Assignment Solutions: - Problems Sort

1) What is an in-place sorting algorithm?
   a) It needs O(1) or O(logn) memory to create auxiliary locations
   b) The input is already sorted and in-place
   c) It requires additional storage
   d) It requires additional space

A) It needs O(1) or O(logn) memory to create auxiliary locations

---

2) In the following scenarios, when will you use selection sort?
   a) The input is already sorted
   b) A large file has to be sorted
   c) Large values need to be sorted with small keys
   d) Small values need to be sorted with large keys

C)Large values need to be sorted with small keys

---

3) Given an integer array and an integer k where k<=size of array, We need to return the kth smallest element of the array.

```
#include <iostream>
using namespace std;

void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i - 1;
```

```
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}


int main() {
     int arr[5]={7,2,32,5,20};
     int size=5;
     int k=3;
     insertionSort(arr,5);
     cout<<arr[k-1]<<endl;
     return 0;

}
```

---

4) Given an array of N elements, where each element is at most K away from its target position, devise an algorithm that sorts in O(N log K)

```
#include <iostream>
using namespace std;
int main() {
     int A[6]={2,6,3,12,56,8};
     int size=6;
     int k=3;
     int i,j,key;
     for (i = 1; i < size; i++) {
        key = A[i];
        j = i - 1;

        while (j >= max(0, i - k) && A[j] > key) {
            A[j + 1] = A[j];
            j--;
        }
        A[j + 1] = key;
     }
```

```
    for(int i=0; i<size; i++)
        cout<<A[i]<<" ";
    cout<<endl;
     return 0;

}
```

---

5) Given an array, arr[] containing n integers, the task is to find an integer (say K) such that after replacing each and every index of the array by |ai − K| where (i ∈ [1, n]), results in a sorted array. If no such integer exists that satisfies the above condition then return -1.

```
#include <iostream>
using namespace std;
int main() {
     int a[5]={10, 5, 4, 3, 2};
     int l = 0, r = 1e9;
    int n=5;
    for (int i = 0; i < n - 1; i++) {



        if (a[i] < a[i + 1]) {
            r = min(r, (a[i] + a[i + 1]) / 2);
        }

        else if (a[i] > a[i + 1]) {
            l = max(l, (a[i] + a[i + 1] + 1) / 2);
        }
    }

    if (l > r) {
        cout << "-1";
    }

    else

        cout << l << endl;
    return 0;
}
```
