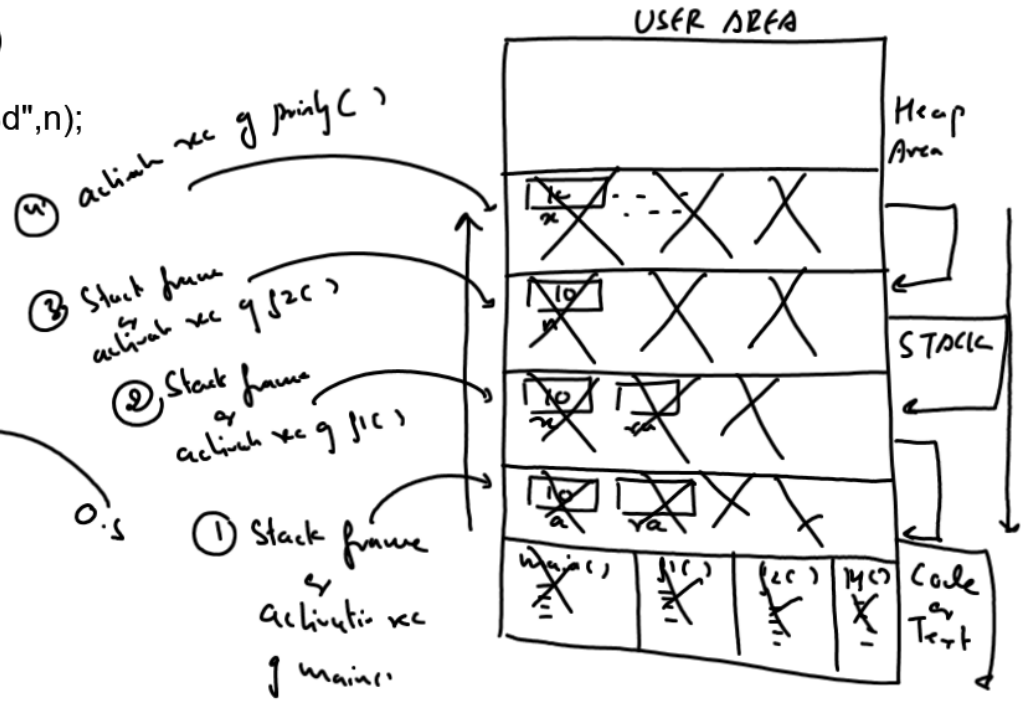
How Memory And Function Calls Are Managed During Run Time ?

==========

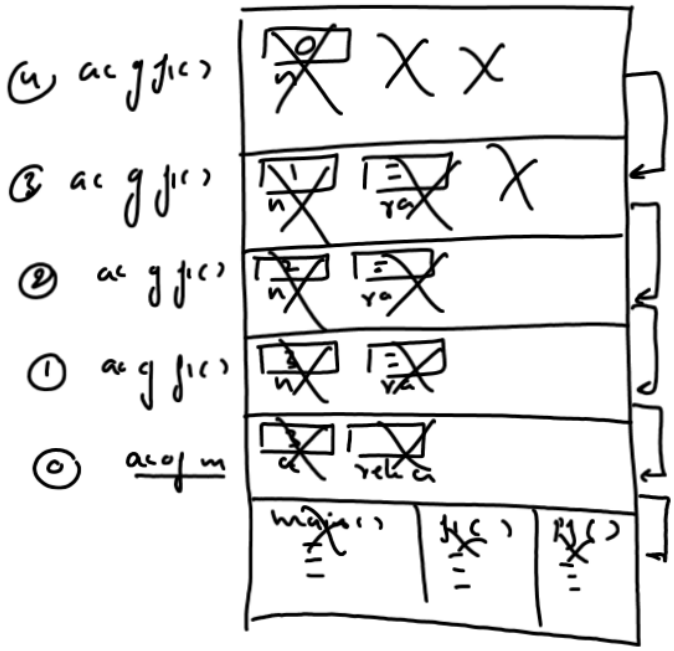
```
void f2(int n)
{
   printf("\n%d",n);
   return;
}
void f1()
{
  int x=10;
  f2(x);
  return;
}
int main()
{
   int a=10;
   f1();
   return 0;
}
```



Tracing Recursion

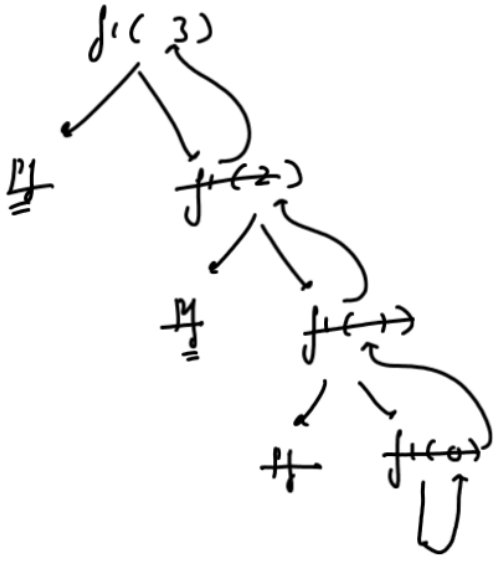
```
void f1(int n)
{
   if(n>0)
    {
       printf("\n%d",n);
       f1(n-1);
    }
  return;
}
int main()
{
   int a=3;
   f1(a);
   return 0;
}
```

OUTPUT: 3, 2, 1



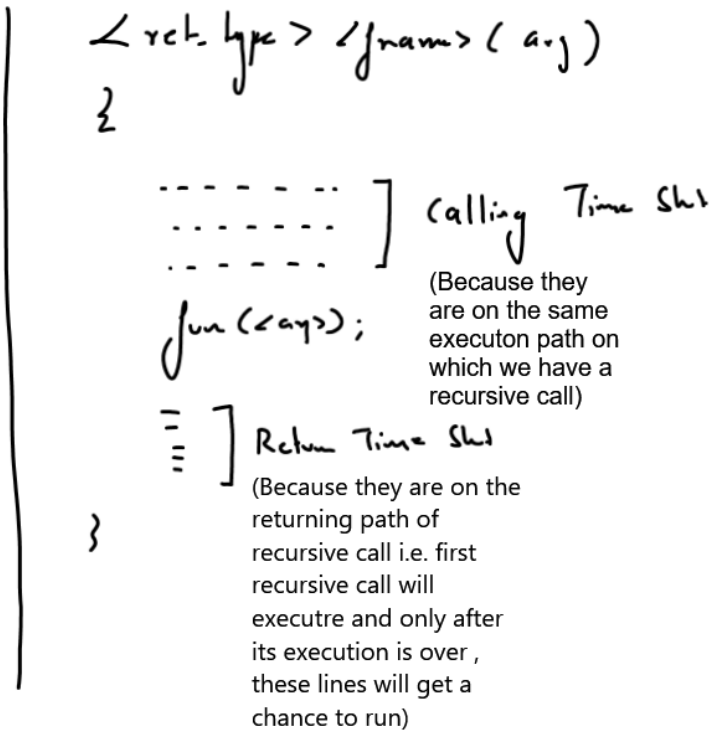
Tracing Recursion Using RecursionTree

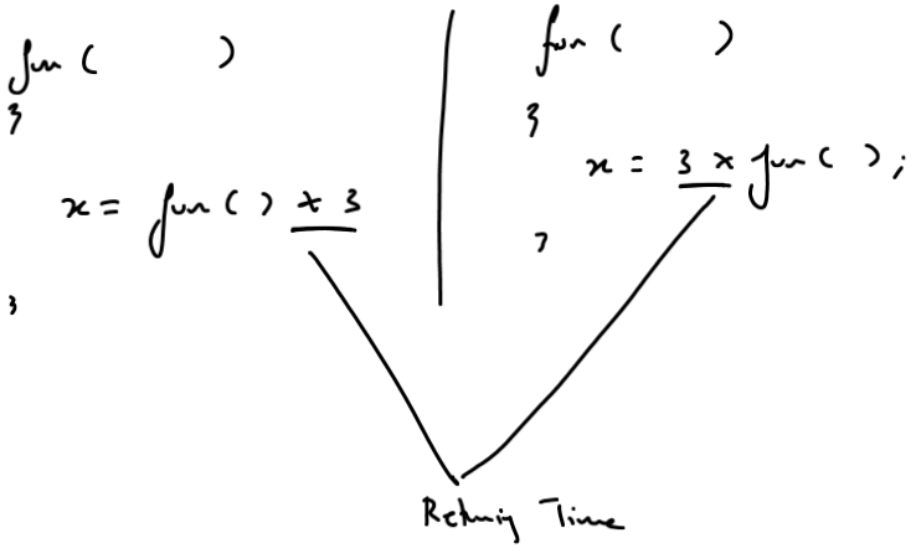
```
void f1(int n)
{
    if(n>0)
     {
        printf("\n%d",n);
        f1(n-1);
      }
  return;
}
int main()
{
    int a=3;
    f1(a);
    return 0;
}
OUTPUT: 3, 2, 1
```



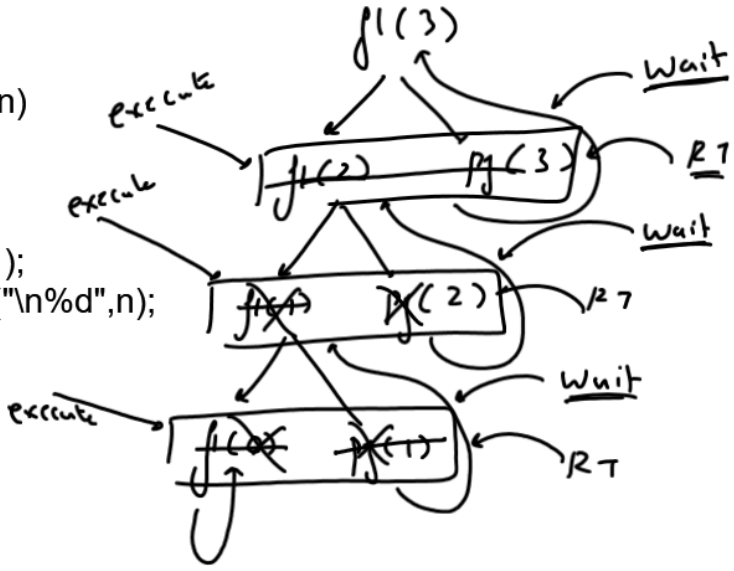
```
  void f1(int n)
  {
      if(n>0)
       {
         f1(n-1);
         printf("\n%d",n);

        }
    return;
  }
  int main()
  {
      int a=3;
      f1(a);
      return 0;
  }
  OUTPUT
```

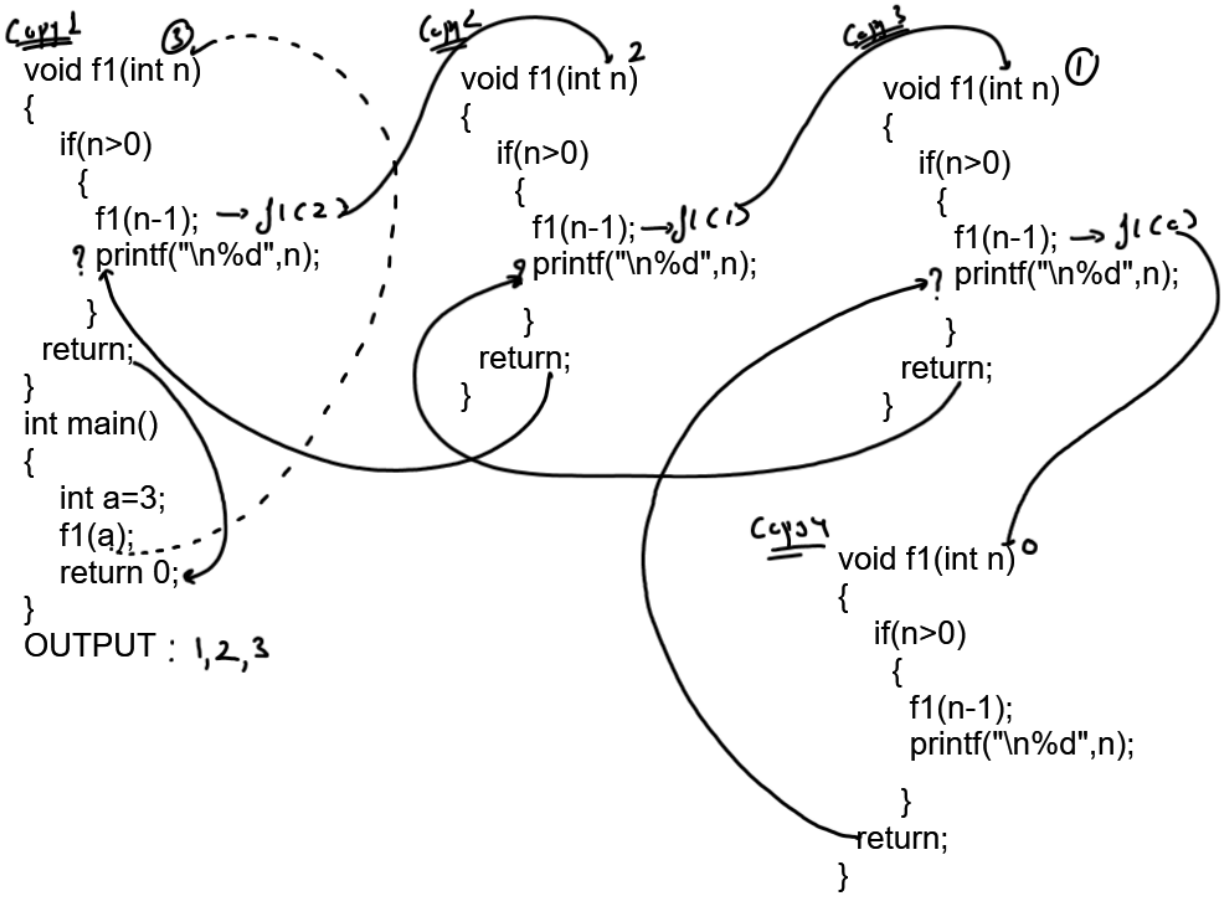


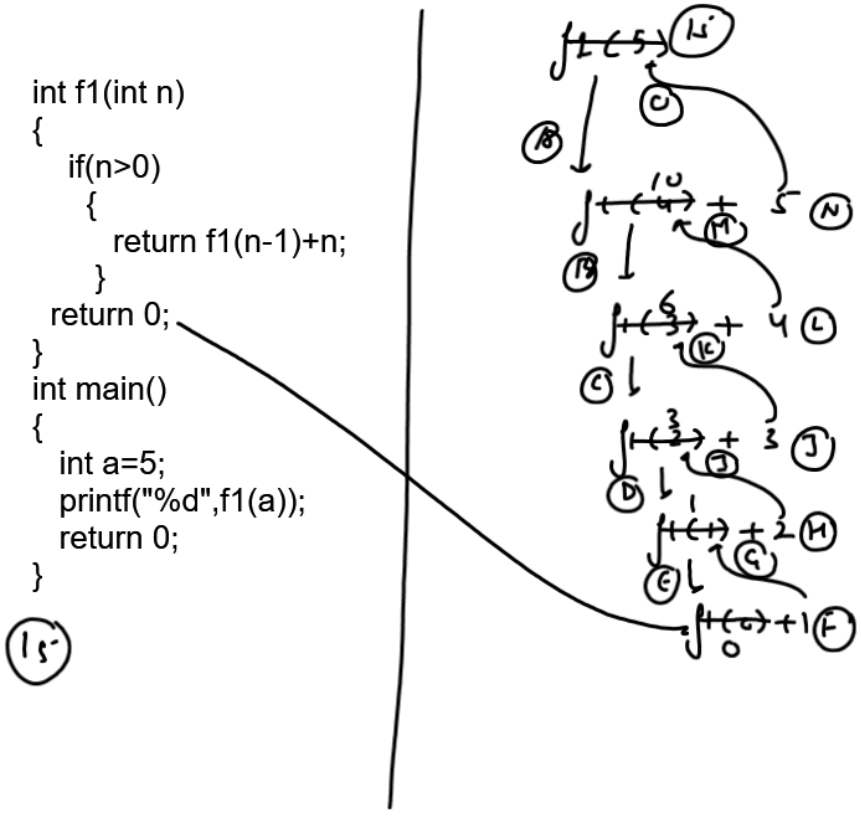



```
void f1(int n)
{
   if(n>0)
    {
      f1(n-1);
      printf("\n%d",n);
    }
  return;
}
int main()
{
   int a=3;
   f1(a);
   return 0;
}
```



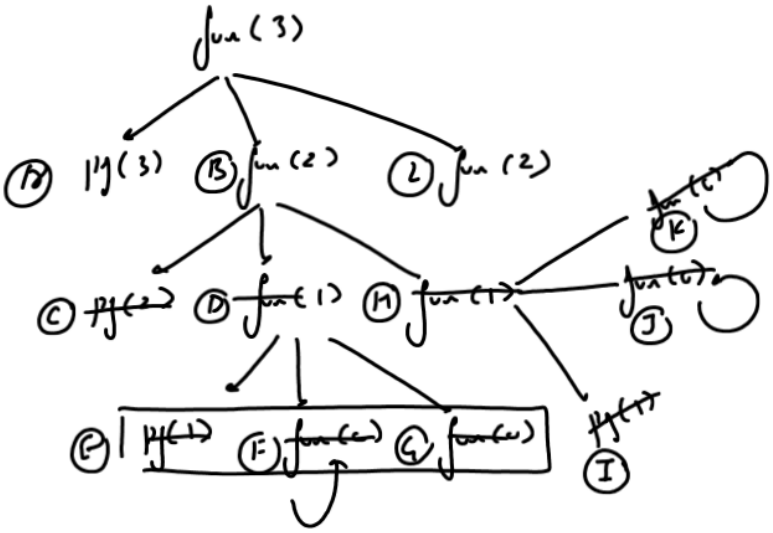
OUTPUT : 1, 2, 3


Copy 1
void f1(int n)
{
if(n>0)
{
f1(n-1);
printf("\n%d",n);
}
return;
}
int main()
{
int a=3;
f1(a);
return 0;
}
OUTPUT : 1,2,3
Copy 2
void f1(int n)
{
if(n>0)
{
f1(n-1);
printf("\n%d",n);
}
return;
}
Copy 3
void f1(int n)
{
if(n>0)
{
f1(n-1);
printf("\n%d",n);
}
return;
}
Copy 4
void f1(int n)
{
if(n>0)
{
f1(n-1);
printf("\n%d",n);
}
return;
}


int f1(int n)
{
if(n>0)
{
return f1(n-1)+n;
}
return 0;
}
int main()
{
int a=5;
printf("%d",f1(a));
return 0;
}

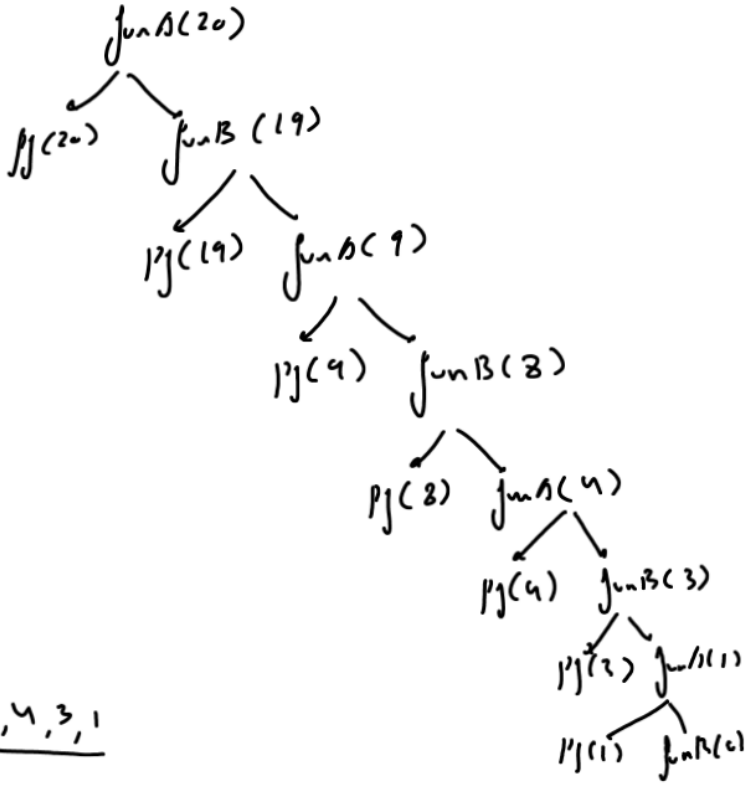
```
void fun(int n)
{
  if(n>0)
    {
       printf("%d,",n);
       fun(n-1);
       fun(n-1);
   }
}
void main()
{
    int a=3;
    fun(a);
}
```

Op: 3,2,1,1,2,1,1



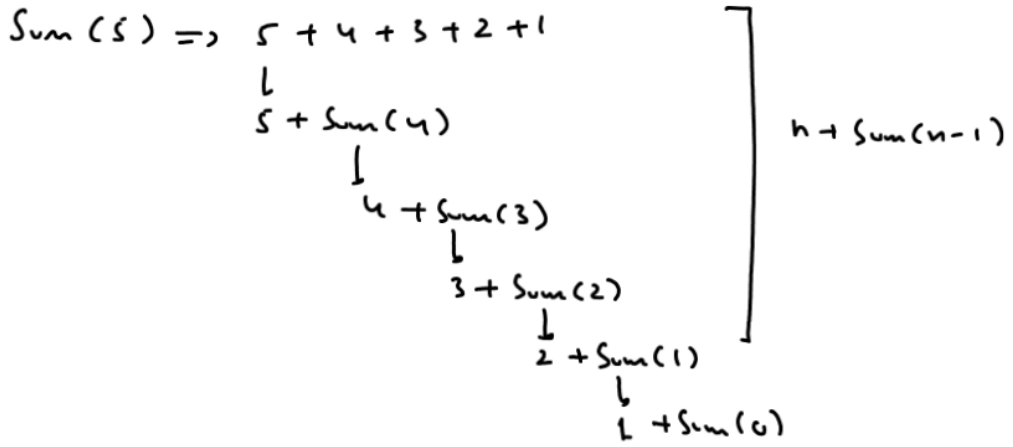

```
void funA(int n)
{
  if(n>0)
    {
       printf("%d,",n);
       funB(n-1);
     }
}
void funB(int n)
{
  if(n>0)
    {
      printf("%d,",n);
       funA(n/2);
    }
}
void main()
{
   int x=20;
   funA(x);
}
```

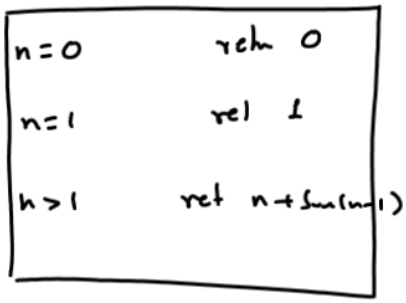
20,19,9,8,4,3,1



Some Popular Recursive Problems

1. Sum of first n natural numbers






```
int Sum(int n)
{
    if(n<=1)
        return n;

    return n+Sum(n-1);
}
```