中  $i = 2n-2$ ；`%xmmj` 存放其虚部， $j = i+1$ 。  
B. `%xmm0` 存结果实部，`%xmm1` 存结果虚部。