

# **Inside Recipe**



## **13016235: C Programming**

**First Semester, 2019**

Software Engineering Program

Faculty of Engineering, KMITL

Tanachat Mongkolporn

62011258

# Proposal

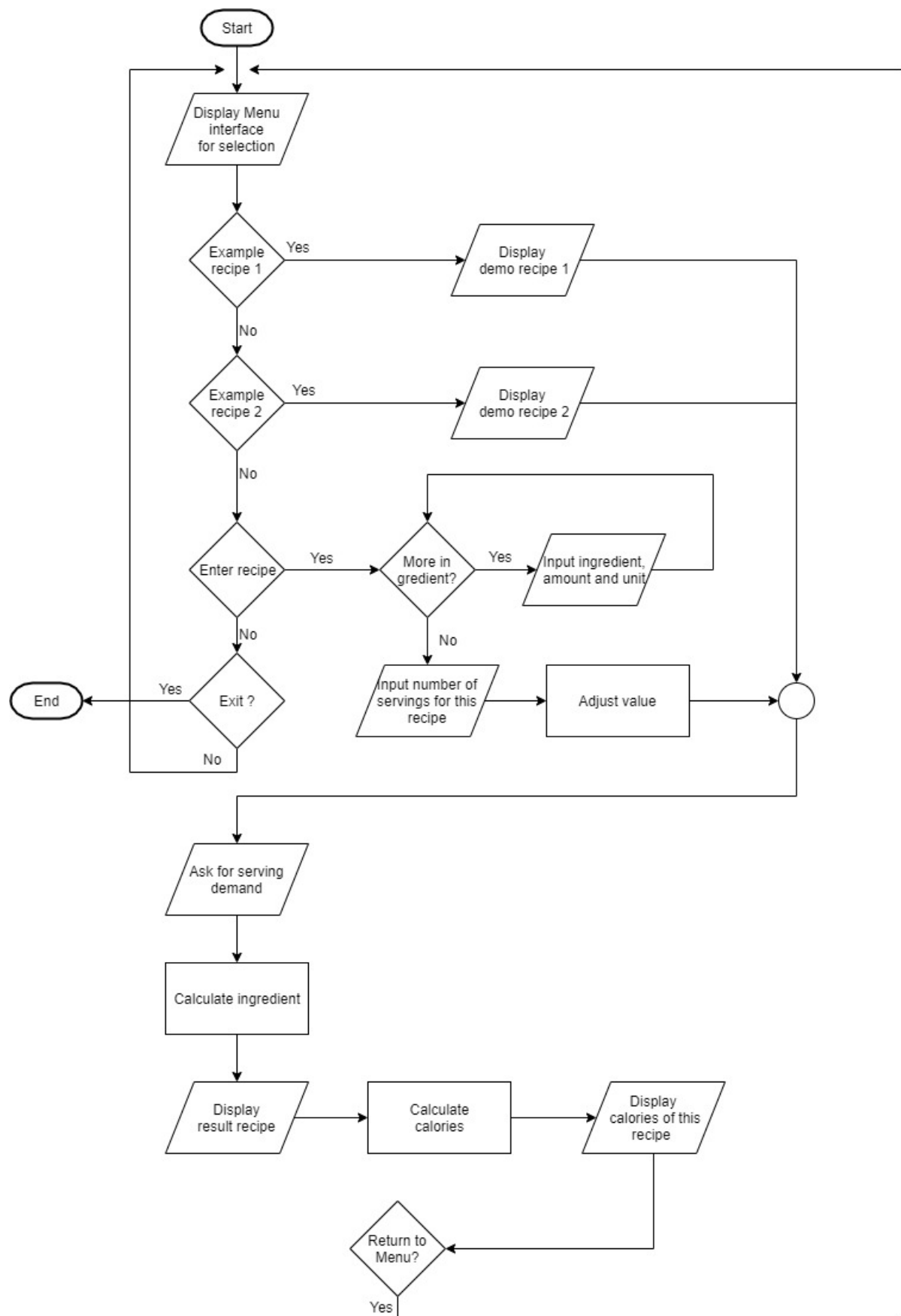
Food is one of the main subjects for living. In our daily life, if we decide to cook, the first thing to keep in mind is what we will cook and what recipe will be used. Once we pick a recipe, we then start to prepare it.

Since we aren't cooking in a professional kitchen, situation that we don't have the required equipment to measure the ingredients according to the recipe happens very often that we would need extra help. The recipe given from searching does not fit for what we want and calculation needed for this particular task is rather difficult for someone who just wanted to cook. It consumes preparation time and effort for cooking just one menu.

Each ingredient: liquids, solids, grains, butter or powders had different densities thus have different scales and different proportion required for calculation. It is hard enough to acquire all the measuring equipment, it is impossible to memorize the proportion and measurements equivalents. For someone who is not good at mathematics, just adjusting the serving size from what the recipe cooked for to something they wanted is hard enough. Adjusting the units of each ingredient to what measuring equipment they have seemed impossible to them.

This project: 'Inside recipe', will conveniently solve all the cooking calculation and finish the task in one place. This includes adjusting the units of the ingredients to a standard measuring unit and calculate the ingredients you needed according to the serving size you wanted. Additionally, calories of food will be given for information. Ingredients this project will scope to provides for dessert.

## Flow Chart



Code:

```
#define _CRT_SECURE_NO_WARNINGS //Set line ending LF(UNIX)
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <Windows.h>
#define max_I 50 //maximum ingredients for recipe
#define maxCharLen 100 //maximum food character length
#define CLEAR "\033[0;0H\033[2J"
static int ingredient = 0;

struct ing {
    char name[maxCharLen];
    char unit[maxCharLen];
    float amount;
};

struct Eunit {
    char* unit;
    double trans;
};

void proportionStruct(struct ing* value, double n){
    int count = 0;
    while(value[count].amount != 0)
    {
        value[count].amount = (value[count].amount)*n;
        count++;
    }
}

void make_lower(struct ing *word)
{
    for(int i = 0; i < ingredient; i++){
        tolower(word[i].name);
    }
}

void convertUnits(struct ing* item){
    struct Eunit u[15];
    u[0].unit = "Tablespoon, tablespoon, tbs, tbsp, T" ;
    u[1].unit = "Teaspoon, teaspoon, tsp, t";
    u[2].unit = "cup, c, C";
    u[3].unit = "pint, pt";
    u[4].unit = "quart, qt";
    u[5].unit = "milliliter, ml, mL, millilitre";
    u[6].unit = "liter, litre, L, l";
    u[7].unit = "deciliter, dl, dL";
    u[8].unit = "gallon, gal";
    u[9].unit = "pinch";
    u[10].unit = "milligram, mg";
    u[11].unit = "kilogram, kg";
    u[12].unit = "ounce, oz";
    u[13].unit = "gram, g";
```

```

u[14].unit = "pound, lb";
u[0].trans = 15;
u[1].trans = 5;
u[2].trans = 250;
u[3].trans = 473;
u[4].trans = 946;
u[5].trans = 1;
u[6].trans = 1000;
u[7].trans = 100;
u[8].trans = 3800;
u[9].trans = 1;
u[10].trans = 0.001;
u[11].trans = 1000;
u[12].trans = 28;
u[13].trans = 1;
u[14].trans = 454;
int count = 0;
char temp[50] = { 0 }, *pt;
while(item[count].amount != 0) //do until end of the ingredient list
{
    int a = 0, check = 1;
    while(a < 15 && check) //sort from unit list
    {
        strcpy(temp, u[a].unit);
        pt = strtok(temp, ",");
        while(pt != NULL && check)
        {
            if(strcmp(item[count].unit, pt) == 0)
            {
                item[count].amount = (item[count].amount) * (u[a].trans);
                if(a < 9)
                {
                    strcpy(item[count].unit, "ml");
                }
                else if(a > 9)
                {
                    strcpy(item[count].unit, "g");
                }
                check = 0; //convert unit already check to go to
next element
                //goto NEXT;
            }
            pt = strtok(NULL, ",");
        }
        a++;
    }
    //NEXT: count++;
    count++;
}
//toConvert = unit to convert, value = amount of its
void convertUnit(char toConvert[max_I][21], float value[max_I], struct Eunit *converse){
    int num = 0, count = 0;
    char *p;
    while(toConvert[count] != NULL) //end of unit list
    {

```

```

        p = strtok(converse[num].unit, ",");
        while(p != NULL)
        {
            if(strcmp(toConvert[num], p)==0){
                printf("%s\n", p);
                p = strtok(NULL, ",");
            }
        }
        num++;
        count++;
    }
}

void getCalorie(structing* item){
    FILE* fptr;
    double calorie = 0, temp=0;
    char word[100], line[100];
    char* p;
    int check;
    if((fptr = fopen("ProjectC_cal.txt", "r")) != NULL){
        for(int i = 0; i < ingredient; i++)//go through all ingredient
        {
            check = 0;
            fseek(fptr, 0, SEEK_SET);
            while(fgets(line, 100, fptr))
            {
                if(strncmp(item[i].name, line, strlen(item[i].name))==0)
                {
                    check = 1;
                    strcpy(word, line);
                    p = strtok(word, ",");
                    p = strtok(NULL, ",");
                    temp = strtod(p, NULL);
                    calorie += (temp*item[i].amount);
                    break;
                }
            }
            if(check==0)
            {
                printf("\nUnknown calories for '%s'", item[i].name);
            }
        }
        printf("\nCalories(from gml): %.2lf", calorie);

        fclose(fptr);
    }
    else
    {
        printf("Error!No calories file found");
    }
}

void printMenu(structing* menu){
    int count = 0;
    printf("Recipe\n-----\n");
    while(menu[count].amount != 0)
    {
        printf("%-20s%-2f %s\n", menu[count].name, menu[count].amount, menu[count].unit);
        count++;
    }
}

```

```

    }
}
void SetPosition(COORD Position){
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Position);
}
COORD GetPosition(){
    HANDLE h =GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_SCREEN_BUFFER_INFO bufferInfo;
    GetConsoleScreenBufferInfo(h, &bufferInfo);
    return bufferInfo.dwCursorPosition;
}
void enterRecipe(){
    structing f[max_I]={ 0 };
    ingredient =0;
    float temp =0;
    char ans;
    printf("****Press double enter to submit recipe***\n");
    printf("Ingredient\t\tAmount\t\tUnit\n");
    do
    {
        printf(" ");
        gets(f[ingredient].name);
        if (f[ingredient].name[0]==NULL)
        {
            break;
        }
        printf("\033[1A\t\t\t");
        if (scanf("%f", &temp)&& temp !=0){
            f[ingredient].amount =temp;
        }
        else
        {
            printf("Invalid input");
            exit(1);
        }
        printf("\033[1A\t\t\t\t");
        scanf("%s", f[ingredient].unit);
        getchar();
        ingredient++;
    } while(1);
    int serve;
    printf("Enter how much is this recipe for:");
    scanf("%d", &serve);
    proportionStruct(f, 10 /serve);

    printf("Enter how much serve do you want:");
    scanf("%d", &serve);
    proportionStruct(f, serve);
    make_lower(f);
    convertUnits(f);

    printf(CLEAR);
    printMenu(f);
    getCalorie(f);
    printf("\nEnter 'y' return to menu:");
    while (scanf("%c", &ans)==0 || tolower(ans)!='y')

```

```

    {
        printf("\n\nEnter 'y' return to menu:");
    }
}
void main0{

    char menu_select;
    float temp = 0;
    char ans;
    int serve;
    COORD current = { 0,0 };
    while(1)
    {
        structing example1[11]={ {"flour","cup",1},{ "salt","teaspoon",1},{ "baking
soda","teaspoon",0.5},
                                {"baking
powder","teaspoon",0.5},{ "brown sugar","tablespoon",4},{ "milk","cup",0.75},
                                {"egg","gram",60},{ "vegetable oil","tablespoon",1},{ "vanilla extract","teaspoon",1},
                                {"semisweet
chocolate","cup",0.5} };
        structing example2[6]={ {"bread flour","cup",3},{ "active dry
yeast","tsp",1},{ "salt","tsp",2},
                                {"sugar","tsp",1},{ "water","cup",1} };

        printf("-----\n");
        printf("--    INSIDE RECIPE    --\n");
        printf("-----\n\n");
        printf(" 1.Example recipe 1\n\n");
        printf(" 2.Example recipe 2\n\n");
        printf(" 3.Enter recipe\n\n");
        printf("Press q to exit :");
        scanf("%C",&menu_select);
        getchar();
        switch(tolower(menu_select))
        {
            case '1':
                printf(CLEAR);
                printMenu(example1);
                ingredient = 10;
                printf("\nEnter how much serve do you want:");
                scanf("%d", &serve);
                proporionStruct(example1, serve);
                convertUnits(example1);
                printf(CLEAR);
                printMenu(example1);
                getCalorie(example1);
                printf("\nEnter 'y' return to menu:");
                while(scanf("%C", &ans)==0 || tolower(ans)!='y')
                {
                    printf("\n\nEnter 'y' return to menu:");
                }
                break;
            case '2':
                printf(CLEAR);

```



```

        printMenu(example2);
        ingredient = 6;
        printf("\nEnter how much serve do you want:");
        scanf("%d", &serve);
        proportionStruct(example2, serve);
        convertUnits(example2);
        printf(CLEAR);
        printMenu(example2);
        getCalorie(example2);
        printf("\nEnter 'y' return to menu:");
        while(scanf("%C", &ans)==0 || tolower(ans)!='y')
        {
            printf("\n\nEnter 'y' return to menu:");
        }
        break;
    case '3':
        printf(CLEAR);
        enterRecipe();
        break;
    case 'q':
        exit(1);
    default:
        break;
};
system("cls");
}
}

```

```
D:\University\C_project_y1\Inside_recipe\Debug\Inside_recipe.exe
-----
--          INSIDE RECIPE          --
-----

1.Example recipe 1
2.Example recipe 2
3.Enter recipe

Press q to exit :
```

```
D:\University\C_project_y1\Inside_recipe\Debug\Inside_recipe.exe
Recipe
-----
flour          1.00 cup
salt           1.00 teaspoon
baking soda    0.50 teaspoon
baking powder  0.50 teaspoon
brown sugar    4.00 tablespoon
milk           0.75 cup
egg            60.00 gram
vegetable oil  1.00 tablespoon
vanilla extract 1.00 teaspoon
semisweet chocolate 0.50 cup

Enter how much serve do you want:
```

```
D:\University\C_project_y1\Inside_recipe\Debug\Inside_recipe.exe
Recipe
-----
flour          500.00 ml
salt           10.00 ml
baking soda    5.00 ml
baking powder  5.00 ml
brown sugar    120.00 ml
milk           375.00 ml
egg            120.00 g
vegetable oil  30.00 ml
vanilla extract 10.00 ml
semisweet chocolate 250.00 ml

Calories(from g/ml): 4133.30
Enter 'y' return to menu:
```