Dictionary_manual_inbuilt_fun & method

Dictionary-related functions without using inbuilt functions

```
def dict_length(d):
    count = 0
    for _ in d:
        count += 1
    return count
```

```
def get_keys(d):
    keys = []
    for key in d:
        keys.append(key)
    return keys
```

```
def get_values(d):
    values = []
    for key in d:
        values.append(d[key])
    return values
```

```
def get_items(d):
    items = []
    for key in d:
        items.append((key, d[key]))
    return items
```

```
def key_exists(d, target):
    for key in d:
        if key == target:
            return True
    return False
```

```
def value_exists(d, target):
    for key in d:
        if d[key] == target:
            return True
    return False
```

```
def count_value_occurrences(d, target):
    count = 0
    for key in d:
```

```
        if d[key] == target:
            count += 1
    return count

def add_key_value(d, key, value):
    d[key] = value
    return d

def update_value(d, key, value):
    if key_exists(d, key):
        d[key] = value
    return d

def delete_key(d, target):
    new_dict = {}
    for key in d:
        if key != target:
            new_dict[key] = d[key]
    return new_dict

def merge_dicts(d1, d2):
    merged = {}
    for key in d1:
        merged[key] = d1[key]
    for key in d2:
        merged[key] = d2[key]
    return merged

def dict_from_list(pairs):
    d = {}
    for pair in pairs:
        key, value = pair
        d[key] = value
    return d

def invert_dict(d):
    inverted = {}
    for key in d:
        value = d[key]
        inverted[value] = key
    return inverted

def dict_get(d, key, default=None):
    for k in d:
        if k == key:
```

```
            return d[k]
    return default
```

```
def dict_has_key(d, key):
    for k in d:
        if k == key:
            return True
    return False
```

```
def dict_update(d1, d2):
    for key in d2:
        d1[key] = d2[key]
    return d1
```

```
def dict_copy(d):
    new_d = {}
    for key in d:
        new_d[key] = d[key]
    return new_d
```