```
int f(int x, double y)
```

return type

formal parameters

```
main() {
    int a;
    double b;
    f(a, b);
```

actual parameters

```
int f(int x) {
    :
    :
}
```

function header.

```
int f(int x);
```

← function prototype

```
vector<int> V = {1, 2, 3, 4, 5};
```

```
int A[10];
    :
vector<int> V(A+5, A+10);
```

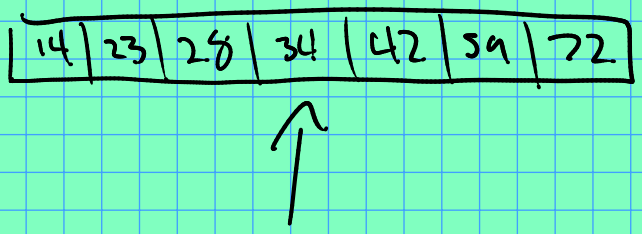
type is int*.

Remember:

cout << A[i]; ≡ cout << *(A+i)

C++: void f(int& x) {
x = ... ;
}

C: void f(int * x) {
*x = ...
}



$n, \frac{n}{2}, \frac{n}{4}, \frac{n}{8} \cdots \frac{n}{2^k}$

$n = 2^k$ when $k = \log_2 n$.

```
bool binarySearch(const vector<int>& V, int x,
                  int low, int high) {
  while (low <= high) {
    int mid = (low+high)/2;
    if( x < V[mid])
      high = mid-1;
    else if (V[mid] < x)
```

```
            low = mid + 1;
        else
            return true;
    }
    return false;
}
```

to call it:

```
vector<int> V;
    ⋮ // put stuff in V

if ( binarySearch (V, 5, 0, V.size() - 1))
        cout << "found 5 in V";
```