17376283 李智健 lab04a

Q1

1. 12, 12
2. 16, 16
3. 76, 76
4. 20, 20
5. 56, 32

Q2

```
typedef struct node {
 double x;
 unsigned short y;
 struct node *next;
 struct node *prev;
} node_t;
node_t n;
void func() {
 node_t *m;
 m = n->next;
 m->y /= 16;
 return;
}
```

Q3

`M = 5, N = 7`

`&array2[j][i] = 4 * (5 * j + i)` 所以列维数 M 等于 5

`&array1[i][j] = 4 * (7 * i + j)` 所以列维数 N 等于 7

Q4

调用之前	
`08 04 86 43`	return address
`bf ff fc 94`	old %ebp
2	%edi
3	%esi
1	%ebx
	buf[4-7]
	buf[0-3]
...	useless stack
-0x14(%ebp)	parameter of gets

调用之后发生栈溢出	
08 04 86 00	"\0"
51 50 49 48	"0123"
57 56 55 54	"6789"
53 52 51 50	"2345"
49 48 57 56	"8901"
55 54 53 52	"4567"
51 50 49 48	"0123"

由于 `\0` ，返回地址变为 `0x08048600`

Q5

1. ```
   buf[0] = 0x64636261
   buf[1] = 0x68676665
   buf[2] = 0x08046900
   ```

2. ```
   %ebp = buf[1] = 0x68676665
   ```

   在调用 `foo` 前栈帧为：

`0x08048523` return address of `callfoo`
old %ebp
&buf[0]

调用后发生缓冲区溢出，`old %ebp` 部分被 `"efgh"` 覆盖，由小端法储存，故为 `68 67 66 65`

3. `%eip = buf[1] = 0x08046900`

   与第二问一样，返回地址的低 16 位被 `'i', '\0'` 覆盖，故低 16 位变成 `6900`

Q6

```
void proc (union ele *up) {
  up->e2.next->e1.y = *(up->e2.next->e1.p) - up->e2.x;
}
```