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	i:0	c:8	s:10	j:12	16	8
B	c:0	l:8	s:16	--	24	8
C	a:0	p:32	c:40	--	48	8
D	c:0	s:6	--	--	12	2
E	c:0	s:0	--	--	6	2

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;
```

No, sizeof(int) is 4, sizeof(char) is 1.

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

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

No, sizeof(int) is 4, sizeof(char *) is 8.

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

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

Yes, both a and b are pointer to pointers.

4. What is a?

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

An Array with 3 elements points to a function with two parameters (int * and int) returning int pointer.

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

Just to let:
username[20] = 0xac;
username[21] = 0x14;
username[22] = 0x40;
username[23] = 0x00;
username[24] = 0x00;
username[25] = 0x00;
username[26] = 0x00;
username[27] = 0x00;