

# PROJET\_INF203

Groupe: INF2

Prénom et Nom: Shaghayegh HAJMOHAMMADKASHI

## 2 Préliminaires:

Cowsay/Cowthink [-bdgpstwy] [-h] [-e eyes] [-f cowfile] [-l] [-n] [-T tongue] [-W wrapcolumn] [message]

Option	Affichage
-b (borg)	<pre>&lt; Hello World! &gt; -----       \   ^__^        (oo)\_____           (__)\       )\/\                 ----w                          </pre>
-d (dead)	<pre>&lt; I'm dead &gt; -----       \   ^__^        (xx)\_____           (__)\       )\/\               u      ----w                          </pre>
-t(tired)	<pre>&lt; I'm tired. &gt; -----       \   ^__^        (--)\_____           (__)\       )\/\                 ----w                          </pre>
-g(greedy)	<pre>&lt; Mmm, grass &gt; -----       \   ^__^        (\$\$)\_____           (__)\       )\/\                 ----w                          </pre>
-p(paranoia)	<pre>&lt; Oh no! &gt; -----       \   ^__^        (@@)\_____           (__)\       )\/\                 ----w                          </pre>

-s(stoned)	<pre> ----- &lt; Ooops! &gt; -----       \   ^__^        \  (oo)\_______             U        ----w                         </pre>	
-w(surprised)	<pre> ----- &lt; Whaaat?! &gt; -----       \   ^__^        \  (oo)\_______             U        ----w                         </pre>	
-y(youthful)	<pre> ----- &lt; Hi! &gt; -----       \   ^__^        \  (..)\_______             U        ----w                         </pre>	
-e(eyes) cowsay -e "^^" "Cute!"	<pre> ----- &lt; Cute! &gt; -----       \   ^__^        \  (^^)\_______             U        ----w                         </pre>	
-T(tongue) cowsay -g -T "v" "I'm rich!"	<pre> ----- &lt; I'm rich! &gt; -----       \   ^__^        \  (\$\$)\_______             U        ----w               v         </pre>	
cowthink -e "oO" "What was that?!"	<pre> ( What was that?! ) -----     o   ^__^     o   (oO)\_______             U        ----w                         </pre>	

# Bash

## Cow\_kindergarten

```
#!/bin/bash

for i in {1..9}; do
    cowsay $i
    sleep 1
done

cowsay -T "U" "10"
```

## Cow\_primaryschool

```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 n"
    echo "  n : nombre jusqu'auquel la vache doit compter"
    exit 1
fi

n=$1

if ! [[ "$n" =~ ^[0-9]+$ ]]; then
    echo "Erreur: '$n' n'est pas un nombre valide"
    exit 1
fi

for i in $(seq 1 $n); do
    cowsay $i
    sleep 1
done

cowsay -T "U" "J'ai fini!"
```

## Cow\_highschool

```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 n"
```

```

    echo "  n : nombre jusqu'auquel la vache doit compter"
    exit 1
fi

n=$1

if ! [[ "$n" =~ ^[0-9]+$ ]]; then
    echo "Erreur: '$n' n'est pas un nombre valide"
    exit 1
fi

for i in $(seq 1 $n); do
    cowsay $((i**2))
    sleep 1
done

cowsay -T "U" "J'ai fini!"

```

## Cow\_college

```

#!/bin/bash
if [[ $# -ne 1 ]]; then
    echo "Usage: $0 n"
    exit 1
fi

n=$1
a=0
b=1

while [[ $a -le $n ]]; do
    echo $a | cowsay
    sleep 1
    c=$((a + b))
    a=$b
    b=$c
done

cowsay -T "U" "J'ai fini!"

```

## Cow\_university.sh

```
#!/bin/bash

n=$1
primes=(2)
for ((i=3; i<=n; i++)); do
    is_prime=true
    for p in "${primes[@]}"; do
        if (( i % p == 0 )); then
            is_prime=false
            break
        fi
    done
    if $is_prime; then
        primes+=("$i")
        echo "$i" | cowsay
        sleep 1
    fi
done
cowsay -T "U" "Je suis une vache intelligente qui connaît les nombres premiers jusqu'à $n!"
```

## Smart\_cow

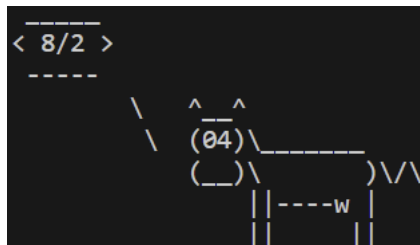
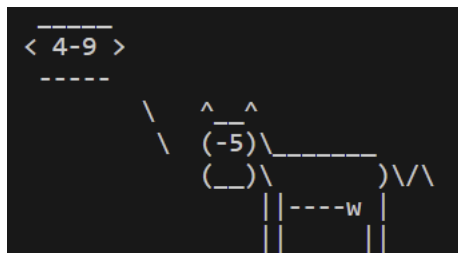
```
#!/bin/bash

# Récupérer l'expression mathématique en argument
expression="$1"

# Résoudre l'expression mathématique
result=$(echo "$expression" | bc)

# Créer la vache avec les yeux affichant le résultat
if [ "$result" -lt 10 ] && [ "$result" -ge 0 ]; then
    cowsay -e "0$result" "$expression"
else
    cowsay -e "$result" "$expression"
fi
```

« bc » est un programme qui permet d'effectuer des calculs mathématiques en ligne de commande dans un terminal Unix ou Linux.



Crazy\_cow

```
#!/bin/bash

echo "Bonjour, comment t'appelles-tu ?"
read prenom

echo "Bonjour $prenom, quel âge as-tu ?"
read age

echo "Je vais te poser quelques questions pour déterminer ton espérance de
vie."
echo "Réponds par oui ou non."

reduction=100
echo "Est-ce que tu fumes ?"|cowsay
read fume
if [ "$fume" == "oui" ]; then
    reduction=$((reduction - 10))
fi

echo "Est-ce que tu bois de l'alcool ?"|cowsay
read alcool
if [ "$alcool" == "oui" ]; then
    reduction=$((reduction - 5))
fi

echo "Est-ce que tu manges souvent de la junk food ?"|cowsay
read junk_food
if [ "$junk_food" == "oui" ]; then
    reduction=$((reduction - 6))
```

```

fi

echo "Est-ce que tu fais régulièrement du sport ?" | cowsay
read workout
if [ "$workout" == "non" ]; then
    reduction=$((reduction - 7))
fi

esp_vie=$((reduction - age))

echo "D'après mes calculs, tu as une espérance de vie de $esp_vie ans." |
cowsay
if [ $esp_vie -lt 0 ]; then
    echo "Désolé $prenom, mais tu ne vas probablement pas survivre jusqu'à ta
retraite !" | cowsay -e "--"
else
    echo "Tu as encore pas mal de temps devant toi, profite-en bien $prenom !" |
cowthink -e "^^"
fi

```

```

Bonjour, comment t'appelles-tu ?
Shaghayegh
Bonjour Shaghayegh, quel âge as-tu ?
22
Je vais te poser quelques questions pour déterminer ton espérance de vie.
Réponds par oui ou non.

```

```

-----
< Est-ce que tu fumes ? >
-----

```

```

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

```

```
non
```

```

-----
< Est-ce que tu bois de l'alcool ? >
-----

```

```

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

```

```
non
```

```

-----
/ Est-ce que tu manges souvent de la junk \
\ food ?                                     /
-----

      ^  ^
      (oo)\_____
      (__) \       )\/\
           ||----w |
           ||       ||

oui

-----
/ Est-ce que tu fais régulièrement du \
\ sport ?                               /
-----

      ^  ^
      (oo)\_____
      (__) \       )\/\
           ||----w |
           ||       ||

non

```

```

-----
/ D'après mes calculs, tu as une \
\ espérance de vie de 65 ans.      /
-----

      ^  ^
      (oo)\_____
      (__) \       )\/\
           ||----w |
           ||       ||

( Tu as encore pas mal de temps devant )
( toi, profite-en bien Shaghayegh !   )
-----

o  ^  ^
o  (^^)\_____
   (__) \       )\/\
        ||----w |
        ||       ||

```

C

Newcow

```

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

```



```

// exo1
// void affiche_vache() {
//     printf("      \\\  ^__^\n");
//     printf("      \\\  (oo)\\\_____\n");
//     printf("      (____)\\\      )\\/\n");
//     printf("      ||----w |\n");
//     printf("      ||      ||\n");
// }

// int main() {
//     affiche_vache();
//     return 0;
// }

// exo 2 et 3
void affiche_vache(char* eyes, char* tongue, int tail) {

    printf("      \\\  ^__^\n");

    if (eyes != NULL) {
        printf("      \\\  (%s)\\\_____\n", eyes);
    } else {
        printf("      \\\  (oo)\\\_____\n");
    }

    if (tail != 0) {
        printf("      (____)\\\      )");
        int i;
        for (i = 0; i < tail; i++) {
            if (i % 2 == 0) {
                printf("\\");
            } else {
                printf("/");
            }
        }
        printf("\n");
    } else {
        printf("      (____)\\\      )\\/\n");
    }

    if (tongue != NULL) {
        printf("      %s  ||----w |\n", tongue);
        printf("      ||      ||\n");
    } else {
        printf("      ||----w |\n");
        printf("      ||      ||\n");
    }

}

int main(int argc, char** argv) {

```

```

// Définition des arguments par défaut
char* eyes = "oo";
char* tongue = NULL;
int tail = 0;
int num_cows = 1; // nombre de vaches à afficher

// Boucle pour traiter les arguments de ligne de commande
int i;
for (i = 1; i < argc; i++) {
    // "-e" suivi d'une chaîne de caractères pour les yeux de la vache
    if (strcmp(argv[i], "-e") == 0) {
        i++;
        eyes = argv[i];
    } // "-T" suivi d'une chaîne de caractères pour la langue de la vache
    else if (strcmp(argv[i], "-T") == 0) {
        i++;
        tongue = argv[i];
    } // "--tail" suivi d'un entier pour la longueur de la queue de la vache
    else if (strcmp(argv[i], "--tail") == 0) {
        i++;
        tail = atoi(argv[i]);
    } // "-b" pour afficher une vache avec des yeux égaux ("==")
    else if (strcmp(argv[i], "-b") == 0) {
        eyes = "==";
    } // "-d" pour afficher une vache avec des yeux en forme de "xx" et une
    // langue "U"
    else if (strcmp(argv[i], "-d") == 0) {
        eyes = "xx";
        tongue = "U";
    } // "-n" suivi d'un entier pour le nombre de vaches à afficher
    else if (strcmp(argv[i], "-n") == 0) {
        i++;
        num_cows = atoi(argv[i]);
    }
}

// Appel de la fonction affiche_vache plusieurs fois avec des arguments
différents
int j;
for (j = 0; j < num_cows; j++) {
    affiche_vache(eyes, tongue, tail);
}

return 0;
}

```

```
hajmohas@im2ag-turing-01: [~/INF203/Projet/C]: ./newcow
```

```
  \      ^__^
   \      (oo)\_____
      (____)\       )\/\
              ||----w |
              ||     ||
```

## wildcow

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

// void affiche_vache_normale() {
//     printf("      \\\  ^__^\n");
//     printf("      \\\  (oo)\\\_____\\n");
//     printf("      (____)\\       )\\/\n");
//     printf("              ||----w |\\n");
//     printf("              ||     ||\\n");
// }

// void affiche_vache_avec_cou_en_bas() {
//     printf("      \\\      \\n");
//     printf("      \\\      _____\\n");
//     printf("      ^__^/      )\\/\n");
//     printf("      (--)//||----w |\\n");
//     printf("      (____) ||     ||\\n");
//     printf("                        \\n");
// }

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

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

void affiche_vache_normale(char* eyes, char* tongue, int tail) {
    printf("      \\\  ^__^\n");

    if (eyes != NULL) {
        printf("      \\\  (%s)\\_____\\n", eyes);
    } else {
        printf("      \\\  (oo)\\_____\\n");
    }

    if (tail != 0) {
        printf("      (____)\\       )");
    }
}
```

```

        int i;
        for (i = 0; i < tail; i++) {
            if (i % 2 == 0) {
                printf("\\");
            } else {
                printf("/");
            }
        }
        printf("\n");
    } else {
        printf("          (__)\\          )\\/\n");
    }
    if (tongue != NULL) {
        printf("          %s  ||----w |\n", tongue);
        printf("          ||      ||\n");
    } else {
        printf("          ||----w |\n");
        printf("          ||      ||\n");
    }
}

void affiche_vache_avec_cou_en_bas(char* eyes, char* tongue, int tail) {
    printf("          \\          \n");
    printf("          \\          _____\n");

    if (tail != 0) {
        printf("          ^__^/\n");
        int i;
        for (i = 0; i < tail; i++) {
            if (i % 2 == 0) {
                printf("\\");
            } else {
                printf("/");
            }
        }
        printf("\n");
    } else {
        printf("          ^__^/\n");
    }
    if (eyes != NULL) {
        printf("          (%s)/||----w |\n", eyes);
    } else {
        printf("          (--)||----w |\n");
    }
    if (tongue != NULL) {
        printf("          (__) ||      ||\n");
        printf("          %s          \n", tongue);
    } else {

```

```

        printf("          (__) ||      ||\n");
        printf("          \n");
    }
}

int main() {
    char* eyes_open = "oo";
    char* eyes_closed = "--";
    int blink_interval = 2; // intervalle en secondes entre chaque clignement
    int tail_length = 0;
    char* tongue = NULL;
    char *tongue_out = "U";

    int x = 0; // position x de la vache
    int y = 0; // position y de la vache
    int dir = 1; // direction de la marche, 1 = droite, -1 = gauche

    while (1) {
        update();

        // affiche la vache à la position courante
        gotoxy(x, y);
        if (dir == 1) {
            affiche_vache_normale(eyes_open, tongue , tail_length);
        } else {
            affiche_vache_avec_cou_en_bas(eyes_open, tongue , tail_length);
        }

        fflush(stdout); // vide le buffer de sortie pour que l'affichage soit
        immédiat
        sleep(blink_interval);

        // efface la vache à la position courante
        gotoxy(x, y);
        printf("          ");

        // met à jour la position de la vache
        x += dir;
        if (x == 80 || x == 0) {
            dir = -dir;
        }

        // change la direction du regard de la vache
        if (x % 10 == 0) {
            eyes_open = NULL;
        } else {
            eyes_open = "oo";
        }
    }
}

```

```

        update();

        // affiche la vache à la nouvelle position avec le cou baissé et les
yeux fermés
        gotoxy(x, y);
        affiche_vache_avec_cou_en_bas(eyes_closed, tongue_out , tail_length);

        fflush(stdout);
        sleep(blink_interval);

        // efface la vache à la position courante
        gotoxy(x, y);
        printf("                ");

        // remet les yeux et le cou à leur position normale
        eyes_open = "oo";
    }

    return 0;
}

```

```

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

```

```

      \  ^__^
      \  (--) /_______
          (__) ||      )\/\
              ||----w |
              ||     ||
      u

```

## reading\_cow

```

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

void affiche_vache(char* eyes, char* tongue, int tail, char* text) {
    int text_len = strlen(text);

```

```

// Affichage de la ligne horizontale supérieure
printf(" ");
for (int i = 0; i < text_len + 5; i++) {
    printf("-");
}
printf("\n");
printf(" <%s>\n", text);
// Affichage de la ligne horizontale inférieure
printf(" ");
for (int i = 0; i < text_len + 5; i++) {
    printf("-");
}
printf("\n");
printf("      \\   ^__^\\n");

if (eyes != NULL) {
    printf("      \\   (%s)\\_____\\n", eyes);
} else {
    printf("      \\   (oo)\\_____\\n");
}

if (tail != 0) {
    printf("      (__)\\          )");
    int i;
    for (i = 0; i < tail; i++) {
        if (i % 2 == 0) {
            printf("\\");
        } else {
            printf("/");
        }
    }
    printf("\\n");
} else {
    printf("      (__)\\          )\\ /\\n");
}

if (tongue != NULL) {
    printf("      %c   ||----w   |\\n", *tongue);
    printf("      ||       ||\\n");
} else {
    printf("      ||----w   |\\n");
    printf("      ||       ||\\n");
}
}

int main(int argc, char** argv) {
    // Définition des arguments par défaut
    char* eyes = "oo";
    char* tongue = NULL;
    int tail = 0;

```

```

int num_cows = 1; // nombre de vaches à afficher

FILE* fp;

if (argc < 2) {
    fp = stdin; // utilise l'entrée standard si aucun fichier n'est
spécifié en ligne de commande
} else {
    fp = fopen(argv[1], "r");
    if (fp == NULL) {
        printf("Erreur lors de l'ouverture du fichier.\n");
        return 1;
    }
}

char c;
char text[100] = "";
int i = 0;
while ((c = fgetc(fp)) != EOF) {
    text[i++] = c;
    text[i] = '\0'; // ajoute le caractère dans la chaîne de texte et
ajoute le caractère de fin de chaîne
    system("clear"); // Efface l'affichage précédent avant d'afficher la
nouvelle vache
    affiche_vache(eyes, &text[i-1], tail, text);
    sleep(1); // Attend 1 seconde avant de lire le prochain caractère
}

tongue = " "; // Met la langue à une valeur fixe après la fin de la
boucle while
if (i > 0 && text[i-1] == '\n') {
    i--;
    text[i] = '\0'; // Supprime le caractère de saut de ligne de la chaîne
de texte
}

system("clear"); // Efface l'affichage précédent avant d'afficher la
dernière vache
affiche_vache(eyes, tongue, tail, text);

if (fp != stdin) {
    fclose(fp); // ferme le fichier si un fichier a été spécifié en ligne
de commande
}

return 0;

```

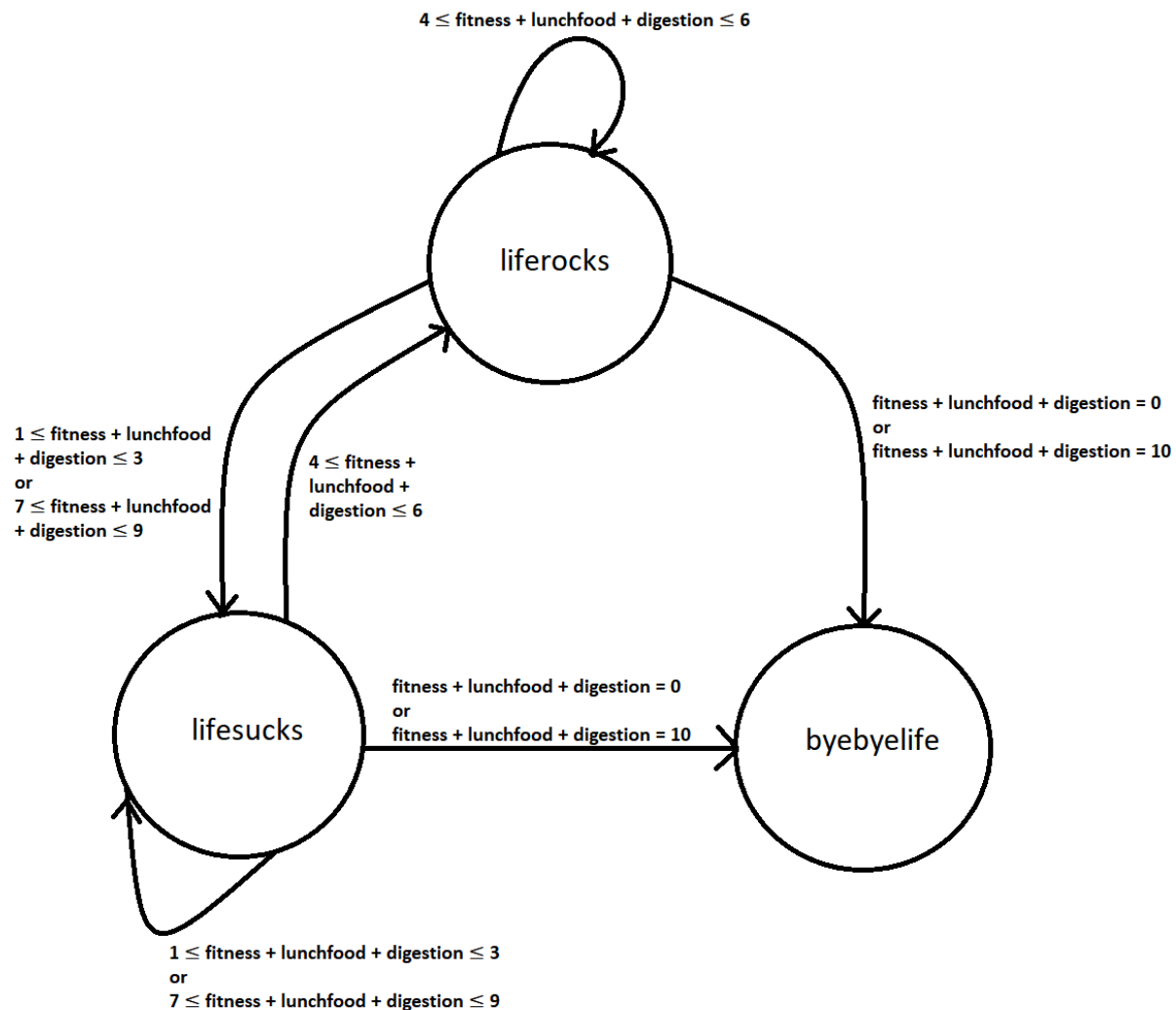


```
}
```



# Automates

```
# Initialisation des variables
life = 5
fitness = 5
stock = 5
digestion = random(-3, 0)
crop= random(-3, 3)
lunchfood≤stock
stock=(stock-lunchfood)+crop
fitness=(fitness+lunchfood)+digestion
```



```

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

// Constantes pour les états de santé de la vache
#define BYEBYLIFE 0
#define LIFESUCKS 1
#define LIFEROCKS 2

// Constantes pour les valeurs minimales et maximales des variables stock et fitness
#define STOCK_MIN 0
#define STOCK_MAX 10
#define FITNESS_MIN 0
#define FITNESS_MAX 10

// Variables globales
int stock = 5;
int fitness = 5;
int duree_de_vie = 0;

// Fonctions pour mettre à jour les variables stock et fitness
int stock_update(int lunchfood) {
    // Met à jour la variable stock en fonction du niveau de forme de la vache
    et de la quantité de nourriture
    int crop = rand() % 3;
    int digestion = rand() % 3;
    int stock_update = lunchfood - crop - digestion;
    if (stock + stock_update < STOCK_MIN) {
        stock = STOCK_MIN;
    } else if (stock + stock_update > STOCK_MAX) {
        stock = STOCK_MAX;
    } else {
        stock += stock_update;
    }
    return stock_update;
}

int fitness_update(int lunchfood) {
    // Met à jour la variable fitness en fonction du niveau de forme de la
    vache et de la quantité de nourriture
    int mood = rand() % 3;
    int fitness_update = lunchfood - mood;
    if (fitness + fitness_update < FITNESS_MIN) {
        fitness = FITNESS_MIN;
    } else if (fitness + fitness_update > FITNESS_MAX) {
        fitness = FITNESS_MAX;
    } else {

```

```

        fitness += fitness_update;
    }
    return fitness_update;
}

int main() {
    // Initialisation du générateur de nombres aléatoires
    time_t t;
    srand((unsigned) time(&t));

    // Boucle principale du jeu
    int etat_sante = LIFEROCKS;
    while (etat_sante != BYEBYLIFE) {
        // Affichage de la vache dans son état actuel
        switch (etat_sante) {
            case BYEBYLIFE:
                printf("La vache est morte. RIP.\n");
                break;
            case LIFESUCKS:
                printf("La vache ne se sent pas bien.\n");
                break;
            case LIFEROCKS:
                printf("La vache est en pleine forme !\n");
                break;
        }

        // Affichage de l'état de la variable stock
        printf("Etat de la reserve de nourriture : %d\n", stock);

        // Demande de la quantité de nourriture allouée à la vache
        int lunchfood;
        printf("Entrez la quantite de nourriture allouee a la vache : ");
        scanf("%d", &lunchfood);
        while (lunchfood < 0 || lunchfood > stock) {
            printf("Erreur : quantite invalide. Entrez une valeur entre 0 et %d : ", stock);
            scanf("%d", &lunchfood);
        }

        // Mise à jour des variables stock et fitness ainsi que des états de la réserve et de la vache

        int stock_update_value = stock_update(lunchfood);
        int fitness_update_value = fitness_update(lunchfood);
        // Affichage des mises à jour des variables stock et fitness
        printf("Mise a jour de la reserve de nourriture : %d\n", stock_update_value);
    }
}

```

```

        printf("Mise a jour de la forme de la vache : %d\n",
fitness_update_value);

        // Mise à jour de l'état de santé de la vache en fonction de ses
variables stock et fitness
        if (stock <= 2 || fitness <= 2) {
            etat_sante = BYEBYLIFE;
        } else if (stock <= 5 || fitness <= 5) {
            etat_sante = LIFESUCKS;
        } else {
            etat_sante = LIFEROCKS;
        }

        // Incrémentation de la durée de vie de la vache
duree_de_vie++;

        // Affichage de la durée de vie de la vache
printf("Duree de vie de la vache : %d\n", duree_de_vie);
    }

    // Affichage de la durée de vie de la vache à la fin du jeu
printf("La vache a vecu %d jours.\n", duree_de_vie);

    return 0;
}

// FIN DU PROGRAMME

```

```
La vache est en pleine forme !
Etat de la reserve de nourriture : 5
Entrez la quantite de nourriture allouee a la vache : 3
Mise a jour de la reserve de nourriture : -1
Mise a jour de la forme de la vache : 2
Duree de vie de la vache : 1
La vache ne se sent pas bien.
Etat de la reserve de nourriture : 4
Entrez la quantite de nourriture allouee a la vache : 2
Mise a jour de la reserve de nourriture : 0
Mise a jour de la forme de la vache : 2
Duree de vie de la vache : 2
La vache ne se sent pas bien.
Etat de la reserve de nourriture : 4
Entrez la quantite de nourriture allouee a la vache : 1
Mise a jour de la reserve de nourriture : 0
Mise a jour de la forme de la vache : -1
Duree de vie de la vache : 3
La vache ne se sent pas bien.
Etat de la reserve de nourriture : 4
Entrez la quantite de nourriture allouee a la vache : 1
Mise a jour de la reserve de nourriture : 0
Mise a jour de la forme de la vache : -1
Duree de vie de la vache : 4
La vache ne se sent pas bien.
Etat de la reserve de nourriture : 4
Entrez la quantite de nourriture allouee a la vache : 1
Mise a jour de la reserve de nourriture : 0
Mise a jour de la forme de la vache : 0
Duree de vie de la vache : 5
La vache ne se sent pas bien.
Etat de la reserve de nourriture : 4
Entrez la quantite de nourriture allouee a la vache : 1
Mise a jour de la reserve de nourriture : -3
Mise a jour de la forme de la vache : 1
Duree de vie de la vache : 6
```