Bubble sort :

```
void bubble_sort (int A[], int n){

  for (i=0; i <n-1 ; i++){
        flag = 0
     for (j=0; j<n -1-i ;j++){
          if (A[j] > A[j+1]){
             temp = A[j];
             A[j] = A[j+1];
              A[j+1] = temp;
             flag= 1;
          }

      }
          if (flag==0){ break; }

     }

}
```

Selection sort :

```
void selection_sort (int A[ ], int n){
    for (i=0 ; i<n-1 ; i++){
        min= i;
      for (j= i+1 ; j<n ; j++){
          if (A[j] < A[min]){
              min=j;
          }
    } ~~else~~ if (min != i){
              temp= A [i];
              A[i ] = A[min];
              A[min] = temp;
          }
        }
  }
}
```

Insertion sort:

```
void insertion_sort (int A[], int n){
    for (i=1 ; i<n ; i++){
        temp = A[i];
        j = i-1;
        while (j>=0 && A[j]> temp){
            A[j+1] = A[j];
            j-- ;
        }
        A[j+1] = temp;
    }
}
```

Linear search:

```
int a[ ];
for (i=0; i<n ; i++){
        if (a [i] = s_data){
            cout << "data found";
        }
    }
```

Binary seach:

```
int binary_search (int A[ ], int n, int dat){
        int l, r ,mid;
        l=0;
        r=n-1;
      mid = (l+r/2);
    if (data == A[mid] ){return mid}
    else if ( data > A[mid] {l = mid+1; }
    else ~~eb~~ ~~l~~{ r = mid-1; }
    }
```