

C, C++, DSA in depth

Doubt class assignment-16



Saurabh Shukla (MySirG)

1
2

First N natural numbers

1 2 3 4 5 . . . n

```
void print_n( int n )  
{  
    if (n > 0)  
    {  
        print_n(n - 1);  
        printf("%d", n);  
    }  
}
```

① print_n(n) 1 2 3 4 .. (n-1) n
RC ② print_n(n-1) 1 2 3 4 ... n-1
RC ③ printf("%d", n);
n → 0

Print first n natural numbers in reverse order

n n-1 n-2 ... 4 3 2 1

void print_reverse_n (int n)

{

if (n > 0)

{
printf ("%d ", n);
print_reverse_n(n-1);

}

① print_reverse_n(n) n n-1 n-2 ... 4 3 2 1

BC ② printf ("%d ", n); n

print_reverse_n(n-1) n-1 n-2 ... 4 3 2 1

BC ③ n → 0

}

Print first n odd natural numbers

1 3 5 7 ... $2 \times n - 1$

```
void print_odd_n(int n) {  
    if (n > 0) {  
        print_odd_n(n-1);  
        printf("%d", 2*n-1);  
    }  
}
```

① $\text{print_odd_n}(n)$ | 3 5 7 9 11... $2n-3$ $2n-1$
② $\text{print_odd_n}(n-1)$ | 3 5 7... $2n-3$
 $\text{printf}(\text{" \%d"}, 2 \times n - 1);$
③ $n \rightarrow 0$

First n odd natural numbers in reverse order

$2n-1 \quad 2n-3 \quad 2n-5 \quad 2n-7 \quad \dots \quad 5 \quad 3 \quad 1$

void print_odd_reverse_n(int n)

{

if ($n > 0$)

{ printf("%d ", 2*n - 1);

print_odd_reverse_n(n-1);

}

① print_odd_reverse_n(n) $2n-1 \quad 2n-3 \dots 5 \quad 3 \quad 1$

RC ② printf("%d ", 2*n - 1);

print_odd_reverse_n(n-1) $2n-3 \quad 2n-5 \dots 5 \quad 3 \quad 1$

}

^{BC} ③ $n \rightarrow 0$

First n even natural numbers

2 4 6 8 ... $2n-2$ $2n$

void print_even_n(int n)

{

if ($n > 0$)

{
 print_even_n($n-1$);
 printf("%d", $2*n$);

}

① Print_even_n(n) 2 4 6 8 ... $2n-2$ $2n$

RC ② print_even_n($n-1$) 2 4 6 8 ... $2n-2$
 printf("%d", $2*n$);

③ $n \rightarrow 0$

}

print first n even natural numbers in reverse

2n 2n-2 2n-4 2n-6 ... 8 6 4 2

void print_even_reverse_n(int n)

{

 if (n > 0)
 { printf("%d ", 2*n);
 print_even_reverse_n(n-1);

}

① print_even_reverse_n(n) 2n 2n-2 ... 6 4 2

RL ② printf("%d ", 2*n);
 print_even_reverse_n(n-1) 2n-2 2n-4...6 4 2

}

BY ③ $n \rightarrow 0$

Print Squares of first n natural numbers

1 4 9 16 25 36 ... $(n-1) \times (n-1)$ $n \times n$

void print_square(int n)

{

if ($n > 0$)

{ print_square($n-1$);
printf("%d", $n \times n$);

}

① print_square(n) 1 4 9 16 ... n^2

RC ② print_square($n-1$) 1 4 9 16 ... $(n-1)^2$
printf("%d", $n \times n$);

B4 ③ $n \rightarrow 0$

}

Decimal to Binary

$$25 \rightarrow 11001$$

```
void dtob(int n)
```

```
{     if (n > 0)  
{         ... }
```

$\text{d}\mapsto b(n/2)$;

```
printf("%d", n/2);
```

7

2

$$\begin{array}{r}
 2 \overline{)25} \\
 2 \overline{)12} \quad 1 \\
 2 \overline{)6} \quad 0 \\
 2 \overline{)3} \quad 0 \\
 2 \overline{)1} \quad 1 \\
 \end{array}$$

① dtob(25) 11001

R^L ② d to b ($\frac{7}{25}$) 1100

```
printf("%d", n % 2);
```

β^c (3) $n \rightarrow 0$

Decimal to Octal

$$25 \rightarrow 31$$

$$\begin{array}{r} 8 | 25 \\ 8 | 3 \\ \hline 0 & 3 \end{array} \quad \uparrow$$

void dtoO(int n)

{

if(n>0)

{
 dtoO(n/8);

 printf("%d", n%8);

}

① $\text{dtoO}(\frac{n}{25})$

RC ② $\text{dtoO}(\frac{\frac{n}{25}}{8})$ 3
 printf("%d", $\frac{25}{n} \% 8$); 1

B ③ $n \rightarrow 0$

}

Reverse a number

$3572 \rightarrow 2753$

void reverse(int n)

```
{  
    if(n>0)  
    {  
        printf(".d",n%10);  
        reverse(n/10);  
    }  
}
```

① reverse($\frac{n}{10}$) 2753
② printf(".d", $\frac{n}{10} \% 10$); 2
reverse($\frac{n}{10}/10$) 753
③ $n \rightarrow 0$