

Homework 6

1. Heterogeneous DS

For each of the following structure declarations, determine the offset of each field, the total size of the structure, and its alignment requirement in bytes under x86-64.

- A. `struct P1 { long i; char c; short s; int j;};`
- B. `struct P2 { char c[2]; long l; short s;};`
- C. `struct P3 { struct P1 a[2]; struct P2 *p; char c;};`
- D. `struct P4 { char c[5]; short s[3];}`
- D. `union P5 { char c[5]; short s[3];};`

Fill in the table that follows for each of the struct given except those marked "--"

	Offset of each field				Total size	Alignment
A						
B				--		
C				--		
D			--	--		
E			--	--		

2. Array && Pointers

Answer following questions and explain why. Assume we use x86-64 machine

1. Is the value of `&(a[1])` equals to value of `(b+1)`?

```
int a[2]; char *b = a;
```

2. Is the value of `&(a[1])` equals to value of `(b+1)`?

```
int a[2]; char **b = a;
```

3. Is the value of `&(a[1])` equals to value of `(b+1)`?

```
int *a[2]; char **b = a;
```

4. What is a?

```
int* (*a[3])(int *,int)
```

3. Stack Overflow

Suppose we have the following function 'verify' to perform verify process in a little endian machine.

```
int verify() {  
    char password[8];  
    gets(password);  
    return check_match_in_database(password);  
}
```

Here is a part of the function's assembly.

```
pushq %rbp  
movq %rsp, %rbp  
subq $32, %rsp  
leal -12(%rbp), %rax  
movq %rax, rdi  
call _gets
```

In the normal process, if the password is ok, the function 'verify_ok' will be called to indicate verify success. We've already known that the address of 'verify_ok' is 0x004014ac. Can you construct an input to make the function 'verify_ok' be called after 'verify returns? You just need to specify the key bytes and their positions rather than the complete input