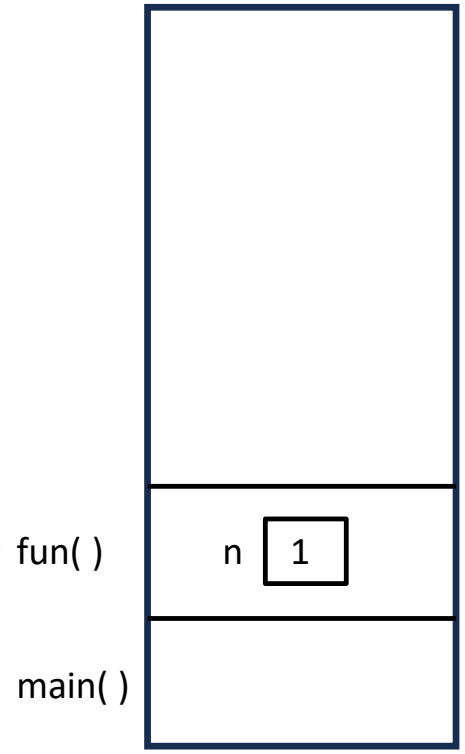
First Call

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

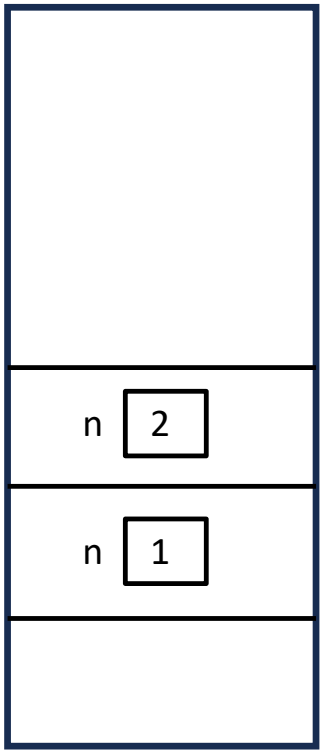
```
void fun(int n)
{
 if(n>5)
      return;
 cout<<“ “<<n//forward recursive
 fun(n+1);

 return;
}
```



OutPut: 1

Second Call

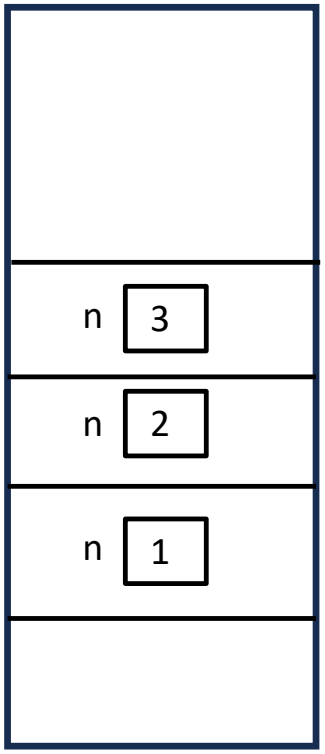
```
void fun(int n)
{
  if(n>5)
        return;
  cout<<"  "<<n//forward recursive
  fun(n+1);

  return;
}
```



OutPut: 1 2

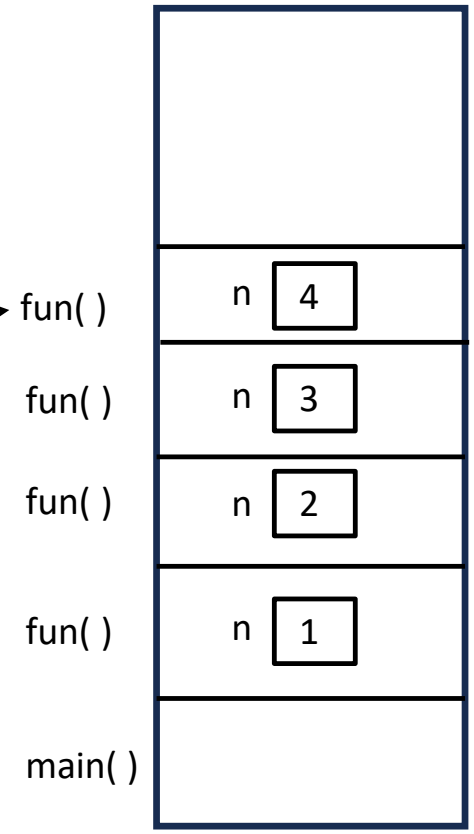
Third Call

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

  return;
}
```



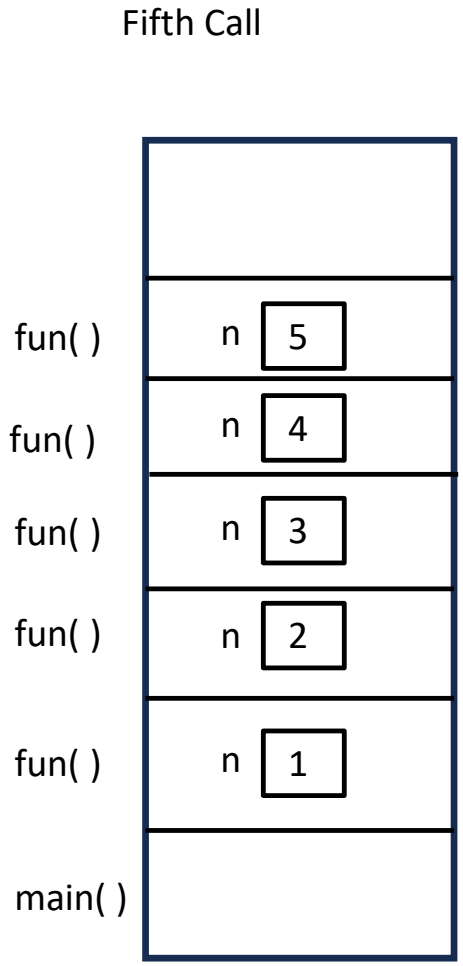
OutPut: 1 2 3

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

  return;
}
```



Output: 1 2 3 4

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

  return;
}
```



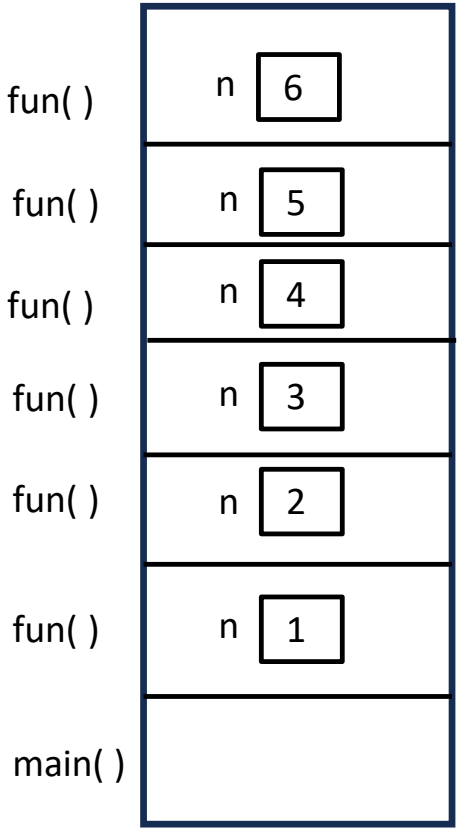
Output: 1 2 3 4 5

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

 return;
}
```

Base condition is true, return

Output: 1 2 3 4 5

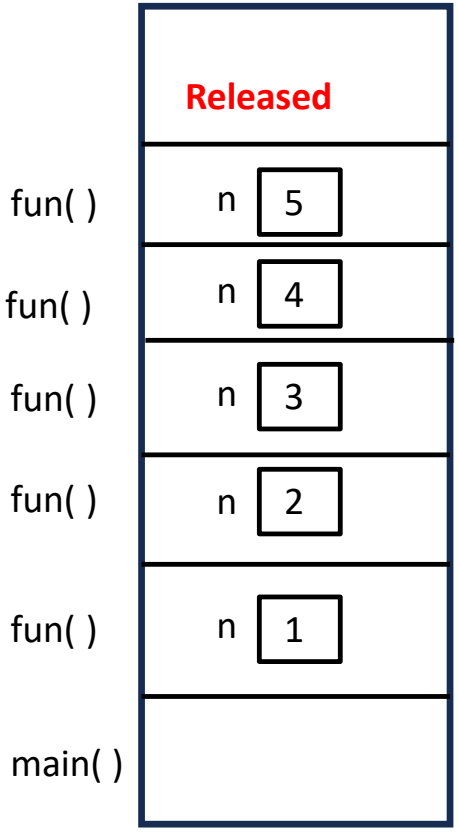
Sixth Call



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

Output: 1 2 3 4 5

Fifth Call



```
void fun(int n)
{
  if(n>5)
        return;
  cout<<“  “<<n//forward recursive
  fun(n+1);
                // execution control returns here
  return;
}
```

Output: 1 2 3 4 5

Forth Call

```
void fun(int n)
{
  if(n>5)
        return;
  cout<<“  “<<n//forward recursive
  fun(n+1);
                // execution control returns here
  return;
}
```

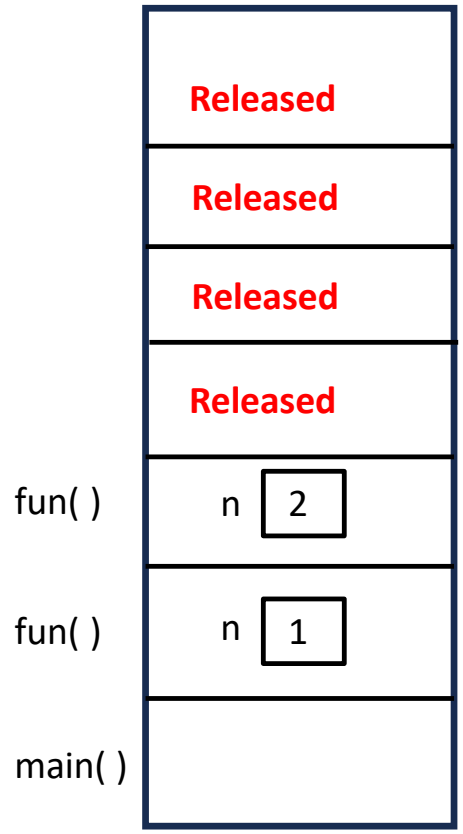
Output: 1 2 3 4 5

Third Call

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

Output: 1 2 3 4 5

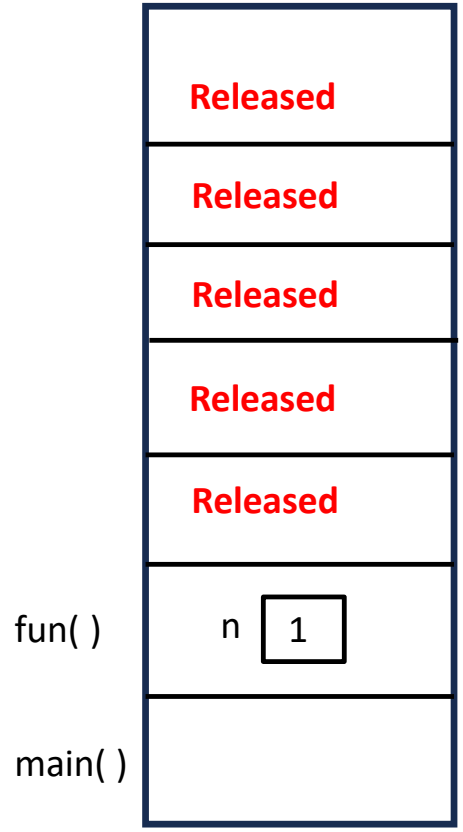
Second Call



```
void fun(int n)
{
  if(n>5)
        return;
  cout<<“  “<<n//forward recursive
  fun(n+1);
                // execution control returns here
  return;
}
```

Output: 1 2 3 4 5

Second Call



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

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

 return;
}
```

Output: 1 2 3 4 5

Returned to main( )

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

Once More 

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

 return;
}
```

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

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

 return;
}
```

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

 return;
}
```

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

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

 return;
}
```

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

 return;
}
```

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

 return;
}
```

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

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

 return;
}
```

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

 return;
}
```

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

 return;
}
```

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

 return;
}
```

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

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

 return;
}
```

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

 return;
}
```

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

 return;
}
```

```
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);

 return;
}
```

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

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

 return;
}
```

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

 return;
}
```

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

 return;
}
```

```
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);

 return;
}
```

```
void fun(int n)  // n = 6   Sixth Call
{
 if(n>5)          // condition true
      return;    //return
 cout<<“ “<<n
 fun(n+1);

 return;
}
```

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

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

 return;
}
```

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

 return;
}
```

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

 return;
}
```

```
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;
}
```

Released

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

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

 return;
}
```

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

 return;
}
```

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

 return;
}
```

Released

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

Released

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

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

 return;
}
```

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

 return;
}
```

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

Released

Released

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;
}
```

Released

Released

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

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

Released

Released

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

Released

Released

Released


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