# 1.大 O 表示法练习

请用大 O 表示法表示算法的时间复杂度:

算法 1：

```
void Algorithm01(){

    int sum = 0, n = 100;
    sum = (1 + n)*n / 2;
    printf("%d\n",sum);
}
```

算法 2：

```
void Algorithm02(){

    int sum = 0, n = 100;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    sum = (1 + n)*n / 2;
    printf("%d\n", sum);
}
```

算法 3：

```
void Algorithm03(int n){

    int i;
    for (i = 0; i < n;i++){
        printf("%d\n",i);
    }
}
```

算法 4：

```
void Algorithm04(int n){

    int count = 1;
    while (count < n ){
        count = count * 2;
    }
}
```

算法 5：

```
void Algorithm05(int n){

    int i, j;
    for (i = 0; i < n;i++){
        for (j = 0; j < n;j++){
            printf("%d\n",i+j);
        }
    }
}
```

算法 6：

```
void Algorithm06(int n){

    int i, j;
    for (i = 0; i < n; i++){
        for (j = i; j < n; j++){
            printf("%d\n", i + j);
        }
    }
}
```

算法 7：

```
void function07(){
    printf("hello world!\n");
}

void Algorithm07(int n){

    int i, j;
    for (i = 0; i < n; i++){
```

```
            function07();
        }
}
```

算法 8：

```
void function08(int n){
    int i;
    for (i = 0; i < n;i++){
        printf("hello world!");
    }
}

void Algorithm08(int n){

    n++;
    function08(n);
    int i, j;
    for (i = 0; i < n;i++){
        function08(n);
    }
    for (i = 0; i < n;i++){
        for (j = i; j < n;j++){
            printf("hello world!\n");
        }
    }
}
```