Practice 1

（x86-64） Consider the following data structure declaration:

```
struct ms_pacman{
 0  short wire;  2
 4  int resistor; 8
 8  union transistor{  24
       0 char bjt; 1
       0 int *mosfet; 8
       0 long vacuum_tube[2]; 16
    } transistor;
24  struct ms_pacman *connector; 32
};
```

Below are given four C functions and four x86-64 code blocks.

下面给出 4 个 c 函数和 4 段 x86-64 代码。

```
char* inky(struct ms_pacman *ptr){
  return &(ptr->transistor.bjt);
}

long blinky(struct ms_pacman *ptr){
  return ptr->connector->
         transistor.vacuum_tube[1];
}

int pinky(struct ms_pacman *ptr){
  return ptr->resistor;
}

int clyde(struct ms_pacman *ptr){
  return ptr->transistor.mosfet;
}
```

A	mov 0x8(%rdi), %rax retq
B	lea 0x8(%rdi), %rax retq
C	mov 0x4(%rdi), %eax retq

D	mov 0x18(%rdi),%rax
	mov 0x10(%rax),%rax
	retq

（x86-64） Consider the following data structure declaration:

```
struct ms_pacman{
  0  short wire;  2
  4  int resistor;  8
  8  union transistor{  24
        0 char bjt;  1
        0 int *mosfet;  8
        0 long vacuum_tube[2];  16
     } transistor;
 24  struct ms_pacman *connector;  32
};
```

Below are given four C functions and four x86-64 code blocks.

下面给出 4 个 c 函数和 4 段 x86-64 代码。

```
char* inky(struct ms_pacman *ptr){
  return &(ptr->transistor.bjt);
}

long blinky(struct ms_pacman *ptr){
  return ptr->connector->
         transistor.vacuum_tube[1];
}

int pinky(struct ms_pacman *ptr){
  return ptr->resistor;
}

int clyde(struct ms_pacman *ptr){
  return ptr->transistor.mosfet;
}
```

A	mov 0x8(%rdi), %rax retq
B	lea 0x8(%rdi), %rax retq
C	mov 0x4(%rdi), %eax retq

D	mov 0x18(%rdi),%rax
	mov 0x10(%rax),%rax
	retq

（x86-64） Consider the following data structure declaration:

```
struct ms_pacman{
 0  short wire;  2
 4  int resistor;  8
 8  union transistor{  24
        0 char bjt;  1
        0 int *mosfet;  8
        0 long vacuum_tube[2];  16
    } transistor;
24  struct ms_pacman *connector;  32
};
```

Below are given four C functions and four x86-64 code blocks.

下面给出 4 个 c 函数和 4 段 x86-64 代码。

```
char* inky(struct ms_pacman *ptr){
  return &(ptr->transistor.bjt);
}

long blinky(struct ms_pacman *ptr){
  return ptr->connector->
         transistor.vacuum_tube[1];
}

int pinky(struct ms_pacman *ptr){
  return ptr->resistor;
}

int clyde(struct ms_pacman *ptr){
  return ptr->transistor.mosfet;
}
```

A	`mov 0x8(%rdi), %rax` `retq`
B	`lea 0x8(%rdi), %rax` `retq`
C	`mov 0x4(%rdi), %eax` `retq`

D	`mov 0x18(%rdi),%rax`
	`mov 0x10(%rax),%rax`
	`retq`

（x86-64） Consider the following data structure declaration:

```
struct ms_pacman{
  0  short wire;  2
  4  int resistor;  8
  8  union transistor{  24
        0 char bjt;  1
        0 int *mosfet;  8
        0 long vacuum_tube[2];  16
     } transistor;
 24  struct ms_pacman *connector;  32
};
```

Below are given four C functions and four x86-64 code blocks.

下面给出 4 个 c 函数和 4 段 x86-64 代码。

```
char* inky(struct ms_pacman *ptr){
  return &(ptr->transistor.bjt);
}

long blinky(struct ms_pacman *ptr){
  return ptr->connector->
         transistor.vacuum_tube[1];
}

int pinky(struct ms_pacman *ptr){
  return ptr->resistor;
}

int clyde(struct ms_pacman *ptr){
  return ptr->transistor.mosfet;
}
```

A	mov 0x8(%rdi), %rax retq
B	lea 0x8(%rdi), %rax retq
C	mov 0x4(%rdi), %eax retq

D	mov 0x18(%rdi),%rax
	mov 0x10(%rax),%rax
	retq

Practice 2

（x86-64） Consider the following data structure declaration:

Consider the following data structure declarations:

```
struct node {
   unsigned uid;
   union data d;
   struct node *next;
};
```

```
union data {
   int x[3];
   long y[3];
};
```

Below are given four C functions and five x86-64 code blocks, compiled on Linux using GCC.

```
int odin(struct node *ptr) {
  return (ptr->d.x[2]);
}
```

A	mov	0x20(%rdi),%rax
	mov	0x8(%rax),%rax

```
unsigned dva(struct node *ptr) {
  return (ptr->uid = (long)ptr->next);
}
```

B	mov	0x10(%rdi),%eax

C	mov	0xc(%rdi),%rax

```
long tri(struct node *ptr) {
  union data *dptr = (union data *)ptr->next;
  return dptr->y[1];
}
```

D	mov	0x20(%rdi),%rax
	add	$0x8,%rax

```
long *chetyre(struct node *ptr) {
  return &ptr->next->d.y[0];
}
```

E	mov	0x20(%rdi),%rax
	mov	%eax,(%rdi)