

# Exercise 2.1

2.

```
//code for insertion sort in non-increasing order
void insertion_sort_d(int number[], int num_number)
{
    for(int j = 1; j<num_number; ++j)
    {
        int key = number[j];
        int i = j-1;
        while (i>=0 && number[i]<key)
        {
            number[i+1] = number[i];
            i = i-1;
        }
        number[i+1] = key;
    }
}
```

3.

```
# code for linear search

def linear_search(A, v):
    returnable = None
    l = len(A)
    for i in range(l):
        if A[i]==v:
            returnable = i
            break
    return returnable
```

4.

```
# code for add_two _n_bit_binary integers
# assuming two inputed numbers are of the same lenght

def n_binary_add(A, B, n):
    carry = 0
    c = [0]*(n+1)
    for i in range (n-1, -1, -1):
        c[i+1] = (A[i]+B[i]+carry)%2
```

```
        carry = (A[i]+B[i]+carry)//2
        #print(carry)
    c[0] = carry
    return c
```