Structures in C

Computing Lab

https://www.isical.ac.in/~dfslab

Indian Statistical Institute

Structures

Definition

A structure is a collection of one or more variables, possibly of different types, grouped together under a single name for convenient handling.

Example:

```
struct point {
    float x; // x and y are called
    float y; // members or fields
} p1, p2;

struct triangle {
    struct point a, b, c;
} t;
```

Operations on structures

- Assignment to members / fields

```
p1.x = 1.0; p1.y = 2.0; t.a.x = 0.0; t.a.y = 0.5;
```

- Assignment / copying of structure variables

```
struct triangle t1, t2;   ...   ; t2 = t1;
```

- Structures may be passed to functions, and returned by functions.
- **But comparison operators (==, !=) don't work!**

Structure operations

- Initialisation

```
struct point {
    float x; // x and y are called
    float y; // members or fields
} p1, p2;

struct triangle {
    struct point a, b, c;
} t = { { 1.0, 1.0 },
        { -1.0, 1.0 },
        { 1.0, -1.0 } };
```

Typedefs

```
typedef unsigned int Length;
Length len, maxlen;
Length lengths[];

typedef char *String;
String p, myStrings[128]; // p - single string, myStrings - array of
    strings
int strcmp(String, String);
p = (String) malloc(100);
```

Typedefs

```
typedef struct {
    float x;
    float y;
} POINT;
POINT p1, p2;

typedef struct {
    struct point a, b, c;
} TRIANGLE;
TRIANGLE t;
```

Pointers to structures

```
struct point *pp; // old scheme (before typedef)

POINT *pp;        // new scheme (after typedef)

(*pp).x = (*pp).y = 0.0; // OR
pp->x = pp->y = 0.0;
```

Memory allocation

```
TRIANGLE *tp;

tp = (TRIANGLE *) malloc(num_triangles * sizeof(TRIANGLE));
```

Structures and functions

- If a large structure is to be passed to a function, it is generally more efficient to pass a pointer than to copy the whole structure.
- Similarly for return values (**but be careful!**)

Review question – slide I

Which of `read_data1`, `read_data2`, `read_data3` is best?

```
typedef struct {
    char name[64];
    int roll, rank;
    float percent;
} STUDENT;

STUDENT *read_data1(void)
{ STUDENT s;
  scanf("%s %d %d %f",
    &(s.name[0]), &(s.roll),
    &(s.rank), &(s.percent));
  return &s;
}
```

Review question – slide II

```
STUDENT read_data2(void)
{ STUDENT s;
  scanf("%s %d %d %f",
    &(s.name[0]), &(s.roll),
    &(s.rank), &(s.percent));
  return s;
}

STUDENT *read_data3(STUDENT *s)
{ scanf("%s %d %d %f",
    &(s->name[0]), &(s->roll),
    &(s->rank), &(s->percent));
  return s;
}
```