



CSCA48

Introduction to Computer Science II

导师 : Vincent

UTSC Week 03 Class | 2025/1/23

Struct

請看上課的例子！！！

Allocation

Example

The C program below manages the user accounts of a credit card company. The program complies without errors, but when run, often creates unexpected (incorrect) outputs. Make changes by printing neatly directly on the code (or beside it) so that the program will work as intended. Change as little as necessary.

```
#include <stdio.h>
#include <string.h>

struct user_account {
    char card_number[9];
    char name[20];
    double balance;
};

void print(struct user_account s) {
    printf("%s: %s\t Account balance: %f\n", s.card_number, s.name, s.balance);
}

void process_xct(struct user_account s, double amount) {
    s.balance = s.balance + amount;
}

int main() {
    // create an account for Vincent Tse (User ID 123456789) with a zero balance

    struct user_account s1;
    strcpy(s1.card_number, "123456789");
    strcpy(s1.name, "Vincent Tse");
    s1.balance = 0.0;

    process_xct(s1, 1000.0);      // Vincent just spent 1000 dollars!

    return 0;
}
```

Example

Complete the code below according to the comments.

```
typedef struct month {
    char name[10];
    int num_days;
    int num_holidays;
    char **holidays;
} Month;

/* Return a value greater than 0 if m1 is longer than m2, less than 0 if m2
 * is longer than m1 and 0 if they have the same number of days
 */
int longer(Month *m1, Month *m2) {
    return m1->num_days - m2->num_days;
}

int main() {

    struct month dec;

    // Set the number of days in dec to 31

    // Set the name of dec to December

    // Allocate space for an array of 2 december holidays
    // and set the holidays to mutable strings "christmas" and "thanksgiving"
```

```
// Change the first letter of each of the holidays to uppercase so they are
// "Christmas" and "Thanksgiving". Do this by changing the strings in place,
// rather than by assigning a completely new string.
```

```
Month *june;
```

```
// Create a second month pointed to by the variable june declared above.
```

```
// Set the number of days to june to 30.
```

```
// Call longer correctly on june and dec so this code works.
```

```
if ( ) {
    printf("December is longer than June.\n");
}
return 0;
}
```

Linked List



- ◆ Let's create a Person Linked List

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_STR_LEN 1024

typedef struct Human {
```

- ◆ Write a function that creates and initializes a Person

```
Person *create_person(
```

- ◆ Insert to head

- ◆ Write a function that insert a new Person at the head of the list.

```
Person *insert_to_head(
```

- ◆ List Person (print)

- ◆ Write a function that prints the Person's name (one Person per line).

```
void print_person()
```

- ◆ Find Person

- ◆ Write a function that finds a Person given the name.

```
Person *find_person()
```

- ◆ Add to tail

- ◆ Write a function that append a new Person at the end of the list.

```
Person *add_to_tail()
```

- ◆ Add to tail (2)

- ◆ Write a function that append a new Person at the end of the list.

```
Person *add_to_tail2(
```

- ◆ Remove head

- ◆ Write a function that removes the first Person in the list.

```
Person *remove_head()
```

- ◆ Remove tail

- ◆ Write a function that removes the last Person in the list.

```
Person *remove_tail()
```

- ◆ Remove Person

- ◆ Write a function that remove the Person by the given name.
`Person *remove_person()`

- ◆ Delete List

- ◆ Write a function that deletes the entire list

```
Person *delete_list()
```

- ◆ Reverse

- ◆ Write a function that reverse the list.

```
Person *reverse(
```