##Prepare 20 pseudo codes in recursion based on C

1.
```
int fun(int n) {
    if(n == 0)
        return 0;
    return fun(n - 1) + n;
}
int main() {
    printf("%d", fun(5));
}
```

Output: 15

2.
```
int fun(int n) {
    if(n <= 1)
        return 1;
    return fun(n - 1) * 2;
}
int main() {
    printf("%d", fun(4));
}
```

Output: 8

3.
```
int fun(int n) {
    if(n == 1)
        return 1;
    return fun(n - 1) + fun(n - 1);
}
int main() {
    printf("%d", fun(3));
}
```

Output: 4

4.
```
int fun(int n) {
    if(n <= 0)
        return 0;
```

```
    return fun(n - 2) + n;
}
int main() {
    printf("%d", fun(7));
}
```

Output: 16

5.
```
int fun(int n) {
    if(n < 10)
        return n;
    return fun(n / 10) + n % 10;
}
int main() {
    printf("%d", fun(345));
}
```

Output: 12

6.
```
int fun(int n) {
    if(n == 0)
        return 1;
    if(n % 2 == 0)
        return fun(n - 1);
    return fun(n - 1) + 2;
}
int main() {
    printf("%d", fun(5));
}
```

Output: 6

7.
```
int fun(int n) {
    if(n <= 1)
        return 1;
    return n + fun(n / 2);
}
int main() {
```

```
    printf("%d", fun(8));
}
```

Output: 15

8.
```
int fun(int n) {
    if(n == 0)
        return 0;
    return fun(n / 10) * 10 + (n % 10);
}
int main() {
    printf("%d", fun(123));
}
```

Output: 123

9.
```
int fun(int n) {
    if(n == 0)
        return 1;
    return fun(n - 1) + fun(n - 1) + 1;
}
int main() {
    printf("%d", fun(3));
}
```

Output: 15

10.
```
int fun(int n) {
    if(n <= 1)
        return n;
    return fun(n - 1) - fun(n - 2);
}
int main() {
    printf("%d", fun(6));
}
```

Output: 1

11.
```
int fun(int n) {
    if(n == 0)
        return 0;
    return fun(n - 1) * 2 + 1;
}
int main() {
    printf("%d", fun(4));
}
```

Output: 15

12.
```
int fun(int n) {
    if(n <= 0)
        return 1;
    return n * fun(n - 2);
}
int main() {
    printf("%d", fun(6));
}
```

Output: 48

13.
```
int fun(int n) {
    if(n < 10)
        return 1;
    return 1 + fun(n / 10);
}
int main() {
    printf("%d", fun(9876));
}
```

Output: 4 // number of digits

14.
```
int fun(int n) {
    if(n == 1)
        return 2;
    return fun(n - 1) * fun(n - 1);
}
```

```
int main() {
    printf("%d", fun(3));
}
```

Output: 16

15.
```
int fun(int n) {
    if(n == 0)
        return 0;
    return fun(n - 1) + (n % 2 == 0);
}
int main() {
    printf("%d", fun(6));
}
```

Output: 3 // count of even numbers from 1 to 6

16.
```
int fun(int n) {
    if(n <= 1)
        return 1;
    return fun(n - 1) + n * fun(n - 2);
}
int main() {
    printf("%d", fun(5));
}
```

Output: 14

17.
```
int fun(int n) {
    if(n == 0)
        return 1;
    return fun(n / 2) + n;
}
int main() {
    printf("%d", fun(4));
}
```

Output: 7

18.
```
int fun(int n) {
    if(n <= 1)
        return 1;
    return fun(n - 2) + fun(n - 1);
}
int main() {
    printf("%d", fun(7));
}
```

Output: 13

19.
```
int fun(int n) {
    if(n == 0)
        return 0;
    return fun(n - 1) + (n / 2);
}
int main() {
    printf("%d", fun(5));
}
```

Output: 6

20.
```
int fun(int n) {
    if(n == 1)
        return 3;
    return fun(n - 1) + fun(n - 1);
}
int main() {
    printf("%d", fun(4));
}
```

Output: 24