Runtime Analysis

a.)
```
void f1(int n) {
    int i = 2;
    while (i < n) {        O(√n)
        i = i * i;         O(1)
    }
}
```

$$T(n) = \sum_{i=2}^{\sqrt{n}} (O(1)) = \boxed{\Theta(\sqrt{n})}$$

b.)
```
void f2(int n) {                                    Θ(n)
    for(int i = 1; i <= n; i++) {                   O(1)
        if((i % (int)sqrt(n)) == 0) {
            for(int k = 0; k < pow(i, 3); k++) {    Θ(i^3)
                /* do something
            }
        }
    }
}
```

$$T(n) = \sum_{i=1}^{n} (\Theta(1)) + \sum_{i=0}^{n-1} (\Theta(i^3))$$

$$= \Theta(n) + \Theta((n-1)^4) = \boxed{\Theta(n-1)^4}$$

2m

c.)
```
for(int i = 1; i <= n; i++) {                  O(n)
    for(int k = 1; k <= n; k++) {              O(n)
        if(A[k] == i) {                        O(n)
            for(int m = 1; m <= n; m = m+m) {  O(log₂n)
}}}}
```

10000	14
n = 1000	10
100	7
10	4

$$T(n) = \sum_{i=1}^{n} (\Theta(1) + \Theta(1) + \Theta(1) + \Theta(\tfrac{1}{i}))$$

$$= \boxed{\Theta(\log n)}$$

d)
```
int f(int n) {
    int *a = new int[10];
    int size = 10;
    for(int i=0; i<n; i++){
        if (i==size) {
            int newsize = 3*size/2;
            int *b = new int [newsize];
            for(int j=0; j<size; j++) b[j]=a[j];
            delete []a;
            a=b;
            size = new size;
        }
        a[i]=i*i;
    }
}
```

$\Theta(n)$

$\Theta(1)$ → only executes once

$$T(n) = \sum_{i=0}^{n} \left(\Theta(1) + \Theta(1)\right)$$

$$= \Theta(n) + \Theta(n) = \boxed{\Theta(n)}$$