Part a)

```
void rhelp(int m) {
    if (m == 1) return;
    else {
        rhelp(m-1);
    }
}
```

} runs n times

```
void rfunc(int n, int m)
{
    if (n < 1) return
    else {
        rhelp(n);
        rfunc(n-m, m);
    }
}
```

→ runs m times since m is sqrt(n) so $m \times m = n$

```
int main() {
    int n;
    cin>>n;
    rfunc(n, sqrt(n));
}
```

$O(n) + O(m) = \underline{\underline{O(n)}}$

$O(m) \rightarrow O(\sqrt{n})$

Part b)

```
int n;
int* A;
int x = n; <-

int f1(int* A) {
  if(x==0) {
    x=n
    return 1;
  }
else if(x %(int) sqrt(n))==0) {
   for(int i=1; i<=n; i++){
     for(int j=0; j<i; j++){
     }
   }
}
else {}
x--;
return 0;
}
```

$\rightarrow O(n^2) \cdot \sum_{i=1}^{n} \sum^{i} O(1)$

$= \sum_{i=1}^{n} O(i) = O(n^2)$

b

this function will always be entered since x is set to n. in the beginning

$\frac{\not{x}\, O(n^2)}{\not{x}} = O(n^2)$