

Recitation 2 Code

From Xiao Liu

Exercise 1

```
double midpoint(long long a, long long b) {  
    return (a + b) / 2.0;  
}
```

Exercise 2

```
int main(void) {  
    char c;  
    //    scanf("%c", &c);  
    c = (char) getchar();  
    printf("%d", c - '0');  
}
```

Exercise 3

```
int main(void){
    int x = 0;
    char c = (char)getchar();
    while (isspace(c))
//        while(c == " ")
        c = (char)getchar();
    while (isdigit(c)) {
//        while(c >= '0' && c <= '9'){
        x = x * 10 + c - '0';
        c = (char)getchar();
    }
    printf("%d", x);
    return 0;
}
```

Exercise 4

```
int main(void){  
    int a = 1;  
    int b = (a++, a + 100, a 10);  
    printf("b = %d", b);  
}
```

Exercise 6

```
unsigned test_bit(unsigned x, unsigned i) {  
    return (x >> i) & 1u;  
}
```

or

```
unsigned test_bit(unsigned x, unsigned i) {  
    return (x & (1u << i)) >> i;  
}
```

Exercise 7

```
unsigned bit_flip(unsigned x, unsigned i) {  
    return x ^ (1u << i);  
}
```

Exercise 8

- First, right-shift by `low` bits, then bitwise-AND it with a number whose last `high - low` bits are all `1` s, and the rest are all `0` s.

```
unsigned bit_slice(unsigned x, unsigned low, unsigned high) {  
    return (x >> low) & ((1u << (high - low)) - 1);  
    // firstly, right shift `low` bits  
    // then we need to AND(&) "0...00011111"  
}
```

- Alternatively, first bitwise-AND it with a number whose last `high` bits are all `1` s and the rest are all `0` s, then right-shift by `low` bits.

```
unsigned bit_slice(unsigned x, unsigned low, unsigned high) {  
    return (x & ((1u << high) - 1)) >> low;  
}
```