←!—

- @Author: JIANG Yilun
- @Date: 2022-04-24 14:28:58
- @LastEditTime: 2022-05-08 14:10:09
- @LastEditors: ThearchyHelios
- @Description: Rapport du projet Cowsay
- @FilePath: /Projet_cowsay_L1S2/rapport_cowsay_JIANGYilun.md
  —→

Projet Cowsay

JIANG Yilun

Sommaire

1. Présentation du Projet `cowsay`

Le projet débute au premier jour du cours INF203 et s'achève lors la dernière semaine de cours. Cette dernière fait date de rendu (dimanche soir minuit de la dernière semaine). Vous pouvez progresser sur le projet a votre rythme, mais nous vous recommandons de prendre de l'avance par rapport au cours, du moins aucun retard. Par exemple, la partie “Bash” devra être

achevée au moment ou les premiers cours de “C” débuteront.

2. Objectif du projet

L’objectif du projet est de découvrir le monde merveilleux de “cowsay”. Au cours du projet, vous réaliserez les objectifs suivants:

1. Préliminaires

Découvrir la commande cowsay a travers son manuel (manpage) et l’ensemble des options qu’elle contient.

2. Bash

Implémenter un script Bash qui fait réciter a la vache la suite des nombres premiers, des nombres de Fibonacci, ou toute autre suite exotique de votre choix.

3. C

Recoder cowsay en C, avec de nouvelles fonctionnalités additionnelles de votre choix (comme par exemple la longueur de la queue).

4. Automates

En s’appuyant sur la théorie des automates, implementer un “cow-Tamagoshi” qu’il s’agit de nourrir et faire survivre aussi longtemps que possible.

Préliminaires

Découvrir du code `cowsay` :

```
1 cowsay -h
```

Nous avons donc le résultat suivant:

```
1 cow{say,think} version 3.03, (c) 1999 Tony Monroe
2 Usage: cowsay [-bdgpstwy] [-h] [-e eyes] [-f cowfile]
3           [-l] [-n] [-T tongue] [-W wrapcolumn] [message]
```

Nous apprenons donc que le projet cowsay contient en fait deux commandes, l'une appelée Cowsay et l'autre Cowthink. Cowsay utilise des lignes droites pour relier la vache aux mots prononcés, tandis que cowthink utilise des cercles.

Par exemple, nous utilise `cowsay` en première:

```
1 cowsay "Hello, my name is JIANG Yilun"
```

Avec les résultats suivant:

```
 _______________________________
< Hello, my name is JIANG Yilun >
 -------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

Ensuit, nous utilise le commande `cowthink` :

```
cowthink "Hello, my name is JIANG Yilun"
```

Avec les résultats suivant:

```
 _______________________________
( Hello, my name is JIANG Yilun )
 -------------------------------
        o   ^__^
         o  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

En fait, le cowsay ne se limite pas à la forme de la vache. Apres nous utilisons le commande `cowsay -l` , nous pouvons constater que nous avons en fait de nombreux modèles à choisir:

```
$ cowsay -l
Cow files in /opt/homebrew/Cellar/cowsay/3.04_1/share/cows:
beavis.zen blowfish bong bud-frogs bunny cheese cower daemon default dragon
dragon-and-cow elephant elephant-in-snake eyes flaming-sheep ghostbusters
head-in hellokitty kiss kitty koala kosh luke-koala meow milk moofasa moose
mutilated ren satanic sheep skeleton small stegosaurus stimpy supermilker
surgery three-eyes turkey turtle tux udder vader vader-koala www
```

Par exemple, on peut utilise la forme `sheep` :

```
$ cowsay -f sheep hello
 _______
< hello >
 -------
  \
   \
        __
       UooU\.'@@@@@@`.
       \__/(@@@@@@@@@@)
            (@@@@@@@@)
            `YY~~~~YY'
             ||    ||
```

On peut aussi utiliser le vase avec des tuyaux:

```
$ ll | cowsay
 ________________________________________
/ total 8 -rw-r--r--@ 1 yilunjiang staff \
| 3.6K Apr 24 15:03                      |
\ rapport_cowsay_JIANGYilun.md           /
 ----------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

En fait, la sortie de cowsay est très pauvre - pratiquement impossible à visualiser très bien. Mais si nous ajoutons la commande -n:

```
$ ll | cowsay -n
 ________________________________________________________________________________
/ total 16                                                                       \
\ -rw-r--r--@ 1 yilunjiang  staff   4.2K Apr 24 15:09 rapport_cowsay_JIANGYilun.md /
 --------------------------------------------------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

De cette façon, les informations décrites peuvent être lues de manière plus visuelle.

La commande cowsay a en fait ces petits extras, par exemple, nous pouvons changer les yeux de la vache:

```
$ cowsay -e -- "Hello, my name is JIANG Yilun"
 _______________________________
< Hello, my name is JIANG Yilun >
 -------------------------------
        \   ^__^
         \  (--)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

Nous avons même réussi à lui faire cracher sa langue:

```
$ cowsay -T U "Hello, my name is JIANG Yilun"
 _______________________________
< Hello, my name is JIANG Yilun >
 -------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
             U ||----w |
                ||     ||
```

Bash

En fait, le code présenté ci-dessous a été modifié une deuxième fois (après avoir vu la vache folle) et comporte deux sections distinctes : une avec un argument et une sans.

cow_kindergarten

```
###
 # @Author: JIANG Yilun
 # @Date: 2022-04-24 15:15:21
 # @LastEditTime: 2022-04-24 17:55:56
 # @LastEditors: JIANG Yilun
 # @Description:
 # @FilePath: /Projet_cowsay_L1S2/cow_kindergarten.sh
###

if [ $# -eq 0 ]; then
    temp=10
    while [ $temp -gt 0 ]; do
        clear
        cowsay $temp
        sleep 1
        temp=$((temp-1))
    done
else
```

```
    temp=$1
    while [ $temp -gt 0 ]; do
        clear
        cowsay $temp
        sleep 1
        temp=$((temp-1))
    done
fi
```

cow_primaryschool

```
###
 # @Author: JIANG Yilun
 # @Date: 2022-04-24 15:33:12
 # @LastEditTime: 2022-04-24 17:54:01
 # @LastEditors: JIANG Yilun
 # @Description:
 # @FilePath: /Projet_cowsay_L1S2/cow_primaryschool.sh
###

i=1
if [ $# -eq 0 ]; then
    echo "Saissez un nombre:"
    read nombre
    while [ $i -le $nombre ]; do
        clear
        cowsay $i
        sleep 1
        i=$((i+1))
    done
else
    nombre=$1
    while [ $i -le $nombre ]; do
        clear
        cowsay $i
        sleep 1
        i=$((i+1))
    done
fi
```

cow_highschool

```
###
 # @Author: JIANG Yilun
 # @Date: 2022-04-24 15:37:56
 # @LastEditTime: 2022-04-24 17:52:24
 # @LastEditors: JIANG Yilun
 # @Description:
 # @FilePath: /Projet_cowsay_L1S2/cow_highschool.sh
###

i=1

if [ $# -eq 0 ]; then
    echo "Saissez un nombre:"
    read nombre
    while [ $i -le $nombre ]; do
        clear
        cowsay $(($i*$i))
        sleep 1
        i=$((i+1))
    done
else
    nombre=$1
    while [ $i -le $nombre ]; do
        clear
        cowsay $(($i*$i))
        sleep 1
        i=$((i+1))
    done
fi
```

cow_college

```
###
 # @Author: JIANG Yilun
 # @Date: 2022-04-24 15:41:00
 # @LastEditTime: 2022-04-24 17:44:13
 # @LastEditors: JIANG Yilun
 # @Description:
 # @FilePath: /Projet_cowsay_L1S2/cow_college.sh
###

# nombres de Finonacci

i=0
j=1
```

```
14
15 if [ $# -eq 0 ]; then
16     echo "Saissez un nombre:"
17     read nombre
18     while [ $j -lt $nombre ]; do
19         cowsay $j
20         temp=$((i+j))
21         i=$j
22         j=$temp
23         sleep 1
24     done
25 else
26     nombre=$1
27     while [ $j -lt $nombre ]; do
28         cowsay $j
29         temp=$((i+j))
30         i=$j
31         j=$temp
32         sleep 1
33     done
34 fi
```

cow_university

```
1  ###
2   # @Author: JIANG Yilun
3   # @Date: 2022-04-24 15:55:25
4   # @LastEditTime: 2022-04-24 17:42:31
5   # @LastEditors: JIANG Yilun
6   # @Description:
7   # @FilePath: /Projet_cowsay_L1S2/cow_university.sh
8  ###
9
10 nbr_premier() {
11     while [ $i -le $m ]
12     do
13         p=$(($m%$i))
14         if [ $p -eq 0 ]
15         then
16             break
17         else
18             i=$((i+1))
19         fi
20         if [ $i -eq $m ]
21         then
22             if [ $m -eq $n ]
23             then
24                 echo "$m est un nombre premier"
```

```
                cowsay -T U "$m"
            else
                echo "$m est un nombre premier"
                cowsay "$m"
            fi
        fi
    done
}

if [ $# -eq 0 ]; then
    echo "donnez le dernier nombres premiers à calculer"
    read n
    i=2     #le premier nombre premier
    a=$(bc <<< "scale=0; sqrt($n)")  #scale=0 n'affiche pas les décimale, scale=1 la première, etc... sqrt() calcule la racine carré. marche grace à la commande bc
    m=3
    echo "voici sa suite de nombres premiers de $i à $n"
    while [ $m -le $n ]
    do
        echo m:$m
        i=2
        nbr_premier $m
        m=$((m+1))
        sleep 1
    done
else
    n=$1
    i=2     #le premier nombre premier
    a=$(bc <<< "scale=0; sqrt($n)")  #scale=0 n'affiche pas les décimale, scale=1 la première, etc... sqrt() calcule la racine carré. marche grace à la commande bc
    m=3
    echo "voici sa suite de nombres premiers de $i à $n"
    while [ $m -le $n ]
    do
        echo m:$m
        i=2
        nbr_premier $m
        m=$((m+1))
        sleep 1
    done
fi
```

smart_cow

```
###
 # @Author: JIANG Yilun
 # @Date: 2022-04-24 16:27:30
 # @LastEditTime: 2022-04-24 16:40:09
```

```
 # @LastEditors: JIANG Yilun
 # @Description:
 # @FilePath: /Projet_cowsay_L1S2/smart_cow.sh
###


if [ $# -eq 0 ]; then
    echo "Donner l'expression à calculer:"
    read expression
    cowsay -e $(echo "$expression" | bc) $expression
else
    cowsay -e $(echo "$1" | bc) $1
fi
```

crazy_cow

```
###
 # @Author: JIANG Yilun
 # @Date: 2022-04-24 16:44:04
 # @LastEditTime: 2022-04-24 17:57:02
 # @LastEditors: JIANG Yilun
 # @Description:
 # @FilePath: /Projet_cowsay_L1S2/crazy_cow.sh
###

for var in "$@"
do
    if [[ "$var" == "-h" || "$var" == "--help" ]]; then
        echo "Usage: $0 [OPTION]... [FILE]..."
        echo "Print a crazy cow."
    elif [[ "$var" == "-v" || "$var" == "--version" ]]; then
        echo "crazy_cow.sh version 1.0"
    elif [[ "$var" == "-a" || "$var" == "--addition" ]]; then
        sh cow_primaryschool.sh ${@: -1}
    elif [[ "$var" == "-c" || "$var" == "--countdown" ]]; then
        sh cow_kindergarten.sh ${@: -1}
    elif [[ "$var" == "-s" || "$var" == "--square" ]]; then
        sh cow_highschool.sh ${@: -1}
    elif [[ "$var" == "-f" || "$var" == "--finonacci" ]]; then
        sh cow_college.sh ${@: -1}
    elif [[ "$var" == "-p" || "$var" == "--premiere" ]]; then
        sh cow_university.sh ${@: -1}
    elif [[ "$var" == "-S" || "$var" == "--smart" ]]; then
        sh smart_cow.sh ${@: -1}
    fi
done
```

C

Question 1

`affiche_vache` :

```c
int affiche_vache()
{
    printf("\n");
    printf("    \\   ^__^\n");
    printf("     \\  (oo)\\_______\n");
    printf("        (__)\\       )\\/\\\n");
    printf("            ||----w |\n");
    printf("            ||     ||\n");
    printf("\n");
    return 0;
}

int main()
{
    affiche_vache();
}
```

Après la compilation, nous avons pu obtenir les résultats suivants:

```
$ gcc newcow.c && ./a.out


   \   ^__^
    \  (oo)\_______
       (__)\       )\/\
           ||----w |
           ||     ||
```

Question 2

```c
/*
 * @Author: JIANG Yilun
 * @Date: 2022-04-24 18:07:27
 * @LastEditTime: 2022-04-24 18:44:29
 * @LastEditors: JIANG Yilun
 * @Description:
 * @FilePath: /Projet_cowsay_L1S2/newcow.c
 */
```

```c

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

int affiche_vache (char *eyes, char *tongue)
{
    if (eyes == NULL && tongue == NULL){
        printf("\n");
        printf("    \\   ^__^\n");
        printf("     \\  (oo)\\_______\n");
        printf("        (__)\\       )\\/\\\n");
        printf("            |\u2500\u2500---w |\n");
        printf("            ||     ||\n");
        printf("\n");
        return 0;
    }
    else if (eyes == NULL && tongue \u2260 NULL){
        printf("\n");
        printf("    \\   ^__^\n");
        printf("     \\  (oo)\\_______\n");
        printf("        (__)\\       )\\/\\\n");
        printf("         %s |\u2500\u2500---w |\n",tongue);
        printf("            ||     ||\n");
        printf("\n");
        return 0;
    }
    else if (eyes \u2260 NULL && tongue == NULL){
        printf("\n");
        printf("    \\   ^__^\n");
        printf("     \\  (%s)\\_______\n",eyes);
        printf("        (__)\\       )\\/\\\n");
        printf("            |\u2500\u2500---w |\n");
        printf("            ||     ||\n");
        printf("\n");
        return 0;
    }
    else
    {
        printf("\n");
        printf("    \\   ^__^\n");
        printf("     \\  (%s)\\_______\n",eyes);
        printf("        (__)\\       )\\/\\\n");
        printf("         %s |\u2500\u2500---w |\n",tongue);
        printf("            ||     ||\n");
        printf("\n");
        return 0;
    }
}

int main (int argc, char *argv[])
{
    char *eyes = NULL;
```

```
    char *tongue = NULL;
    char *message = NULL;
    char *tail = NULL;
    for (int i = 1; i < argc; i++)
    {
        if (strcmp(argv[i], "-e") == 0 || strcmp(argv[i], "--eyes") == 0)
        {
            eyes = argv[i+1];
        }
        if (strcmp(argv[i], "-t") == 0 || strcmp(argv[i], "--tongue") == 0)
        {
            tongue = argv[i+1];
        }
    }
    affiche_vache(eyes, tongue);
}
```

Après la compilation, nous avons pu obtenir les résultats suivants:

```
$ gcc test.c && ./a.out -e AA -t U

   \   ^__^
    \  (AA)\_______
       (__)\       )\/\
        U |├ ---w |
          ||     ||
```

Question 3

```
/*
 * @Author: JIANG Yilun
 * @Date: 2022-04-24 18:07:27
 * @LastEditTime: 2022-04-24 21:10:24
 * @LastEditors: JIANG Yilun
 * @Description:
 * @FilePath: /Projet_cowsay_L1S2/newcow.c
 */

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

int affiche_vache(int *length, char *message, char *eyes, char *tongue, int *tail)
{
    printf(" -");
    for (int i = 0; i < *length; i++)
    {
```

```
        printf("-");
    }
    printf("\n");
    printf("< %s >\n", message);
    printf(" -");
    for (int i = 0; i < *length; i++)
    {
        printf("-");
    }
    printf("\n");
    printf("    \\   ^__^\n");
    printf("     \\  (%s)\\_______\n", eyes);
    printf("        (__)\\       )\\/\\");
    for (int i = 0; i < *tail; i++)
    {
        printf("/\\");
    }
    printf("\n");
    printf("         %s ||----w |\n", tongue);
    printf("            ||     ||\n");
    printf("\n");
    return 0;
}

void update() { printf("\033[H\033[J"); }

void gotoxy(x, y) { printf("\033[%d;%dH", x, y); }

int main(int argc, char *argv[])
{
    char *eyes = "oo"; // default eyes
    char *tongue = "  "; // default tongue
    char *message = "--help to display help"; // default message
    int tail = 1; // default tail
    for (int i = 1; i < argc; i++)
    {
        if (strcmp(argv[i], "-e") == 0 || strcmp(argv[i], "--eyes") == 0)
        {
            eyes = argv[i + 1];
        }
        if (strcmp(argv[i], "-T") == 0 || strcmp(argv[i], "--tongue") == 0)
        {
            tongue = argv[i + 1];
        }
        if (strcmp(argv[i], "-m") == 0 || strcmp(argv[i], "--message") == 0)
        {
            message = argv[i + 1];
        }
        if (strcmp(argv[i], "-t") == 0 || strcmp(argv[i], "--tail") == 0)
        {
            tail = atoi(argv[i + 1]);
        }
```

```
        if (strcmp(argv[i], "-h") == 0 || strcmp(argv[i], "--help") == 0)
        {
            printf("\n");
            printf("Usage: newcow [OPTION]...\n");
            printf("\n");
            printf("Options:\n");
            printf("  -e, --eyes=STRING  eyes of the cow (default: oo)\n");
            printf("  -t, --tongue=STRING tongue of the cow (default:  )\n");
            printf("  -m, --message=STRING message to display (default: none)\n");
            printf("  -h, --help          display this help and exit\n");
            printf("\n");
            return 0;
        }
    }
    int length = strlen(message) + 1;
    affiche_vache(&length, message, eyes, tongue, &tail);
}
```

On peut ajouter d'argument "eyes" ou argument "tongue".

S'il n'y a pas de message d'entrée:

```
$ gcc newcow.c && ./a.out -e AA -T U
 ------------------------
< --help to display help >
 ------------------------
    \   ^__^
     \  (AA)\_______
        (__)\       )\/\
         U |├ ---w |
           ||     ||
```

Si je veux obtenir des informations d'aide:

```
$ gcc newcow.c && ./a.out -h

Usage: newcow [OPTION]...

Options:
  -e, --eyes=STRING  eyes of the cow (default: oo)
  -t, --tongue=STRING tongue of the cow (default:  )
  -m, --message=STRING message to display (default: none)
  -h, --help          display this help and exit
```

Bien sûr, la possibilité d'afficher des messages est essentielle:

```
$ gcc newcow.c && ./a.out -e AA -T UU -m "Hello, my name is JIANG Yilun"
 -------------------------------
< Hello, my name is JIANG Yilun >
 -------------------------------
    \   ^__^
     \  (AA)\_______
        (__)\       )\/\
         UU |+-ww-w |
            ||     ||
```

En même temps, nous pouvons définir la longueur du tail:

```
$ gcc newcow.c && ./a.out -e AA -T UU -m "Hello, my name is JIANG Yilun" -t 10
 -------------------------------
< Hello, my name is JIANG Yilun >
 -------------------------------
    \   ^__^
     \  (AA)\_______
        (__)\       )\/\/\/\/\/\/\/\/\/\/\
         UU |+-ww-w |
            ||     ||
```

Question 5

```c
/*
 * @Author: JIANG Yilun
 * @Date: 2022-04-25 13:34:08
 * @LastEditTime: 2022-04-26 17:43:29
 * @LastEditors: JIANG Yilun
 * @Description:
 * @FilePath: /Projet_cowsay_L1S2/reading_cow.c
 */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>

#define MAX_LENGTH 512

void affiche_vache(int *length, char *message, char *eyes, char *tongue, int *tail)
{
    printf(" -");
    for (int i = 0; i < *length; i++)
    {
        printf("-");
```

```c
    }
    printf("\n");
    printf("< %s >\n", message);
    printf(" -");
    for (int i = 0; i < *length; i++)
    {
        printf("-");
    }
    printf("\n");
    printf("    \\   ^__^\n");
    printf("     \\  (%s)\\_______\n", eyes);
    printf("        (__)\\       )\\/\\");
    for (int i = 0; i < *tail; i++)
    {
        printf("/\\");
    }
    printf("\n");
    printf("         %s ||----w |\n", tongue);
    printf("            ||     ||\n");
    printf("\n");
}

void update() { printf("\033[H\033[J"); }

void gotoxy(x, y) { printf("\033[%d;%dH", x, y); }

int main(int argc, char *argv[])
{
    FILE *ficher = NULL;
    ficher = fopen(argv[1], "r");
    if (ficher == NULL)
    {
        printf("Error opening file\n");
        return 1;
    }
    else
    {
        char *eyes = "oo";   // default eyes
        char *tongue = "  "; // default tongue
        char message[MAX_LENGTH] = "";  // default message
        int tail = 1;        // default tail
        int length = 0;
        char c;
        while ((c = fgetc(ficher)) != EOF)
        {
            length++;
            affiche_vache(&length, message, eyes, &c, &tail);
            sleep(1);
            update();
            message[length - 1] = c;
            message[length] = '\0';
        }
```

```
        fclose(ficher);
        length++;
        affiche_vache(&length, message, eyes, tongue, &tail);
    }
}
```

Dans le ficher `mot` :

`bonjour, je m'appelle JIANG Yilun`

```
$ gcc reading_cow.c && ./a.out mot

 ---------
< bonjour >
 ---------
    \   ^__^
     \  (oo)\_______
        (__)\       )\/\
          |+---w |
            ||     ||

 -----------------------------------
< bonjour, je m'appelle JIANG Yilun >
 -----------------------------------
    \   ^__^
     \  (oo)\_______
        (__)\       )\/\
            |+---w |
            ||     ||

```

La vache épellera le contenu du document un mot à la fois et mettra le caractère suivant dans sa tongue.

Dans ce fichier, j'ai également utilisé `affiche_vache` pour générer la apparition de la vache.

Automates

Je n'ai pas réussi à concevoir un automate, car je ne pense pas avoir identifié toutes les possibilités. Mais j'ai créé un mini-jeu "CowSay" qui n'utilise pas d'automate.

Les joueurs doivent répondre à des questions de mathématiques pour obtenir de la nourriture, avec une autre chance d'obtenir de la nourriture toutes les heures.

Ce jeu comporte cinq modèles mathématiques : déterminer si un nombre est premier ou non, ajouter, soustraire, multiplier et trouver le mod entre des nombres.

De plus, des événements aléatoires se produiront dans le scénario du jeu et le joueur peut obtenir des effets négatifs, tels que la perte de nourriture ou de life. Ou vous pouvez activer l'effet "ange", qui vous apporte de la nourriture.

```c
/*
 * @Author: JIANG Yilun
 * @Date: 2022-04-25 15:51:26
 * @LastEditTime: 2022-05-02 14:32:41
 * @LastEditors: JIANG Yilun
 * @Description:
 * @FilePath: /Projet_cowsay_L1S2/Tamagoshi-vache.c
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>

#define MAX_LENGTH 512

/*
 * @description: Update(refresh) the Terminal
 * @param {type}: void
 * @return: void
 */
void update() { printf("\033[H\033[J"); }

/*

* @description: Make the pointer to x, y in Terminal
* @param {type}: int x, int y
* @return: void
*/
void gotoxy(x, y) { printf("\033[%d;%dH", x, y); }

int life = 5; // Define the valeur initial to 5

/*

* @description: Print the etat of the cow
* @param {type}: int life
* @return: void
*/
void etat(int life)
{
```

```c
    if (life == 0 || life == 10)
    {
        printf("byebyelife");
    }
    else if (life ⩽ 3 && life ⩾ 1 || life ⩽ 9 && life ⩾ 7)
    {
        printf("lifesucks");
    }
    else if (life ⩽ 6 && life ⩾ 4 || life)
    {
        printf("liferocks");
    }
}

/*

* @description: Print the cow
* @param {type}: int *length_vache(for define the length of divise), char *message_vache(To print the message), char *eyes_vache(To print the cow's eyes(in default: "oo")), char *tongue_vache(To print the cow's tongue), int *tail_vache(To print the cow's tail(This is also the life of cow)), int time_tick_vache(To define how much time the anmie will take), int hour, int minite, int food
* @return: void
*/
void affiche_vache(int *length_vache, char *message_vache, char *eyes_vache, char *tongue_vache, int *tail_vache, int time_tick_vache, int hour, int minite, int food)
{
    update();
    for (int i = 0; i < time_tick_vache; i++)
    {
        update();
        gotoxy(0, 0);
        printf("Time for now: %d:%d\tFood: %d\n", hour, minite, food);
        gotoxy(5, 0);
        time_t t;
        t = time(NULL);
        if (t % 2 == 0)
        {
            printf(" -");
            for (int i = 0; i ⩽ *length_vache; i++)
            {
                printf("-");
            }
            printf("\n");
            printf("< %s >\n", message_vache);
            printf(" -");
            for (int i = 0; i ⩽ *length_vache; i++)
            {
                printf("-");
            }
            printf("\n");
            printf("    \\   ^__^\n");
```

```
            printf("     \\  (%s)\\_______\n", eyes_vache);
            printf("        (__)\\       )\\/\");
            for (int i = 0; i < *tail_vache; i++)
            {
                printf("/\\");
            }
            printf("\n");
            printf("         %s |╞---w |\n", tongue_vache);
            printf("            ||     ||\n");
            printf("\n");
        }
        else
        {
            printf(" -");
            for (int i = 0; i ⩽ *length_vache; i++)
            {
                printf("-");
            }
            printf(" \n");
            printf("< %s >\n", message_vache);
            printf(" -");
            for (int i = 0; i ⩽ *length_vache; i++)
            {
                printf("-");
            }
            printf(" \n");
            printf("     \\   ^__^\n");
            printf("      \\  (%s)\\_______\n", "oo");
            printf("         (__)\\       )\\/\");
            for (int i = 0; i < *tail_vache; i++)
            {
                printf(" /\\");
            }
            printf(" \n");
            printf("          %s |╞---w |\n", "  ");
            printf("             ||     ||\n");
            printf(" \n");
        }
        // gotoxy(10, 0);
        sleep(1);
    }
}

/*
* @description: count the time
* @param {type}: int *hour, int *minite, int *food
* @return: arr[hour, minite]
*/
void time_count(int time_tick, int hour, int minite, int arr[])
{
    minite += 5;
    if (minite ⩾ 60)
```

```
    {
        hour += 1;
        minite -= 60;
    }
    if (hour ⩾ 24)
    {
        hour -= 24;
    }
    arr[0] = hour;
    arr[1] = minite;
}

/*
* @description: check if the enter number is a prime number
* @param {type}: int nombre
* @return: int
*/

int check_prime_number(int nombre)
{
    int i;
    for (i = 2; i < nombre; i++)
    {
        if (nombre % i == 0)
        {
            return 0;
        }
    }
    return 1;
}

/*
* @description: Main function
* @param {type}: int argc, char *argv[]
* @return: int
*/

int main(int argc, char *argv[])
{
    // Begin the game, define the variables
    int tail = 1;
    int life = 5;
    int food = 10;
    int time_tick = 5;
    char *eyes = "oo";
    char *tongue = "  ";
    int minite = 0;
    int hour = 0;

    // Define the opening message
    char message[MAX_LENGTH] = "Welcome to the COWSAY Game!";
    int length = strlen(message);
```

```
194     tongue = "U ";
195     affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour, minite,
    food);
196 
197     // Define the message of providing the name of the player
198     strcpy(message, "Please enter your name: ");
199     // *message = "Please enter your name: ";
200     length = strlen(message);
201     affiche_vache(&length, message, eyes, tongue, &tail, 2, hour, minite, food);
202 
203     // Define the name of the player and get the name from keyboard.
204     char name[20];
205     scanf("%s", name);
206 
207     // Welcome the player
208     strcpy(message, "Hello ");
209     strcat(message, name);
210     length = strlen(message);
211     affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour, minite,
    food);
212 
213     // Define the speed of the game, (4) for super fast (developer mode xd)
214     strcpy(message, "Please choose the speed of the game: (1) for slow, (2) for
    medium (default), (3) for fast");
215     length = strlen(message);
216     affiche_vache(&length, message, eyes, tongue, &tail, 1, hour, minite, food);
217 
218     // Define the speed of the game, and get the value from keyboard.
219     int game_speed;
220     scanf("%d", &game_speed);
221 
222     if (game_speed == 1)
223     {
224         strcpy(message, "You chose the slow speed, the game will be played in 10
    seconds");
225         length = strlen(message);
226         affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
227         time_tick = 8;
228     }
229     else if (game_speed == 2)
230     {
231         strcpy(message, "You chose the medium speed, the tivk will be 5 seconds");
232         length = strlen(message);
233         affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
234         time_tick = 5;
235     }
236     else if (game_speed == 3)
237     {
238         strcpy(message, "You chose the fast speed, the tick will be 3 seconds");
239         length = strlen(message);
```

```
240         affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
241         time_tick = 3;
242     }
243     else if (game_speed == 4)
244     {
245         strcpy(message, "You chose the super fast speed, the tick will be 1
    seconds");
246         length = strlen(message);
247         affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
248         time_tick = 1;
249     }
250     else
251     {
252         strcpy(message, "You chose the wrong thing, so the game will play in default
    speed, the tick will be 5 seconds");
253         length = strlen(message);
254         affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
255         time_tick = 5;
256     }
257 
258     int i = 0; // i is the number of the loop, not using later but just left there
    if needed
259 
260     while (life > 0 && life < 10)
261     {
262         // Count the time, with returning the value of the time in an array:
    arr[hour, minite]
263         int arr[2];
264         strcpy(message, "...");
265         length = strlen(message);
266         time_count(time_tick, hour, minite, arr);
267         hour = arr[0];
268         minite = arr[1];
269         tail = life;
270         affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
271 
272         i++;
273         update();
274 
275         // Check if minite is equal to 60, if yes its time to play a game/
276         if (minite == 0)
277         {
278             // Define the message of playing game.
279             strcpy(message, "It's time to think about something!");
280             length = strlen(message);
281             affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
282 
```

```
283             // Define a random number between 1 to 5, to chose which kind of math
problem we will use.
284             int random_number = rand() % (4) + 1;
285             srand(time(NULL)); // Initialize the random number generator.
286 
287             // If random is equal to 1 then we will use the prime problem.
288             if (random_number == 1)
289             {
290                 // Define the message of the prime problem, from 0 to 100.
291                 int nombre_premier = rand() % 100;
292                 // Initialize the message to be displayed.
293                 strcpy(message, "Is it a prime number? (1) for yes, (2) for no: ");
294                 // Make the prime number to be displayed.
295                 char str_nombre_premier[MAX_LENGTH] = "";
296                 sprintf(str_nombre_premier, "%d", nombre_premier);
297                 // Make message and prime number to be together.
298                 strcat(message, str_nombre_premier);
299 
300                 length = strlen(message);
301                 affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
hour, minite, food);
302 
303                 // Get the answer from keyboard.
304                 int answer = 0;
305                 scanf("%d", &answer);
306                 if (answer == 1)
307                 {
308                     if (check_prime_number(nombre_premier))
309                     {
310                         strcpy(message, "Yes, it is a prime number! You got it! Food
+ 2");
311                         length = strlen(message);
312                         eyes = "^^";
313                         tongue = "~~";
314                         affiche_vache(&length, message, eyes, tongue, &tail,
time_tick, hour, minite, food);
315                         food += 2;
316                     }
317                     else
318                     {
319                         strcpy(message, "No, it is not a prime number! You lost
it!");
320                         length = strlen(message);
321                         affiche_vache(&length, message, eyes, tongue, &tail,
time_tick, hour, minite, food);
322                     }
323                 }
324                 else
325                 {
326                     if (check_prime_number(nombre_premier))
327                     {
328                         strcpy(message, "Yes, it is a prime number! You lost it!");
```

```
                        length = strlen(message);
                        affiche_vache(&length, message, eyes, tongue, &tail,
time_tick, hour, minite, food);
                    }
                    else
                    {
                        strcpy(message, "No, it is not a prime number! You got it!
Food + 2");
                        length = strlen(message);
                        eyes = "^^";
                        tongue = "~~";
                        affiche_vache(&length, message, eyes, tongue, &tail,
time_tick, hour, minite, food);
                        food += 2;
                    }
                }
            }
            // If random is equal to 2 then we will use the addition problem.
            else if (random_number == 2)
            {
                // Define the 2 values of the addition problem, from 0 to 100.
                int nombre_1 = rand() % 100;
                int nombre_2 = rand() % 100;
                // Initialize the message to be displayed.
                strcpy(message, "What is the sum of ");
                // Make the two numbers to be displayed.
                char str_nombre_1[MAX_LENGTH] = "";
                char str_nombre_2[MAX_LENGTH] = "";
                sprintf(str_nombre_1, "%d", nombre_1);
                sprintf(str_nombre_2, "%d", nombre_2);
                strcat(message, str_nombre_1);
                strcat(message, " and ");
                strcat(message, str_nombre_2);
                strcat(message, ": ");

                length = strlen(message);
                affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
hour, minite, food);

                // Get the answer from keyboard.
                int answer = 0;
                scanf("%d", &answer);
                if (answer == nombre_1 + nombre_2)
                {
                    strcpy(message, "Yes, it is the sum of ");
                    strcat(message, str_nombre_1);
                    strcat(message, " and ");
                    strcat(message, str_nombre_2);
                    strcat(message, "! You got it! Food + 2");
                    length = strlen(message);
                    eyes = "^^";
                    tongue = "~~";
```

```
377                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
378                     food += 2;
379                 }
380                 else
381                 {
382                     strcpy(message, "No, it is not the sum of ");
383                     strcat(message, str_nombre_1);
384                     strcat(message, " and ");
385                     strcat(message, str_nombre_2);
386                     strcat(message, "! You lost it!");
387                     length = strlen(message);
388                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
389                 }
390             }
391             // multiplication
392             else if (random_number == 3)
393             {
394                 int nombre_1 = rand() % 100;
395                 int nombre_2 = rand() % 10;
396                 strcpy(message, "What is the product of ");
397                 char str_nombre_1[MAX_LENGTH] = "";
398                 char str_nombre_2[MAX_LENGTH] = "";
399                 sprintf(str_nombre_1, "%d", nombre_1);
400                 sprintf(str_nombre_2, "%d", nombre_2);
401                 strcat(message, str_nombre_1);
402                 strcat(message, " and ");
403                 strcat(message, str_nombre_2);
404                 strcat(message, ": ");
405 
406                 length = strlen(message);
407                 affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
408                 int answer = 0;
409                 scanf("%d", &answer);
410                 if (answer == nombre_1 * nombre_2)
411                 {
412                     strcpy(message, "Yes, it is the product of ");
413                     strcat(message, str_nombre_1);
414                     strcat(message, " and ");
415                     strcat(message, str_nombre_2);
416                     strcat(message, "! You got it! Food + 2");
417                     length = strlen(message);
418                     eyes = "^^";
419                     tongue = "~~";
420                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
421                     food += 2;
422                 }
423                 else
424                 {
```

```
425                     strcpy(message, "No, it is not the product of ");
426                     strcat(message, str_nombre_1);
427                     strcat(message, " and ");
428                     strcat(message, str_nombre_2);
429                     strcat(message, "! You lost it!");
430                     length = strlen(message);
431                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
432                 }
433             }
434             // difference
435             else if (random_number == 4)
436             {
437                 int nombre_1 = rand() % 100;
438                 int nombre_2 = rand() % 100;
439                 strcpy(message, "What is the difference of ");
440                 char str_nombre_1[MAX_LENGTH] = "";
441                 char str_nombre_2[MAX_LENGTH] = "";
442                 sprintf(str_nombre_1, "%d", nombre_1);
443                 sprintf(str_nombre_2, "%d", nombre_2);
444                 strcat(message, str_nombre_1);
445                 strcat(message, " and ");
446                 strcat(message, str_nombre_2);
447                 strcat(message, ": ");
448 
449                 length = strlen(message);
450                 affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
451                 int answer = 0;
452                 scanf("%d", &answer);
453                 if (answer == nombre_1 - nombre_2)
454                 {
455                     strcpy(message, "Yes, it is the difference of ");
456                     strcat(message, str_nombre_1);
457                     strcat(message, " and ");
458                     strcat(message, str_nombre_2);
459                     strcat(message, "! You got it! Food + 2");
460                     length = strlen(message);
461                     eyes = "^^";
462                     tongue = "~~";
463                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
464                     food += 2;
465                 }
466                 else
467                 {
468                     strcpy(message, "No, it is not the difference of ");
469                     strcat(message, str_nombre_1);
470                     strcat(message, " and ");
471                     strcat(message, str_nombre_2);
472                     strcat(message, "! You lost it!");
473                     length = strlen(message);
```

```
474                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
475                 }
476             }
477             // mod
478             else if (random_number == 5)
479             {
480                 int nombre_1 = rand() % 100;
481                 int nombre_2 = rand() % 10;
482                 strcpy(message, "What is the mod of ");
483                 char str_nombre_1[MAX_LENGTH] = "";
484                 char str_nombre_2[MAX_LENGTH] = "";
485                 sprintf(str_nombre_1, "%d", nombre_1);
486                 sprintf(str_nombre_2, "%d", nombre_2);
487                 strcat(message, str_nombre_1);
488                 strcat(message, " and ");
489                 strcat(message, str_nombre_2);
490                 strcat(message, ": ");
491 
492                 length = strlen(message);
493                 affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
494                 int answer = 0;
495                 scanf("%d", &answer);
496                 if (answer == nombre_1 % nombre_2)
497                 {
498                     strcpy(message, "Yes, it is the mod of ");
499                     strcat(message, str_nombre_1);
500                     strcat(message, " and ");
501                     strcat(message, str_nombre_2);
502                     strcat(message, "! You got it! Food + 2");
503                     length = strlen(message);
504                     eyes = "^^";
505                     tongue = "~~";
506                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
507                     food += 2;
508                 }
509                 else
510                 {
511                     strcpy(message, "No, it is not the mod of ");
512                     strcat(message, str_nombre_1);
513                     strcat(message, " and ");
514                     strcat(message, str_nombre_2);
515                     strcat(message, "! You lost it!");
516                     length = strlen(message);
517                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick,
    hour, minite, food);
518                 }
519             }
520         }
521 
```

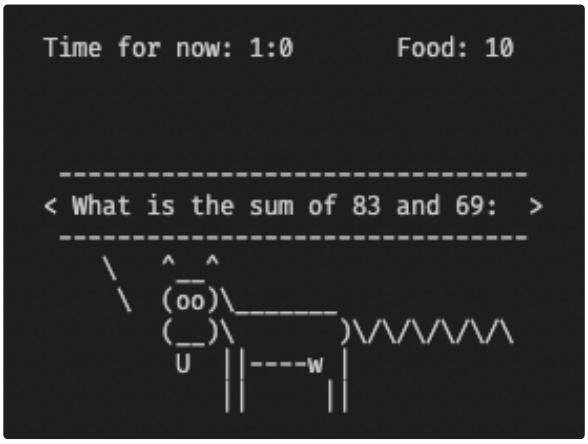
```
522         // If it's 6h, 12h, 18h, 24h then its time to eat, the player can choose if they want to feed the cow or not
523         if (hour % 6 == 0 && minite == 0)
524         {
525             strcpy(message, "It's time to eat! Do you want to eat? (1 for yes, food - 5, life + 2; 0 for no, life - random number)");
526             length = strlen(message);
527             affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour, minite, food);
528 
529             // Ask for the player if he wants to feed
530             int answer = 0;
531             scanf("%d", &answer);
532             if (answer == 1)
533             {
534                 if (food > 0)
535                 {
536                     food -= 5;
537                     strcpy(message, "You ate! Food - 5");
538                     length = strlen(message);
539                     life += 2;
540                     eyes = "^^";
541                     tongue = "~~";
542                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour, minite, food);
543                 }
544                 else
545                 {
546                     strcpy(message, "You don't have enough food!");
547                     length = strlen(message);
548                     int random_number = (rand() % (life - 2)) + 1;
549                     life -= random_number;
550                     eyes = "~~";
551                     tongue = "^";
552                     affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour, minite, food);
553                 }
554             }
555             else
556             {
557                 int random_number = (rand() % (life - 2)) + 1;
558                 life -= random_number;
559                 strcpy(message, "You didn't eat! Life - ");
560                 char str_random_number[MAX_LENGTH] = "";
561                 sprintf(str_random_number, "%d", random_number);
562                 strcat(message, str_random_number);
563                 length = strlen(message);
564                 eyes = "~~";
565                 tongue = "^";
566                 affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour, minite, food);
567             }
```

```
568         }
569         // random event
570         // Thunder: life - 2
571         if (hour + minite == rand() % 100)
572         {
573             strcpy(message, "It's a thunder! You lost life!");
574             length = strlen(message);
575             life -= 2;
576             eyes = "~~";
577             tongue = "^";
578             affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
579         }
580         // Hunger: food - 4
581         else if (hour + minite == rand() % 100)
582         {
583             strcpy(message, "It's a hunger! You lost food!");
584             length = strlen(message);
585             food -= 4;
586             eyes = "~~";
587             tongue = "^";
588             affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
589         }
590         // Fire: life - 2 && food - 2
591         else if (hour + minite == rand() % 100)
592         {
593             strcpy(message, "It's a fire! You lost life and food!");
594             length = strlen(message);
595             life -= 2;
596             food -= 2;
597             eyes = "~~";
598             tongue = "^";
599             affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
600         }
601         // Mercy: food + 5
602         else if (hour + minite == rand() % 100)
603         {
604             strcpy(message, "Mercy is coming! Food + 5");
605             length = strlen(message);
606             eyes = "^^";
607             tongue = "~~";
608             food += 5;
609             affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
    minite, food);
610         }
611     }
612     if (life <= 0)
613     {
614         strcpy(message, "You died because of hunger! Game over!");
615         length = strlen(message);
```

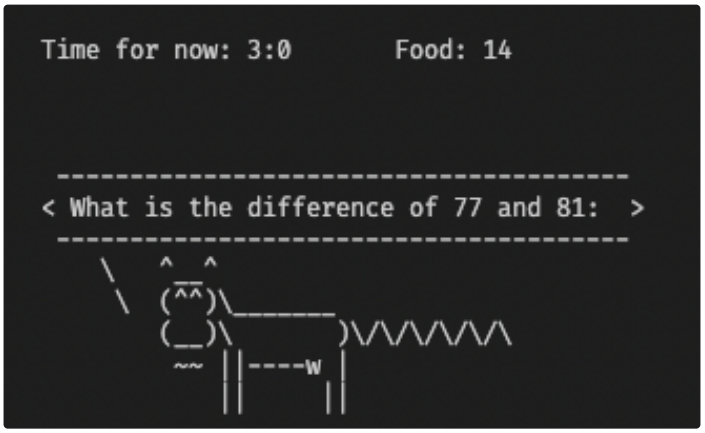
```
616          eyes = "xx";
617          tongue = "U ";
618          affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
     minite, food);
619      }
620      else if (life ⩾ 10)
621      {
622          strcpy(message, "You died because of trop plein! Game over!");
623          length = strlen(message);
624          eyes = "xx";
625          tongue = "U ";
626          affiche_vache(&length, message, eyes, tongue, &tail, time_tick, hour,
     minite, food);
627      }
628  }
```

Problème Mathématique

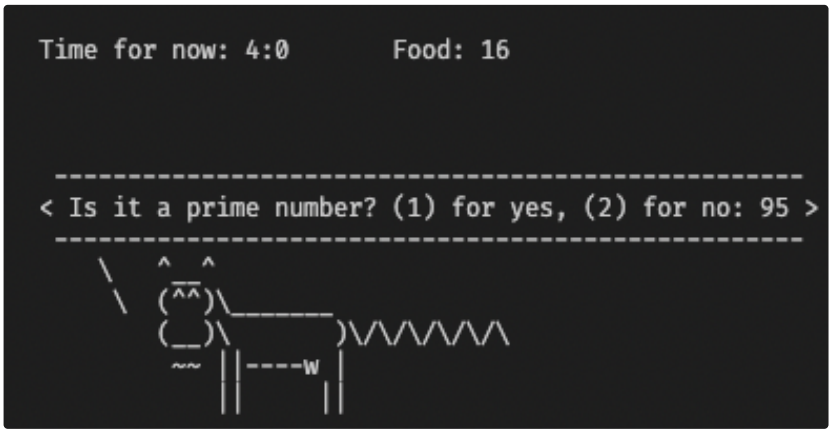
Case addition:



Case difference:

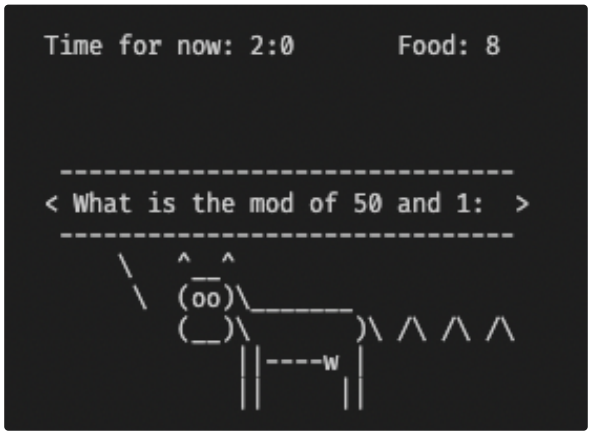


Case nombre prime:



Case multiplier:

Case mod:



Temp de manger:

```
Time for now: 6:0       Food: 22


 ------------------------------------------------------------------------------------------------
< It's time to eat! Do you want to eat? (1 for yes, food - 5, life + 2; 0 for no, life - random number) >
 ------------------------------------------------------------------------------------------------
    \   ^__^
     \  (oo)\_______
        (__)\       )\ /\ /\ /\ /\ /\
            ||----w |
            ||     ||
```

Mort de faim

```
Time for now: 6:35      Food: 27


 ---------------------------------------
< You died because of hunger! Game over! >
 ---------------------------------------
    \   ^__^
     \  (oo)\_______
        (__)\       )\ /\ /\
            ||----w |
            ||     ||
```