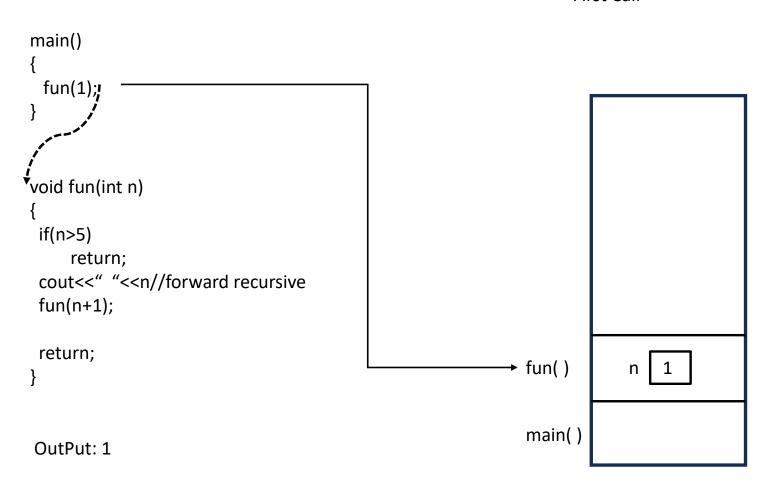
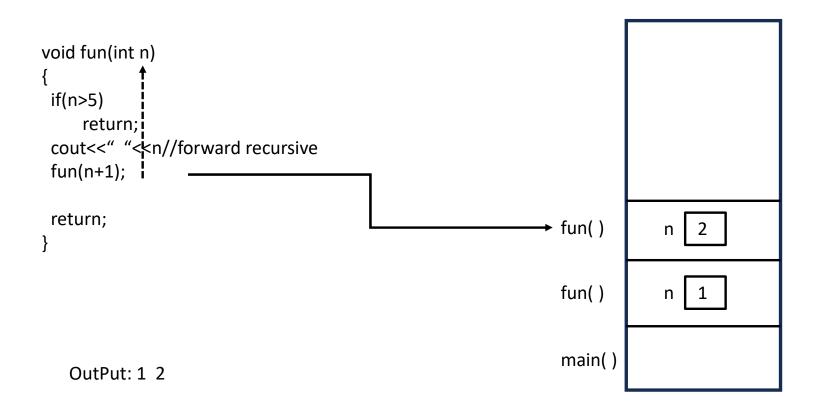
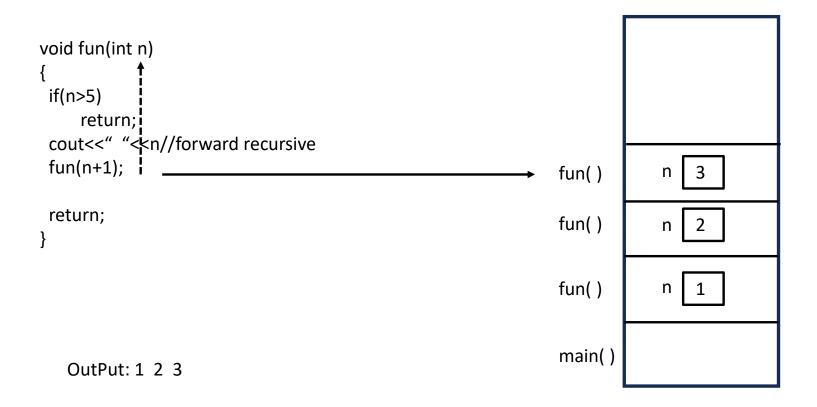
First Call

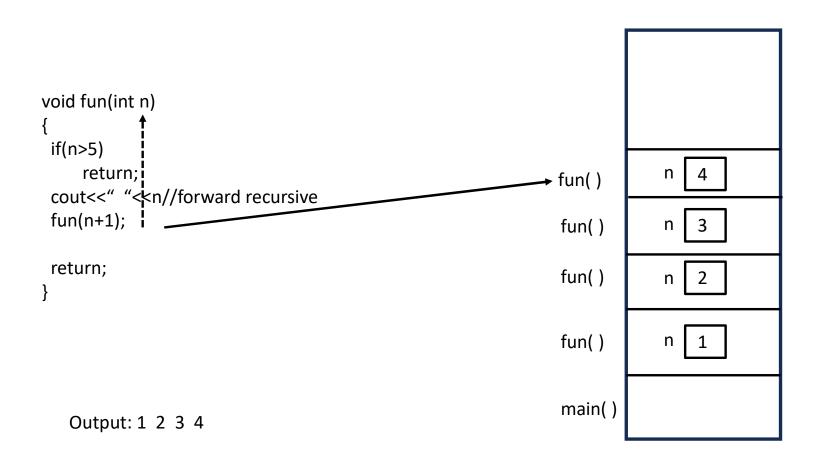


Second Call

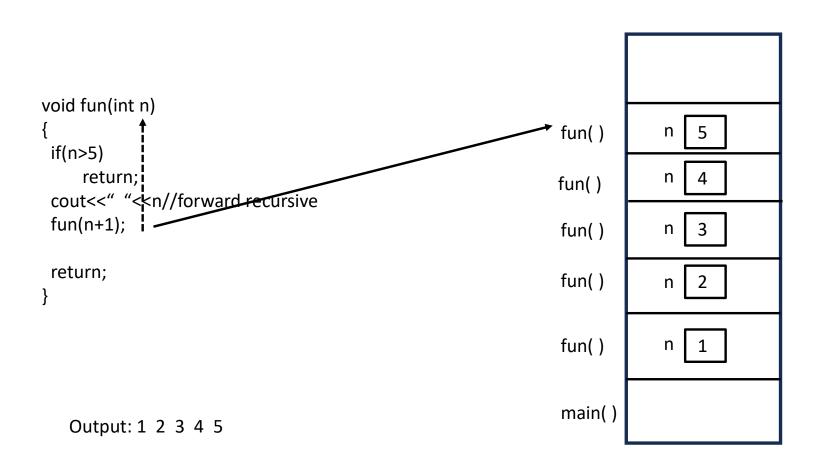




Forth Call



Fifth Call



Sixth Call

void fundint n	fun()	n 6
<pre>void fun(int n) { if(n>5) Base condition is</pre>	fun()	n 5
return; true, return cout<<" "< <n r="">/forward recursive</n>	fun()	n 4
fun(n+1);	fun()	n 3
return; }	fun()	n 2
	fun()	n 1
Output: 1 2 3 4 5	main()	

Fifth Call

		Released
<pre>void fun(int n) { if(n>5) return; cout<<" "<<n control="" execution="" forward="" fun(n+1);="" here="" pre="" recursive="" return;="" returns="" }<=""></n></pre>	fun()	n 5
	fun() fun()	n 4 n 3
	fun()	n 2
	fun()	n 1
Output: 1 2 3 4 5	main()	

Forth Call

```
Released
void fun(int n)
                                                                      Released
 if(n>5)
    return;
                                                                       n
                                                           fun()
cout<<" "<<n//forward recursive
fun(n+1);
                                                           fun()
                                                                       n 3
         // execution control returns here
 return;
                                                           fun()
                                                                       n
                                                           fun()
                                                                       n | 1
                                                           main()
   Output: 1 2 3 4 5
```

Third Call

```
Released
void fun(int n)
                                                                      Released
 if(n>5)
    return;
                                                                      Released
cout<<" "<<n//forward recursive
fun(n+1);
                                                            fun()
                                                                        n 3
         // execution control returns here
 return;
                                                            fun()
                                                                        n
                                                            fun()
                                                                        n | 1
                                                            main()
   Output: 1 2 3 4 5
```

Second Call

```
Released
void fun(int n)
                                                                      Released
 if(n>5)
    return;
                                                                      Released
cout<<" "<<n//forward recursive
fun(n+1);
                                                                      Released
         // execution control returns here
 return;
                                                            fun()
                                                                       n 2
                                                            fun()
                                                                        n | 1
                                                            main()
   Output: 1 2 3 4 5
```

Second Call

```
Released
void fun(int n)
                                                                      Released
 if(n>5)
    return;
                                                                      Released
cout<<" "<<n//forward recursive
fun(n+1);
                                                                      Released
         // execution control returns here
 return;
                                                                      Released
                                                            fun()
                                                                       n 1
                                                            main()
   Output: 1 2 3 4 5
```

```
main()
                                                                   Returned to main()
  fun(1);
          // execution control returns here
}
                                                                       Released
void fun(int n)
                                                                       Released
 if(n>5)
                                                                       Released
    return;
 cout<<" "<<n//forward recursive
                                                                       Released
fun(n+1);
                                                                       Released
 return;
                                                                       Released
                                                             main()
   Output: 1 2 3 4 5
```

```
main()
{
  fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
    return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```



```
main()
{
fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
    return;
  cout<<" "<<n //Execute
  fun(n+1);
  return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
main()
{
  fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
     return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 3 Third Call
{
  if(n>5)
    return;
  cout<<" "<<n //forward recursive
  fun(n+1);
  return;
}</pre>
```

```
main()
{
  fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
     return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 3 Third Call
{
  if(n>5)
     return;
  cout<<" "<<n//forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 4 Forth Call
{
  if(n>5)
    return;
  cout<<" "<<n //forward recursive
  fun(n+1);
  return;
}</pre>
```

```
main()
{
  fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
     return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 3 Third Call
{
  if(n>5)
    return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 4 Forth Call
{
  if(n>5)
    return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 5 Fifth Call
{
  if(n>5)
    return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
main()
{
  fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
     return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 3 Third Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 4 Forth Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 5 Fifth Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
main()
{
  fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
    return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 3 Third Call
{
  if(n>5)
    return;
  cout<<" "<<n//forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 4 Forth Call
{
  if(n>5)
     return;
  cout<<" "<<n//>forward recursive
  fun(n+1);

return;
}
```

```
void fun(int n) // n = 5 Fifth Call
{
  if(n>5)
    return;
  cout<<" "<<n//forward recursive
  fun(n+1);
    // control returns
  return;
}</pre>
```

```
main()
{
  fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
    return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 3 Third Call
{
  if(n>5)
     return;
  cout<<" "<<n//forward recursive
  fun(n+1);
  return;
}</pre>
```

```
void fun(int n) // n = 4 Forth Call
{
  if(n>5)
     return;
  cout<<" "<<n//forward recursive
  fun(n+1);
     // control returns
  return;
}</pre>
```

```
main()
{
  fun(1);
}
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
     return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
     return;
  cout<<" "<<n //forward recursive
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 3 Third Call
{
  if(n>5)
    return;
  cout<<" "<<n//forward recursive
  fun(n+1);
    // control returns
  return;
}</pre>
```

Released

```
main()
{
  fun(1);
}
```

```
Released
```

```
Released
```

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
    return;
  cout<<" "<<n //Execute
  fun(n+1);

return;
}</pre>
```

```
void fun(int n) // n = 2 Second Call
{
  if(n>5)
    return;
  cout<<" "<<n //forward recursive
  fun(n+1);
    // control returns
  return;
}</pre>
```

```
main()
{
  fun(1);
}
```

Released

```
void fun(int n) // n = 1 First Call
{
  if(n>5)
    return;
  cout<<" "<<n //Execute
  fun(n+1);
    // control returns
  return;
}</pre>
```

Released

Released

```
main()
{
  fun(1);
  // control returns
}
```

Released

Released

Released

Released