CSE344 - HW 4 – REPORT

In the program, there are 7 threads, 6 for chefs' operations and main thread for wholesaler. Program flow goes as below.

Main() →	start_work()	chef_thread_func() // 6 chefs' threads work in this functions wholesaler_func() // main thread works in this function

Semaphores: *Sems* struct type is defined in *base.h* file to keep track of semaphores and used as a global variable in *program.c* file.

```
typedef struct{
        sem_t dess_ready; // dessert ready for wholesaler to take
        sem_t ingr_ready; // ingredients ready for chefs to take
        sem_t lock_ingr;        // protects new_ings
}Sems;
```

Communication: To manage, ingredient receiving, I prefer to use an int value that have a meaning in terms of binary format.

Also as a reference to ingredients an enum type is defined as fallows,

```
typedef enum {M=1, F=2, W=4, S=8} ingredients;
```

based on this enum, an integer value can hold one piece of four distinct ingredients. In our scenario this usage is sufficient.

To be able to notify the chefs when the delivery is done, `TERMINATE_CHEF` macro is used to make program more readable.

```
#define TERMINATE_CHEF -1
```

Example:

```
int ingr;
ingr = (M | F | W | S); // ingr stores one each milk, flour, walnuts and sugar
ingr = (M | F | W );    // ingr stores one each milk, flour and walnuts
ingr = (M | S);         // ingr stores one each milk and sugar
ingr = (W | S);         // ingr stores one each walnuts and sugar
ingr = (W);             // ingr stores one piece of walnuts
```

Wholesaler side:

- The input file is read, assuming it is valid according to homework instruction text.
- Pseudocode;
  1) read one line from input file
  2) if file has ended, go to step 8
  3) recognize ingredients
  4) Set **new_ingr*
  5) sem_post for *ingr_ready*
  6) sem_wait for *dess_ready*
  7) obtain the dessert and go back step 1
  8) set `TERMINATE_CHEF` to **new_ingr* (notifies chefs that the delivery is done)
  9) sem_post for *ingr_ready* and return

Chef Side:

Wait for wholesaler to brings ingredients
Observe ingredients
if ingredients value indicates `TERMINATE_CHEF` return
if they match take it and prepare dessert and come back to wait new ingredients
if ingredients do not complete the recipe, let other chefs have a look to them

Pseudocode;

1) Initialize infinite ingredients and chef index
2) Detect absence ingredients
3) sem_wait for *ingr_ready*
4) if **new_ingr* is `TERMINATE_CHEF` go to step 9
5) if the **new_ingr* does not complete the recipe go to step 8
6) prepare the dessert
7) sem_post for *dess_ready* and go back step 3
8) sem_post for *ingr_ready* and go back step 3
9) sem_post for *ingr_ready* and return