Listing 1: Implementacja *quicksort* w C++

```
void quickSort(int arr[], int left, int right) {
    int i = left, j = right;
    int tmp;
    int pivot = arr[(left + right) / 2];

    /* partition */
    while (i <= j) {
        while (arr[i] < pivot)
              i++;
        while (arr[j] > pivot)
              j--;
        if (i <= j) {
              tmp = arr[i];
              arr[i] = arr[j];
              arr[j] = tmp;
              i++;
              j--;
        }
    };

    /* recursion */
    if (left < j)
        quickSort(arr, left, j);
    if (i < right)
        quickSort(arr, i, right);
}
```

Listing 2: Implementacja *quicksort* w Javie

```
int partition(int arr[], int left, int right)
{
    int i = left, j = right;
    int tmp;
    int pivot = arr[(left + right) / 2];

    while (i <= j) {
        while (arr[i] < pivot)
              i++;
        while (arr[j] > pivot)
              j--;
        if (i <= j) {
              tmp = arr[i];
              arr[i] = arr[j];
              arr[j] = tmp;

              i++;
              j--;
        }
    };

    return i;
}

void quickSort(int arr[], int left, int right) {
    int index = partition(arr, left, right);

    if (left < index - 1)
        quickSort(arr, left, index - 1);
    if (index < right)
        quickSort(arr, index, right);
}
```

Spis listingów