

CS136 EXERCISE SOLUTIONS: MODULE 11

GENERIC ALGORITHMS

```
int lastdigit_cmp(const void *v0, const void *v1) {
    const int *i0 = v0;
    const int *i1 = v1;
    int last0 = (*i0) % 10;
    int last1 = (*i1) % 10;
    return last0 - last1;
}
```

```
int alphcmp(const void *a, const void *b) {
    char * const *A = a;
    char * const *B = b;
    return strcmp(*A, *B);
}
```

```
int lengcmp(const void *a, const void *b) {
    char * const *A = a;
    char * const *B = b;
    int diff = strlen(*A) - strlen(*B);
    if (diff) return diff;
    else return strcmp(*A, *B);
}
```

```
int *generic_max(void *p, int len, int elsize, int (*f)(const void*, const void*)) {
    void *best = p;
    for(int i = 0; i < len; ++i) {
        if (f(best, p + elsize * i) < 0) {
            best = p + elsize * i;
        }
    }
    return best;
}
```

GENERIC ARRAY ADT

```
void array_destroy(struct array *s) {
    assert(s);
    for (int i = 0; i < s->len; ++i) {
        s->destroy_item(s->data[i]);
    }
    free(s->data);
    free(s);
}
```

```
void *array_get_at(const struct array *s, int index) {  
    assert(s);  
    assert(0 <= index && index < s->len);  
    return s->data[index];  
}
```

```
void array_print(const struct array *s) {  
    printf("[");  
    for (int i = 0; i < s->len; ++i) {  
        printf(" ");  
        if (s->data[i] == NULL) {  
            printf("NULL");  
        } else {  
            s->print_item(s->data[i]);  
        }  
    }  
    printf(" ]\n");  
}
```