

UNIPRO (Univers Professionnel)

Département : Génie Informatique

Classe : L1GI A (Licence 1)

Module : Langage C && Algo 2



RAPPORT DU PROJET EN LANGUAGE C

Période du projet : Du 22/05 AU 02/06/2023

Année Académique 2023/2024

Présenté et soutenu par le Groupe1 :

1. Jophret Danicha BAKANA
2. Awa SALL
3. Françoise BASSE
4. Fils Kris Nephtali MEMIAGHE
5. Hawa DEMBA KEITA
6. ONDI APILI Ronh Geoffrey

Encadré par :

Mr Youssou NDIONE

SOMMAIRE

Introduction

I-Tâche 1

Création de six programmes.

II-Tâche 2

Regroupement des différents programmes à l'aide l'étiquetage (**goto**) et des **switch case**.

II-Tâche 3

Rassemblement des différents programmes à l'aide des boucles **for**, **while** et **Do-while**.

Conclusion

INTRODUCTION

Dans le cadre de notre apprentissage du langage de programmation C, nous avons été chargés de développer une série de programmes visant à maîtriser les bases du C. Ce projet nous a permis de mettre en pratique des concepts essentiels tels que la manipulation des entrées et des sorties, les structures de contrôle, les calculs arithmétiques et la gestion des conditions. La première partie du projet consistait à développer ces programmes individuellement, tandis que la seconde partie consistait à les regrouper en un programme unique sans utiliser de boucles. La troisième partie nous demandait de regrouper les programmes en utilisant des boucles. Ce projet a été conçu pour renforcer notre compréhension des concepts de base du C et développer nos compétences en résolution de problèmes.

I-TACHE 1

DESCRIPTION DE LA TACHE

On nous demande de créer 6 les programmes. Dès lors les codes sources de ces programmes sont disponibles dans le dossier compressé ci-joint. L'explicitation de ces programmes est disponible ci-dessous :

PROGRAMME 1 :

Ce programme en C permet de saisir, valider et afficher le prénom, le nom et l'âge de l'utilisateur basé sur sa date de naissance. Il commence par demander le prénom et le nom, en vérifiant que chaque entrée contient uniquement des lettres grâce à **isalpha** (qui est une fonction de la bibliothèque **ctype.h** permettant de vérifier si le premier caractère est une lettre de l'alphabet). Si une entrée est invalide, une erreur est affichée et le programme se termine. Au cas contraire, l'utilisateur est invité à saisir sa date de naissance au format JJ/MM/AAAA. Cette date est

soumise à une condition de validité si le jour est bien compris entre 1 et 31 ou le mois est bien compris entre 1 et 12 ou l'année est bien comprise entre 1900 et 2025. Si la date est validée alors l'âge est calculé en soustrayant l'année de naissance de l'année actuelle (2024) et ajusté pour le mois que ce dernier a saisi. Enfin, le programme affiche un message de salutation incluant le prénom, le nom et l'âge de l'utilisateur.

Illustration :

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

int main()
{
    char prenom[15], nom[15];
    char car='/';
    int D = 2024, jour, mois, annee;
    float age;
    printf("Veuillez saisir votre prenom(Nous considerons qu'un seul prenom) :\n");
    scanf("%s", &prenom);
    if (!isalpha(prenom[0]))
    {
        printf("Erreur: le prenom doit contenir que des lettres\n\n");
        return 0;
    }
    printf("Veuillez saisir votre nom(Nous considerons qu'un seul nom) :\n");
    scanf("%s", &nom);
    if (!isalpha(nom[0]))
    {
        printf("Erreur: le nom doit contenir que des lettres\n\n");
        return 0;
    }

    printf("Veuillez saisir votre date de naissance (JJ/MM/AAAA):\n");
    scanf("%d%c%d%c%d", &jour, &car, &mois, &car, &annee);
    if (jour < 1 || jour > 31 || mois < 1 || mois > 12 || car != '/' || annee < 1900 || annee > 2025 )
    {
        printf("La Date est invalide\n\n");
    }
    age = D - annee;
    if (mois > 2 && mois <= 12)
    {
        age += (mois - 1) * 0.1;
    }
    printf("Salut %s %s, vous etes age de %.1f ans.\n\n", prenom, nom, age);
    return 0;
}
```

PROGRAMME 2 :

Ce programme permet à l'utilisateur de saisir trois nombres réels et calcule ensuite le maximum et le minimum de ces nombres. De ce fait, Le programme invite l'utilisateur à saisir trois nombres réels, un à la fois, pour chaque saisie, il vérifie la validité de l'entrée, c'est-à-dire que l'utilisateur a bien entré un nombre réel et qu'aucun caractère supplémentaire n'est présent après le nombre. Le maximum est calculé en utilisant des opérateurs ternaires pour comparer les trois nombres. Le programme Compare N1 et N2 : si N1 est supérieur à N2, il est ensuite comparé à

N3 au cas contraire il compare N2 et N3 : si N2 est supérieur à N3, il est retenu comme maximum. De la même manière que pour le maximum, le minimum est déterminé par des opérateurs ternaires mais ici le résultat sera retenu comme minimum. Enfin, Les résultats des calculs (le maximum et le minimum) sont affichés avec une précision de trois décimales pour une meilleure clarté.

Illustration :

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

int main()
{
    float N1, N2, N3, Max, Min;
    char extra;
    printf("Veuillez saisir le premier reel\n");
    if (scanf("%f%c", &N1, &extra) != 2 || extra != '\n'){
        printf("Erreur : Veuillez saisir seulement des nombres la prochaine fois. Au revoir \n\n");
        return 0;
    }
    printf("Veuillez saisir le deuxieme reel\n");
    if (scanf("%f%c", &N2, &extra) != 2 || extra != '\n'){
        printf("Erreur : Veuillez saisir seulement des nombres la prochaine fois. Au revoir \n\n");
        return 0;
    }
    printf("Veuillez saisir le troisieme reel\n");
    if (scanf("%f%c", &N3, &extra) != 2 || extra != '\n'){
        printf("Erreur : Veuillez saisir seulement des nombres la prochaine fois. Au revoir \n\n");
        return 0;
    }

    Max = (N1 > N2) ? (N1 > N3 ? N1 : N3) : (N2 > N3 ? N2 : N3);
    Min = (N1 < N2) ? (N1 < N3 ? N1 : N3) : (N2 < N3 ? N2 : N3);

    printf("Le maximum est: %.3f\n", Max);
    printf("Le minimum est: %.3f\n", Min);
    return 0;
}
```

PROGRAMME 3 :

Il s'agit d'écrire un programme qui demande à l'utilisateur de saisir un nombre strictement positif et qui affiche si ce dernier est premier ou pas. Pour rappel, un nombre premier est un nombre qui n'a que deux diviseurs **zéro** et **lui-même**, pour se faire nous avons fait appel aux structures conditionnelles. Premièrement, le programme demande à l'utilisateur de saisir un nombre strictement positif, si et seulement si l'utilisateur tape une lettre, on lui envoie automatiquement un message d'erreur car ici le **scanf** est soumis à une condition à l'aide de la variable **extra** (de type char) dans la mesure où si l'utilisateur saisi des valeurs différentes d'un

entier et d'un saut de page(\n) alors il revoie un message d'erreur et le programme se termine ou si l'utilisateur tape un nombre qui n'est pas strictement positif, il reçois aussi un message d'erreur et se termine. Dans le cas contraire, le programme vérifie si le nombre saisi est inférieur à 2 si c'est le cas, il n'est pas premier, car il s'agit de 1. Ensuite si le nombre est égal à 2,3,5 ou 7, il est premier. Pour continuer, s'il est divisible par 2,3,5, ou 7 et que le reste de la division est égale à zéro, ce dernier n'est pas premier. Enfin, s'il est divisible par 2 ou 3 et que le résultat est différent de zéro ce nombre est premier dans le cas contraire il ne l'ait pas Et le programme lui montre le résultat.

Illustration :

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

int main()
{
    int nbr;
    char extra;
    printf("Veuillez saisir un entier strictement positif!\n");
    if (scanf("%d%c", &nbr, &extra) != 2 || extra != '\n') {
        printf("Erreur : Veuillez saisir seulement des entiers strictement positifs la prochaine fois. Au revoir \n\n");
        return 0;
    }
    if(nbr>0)
    {
        if (nbr<2)
        {
            printf("L'entier %d n'est pas premier \a\n\n",nbr);
        }

        else if (nbr==2 || nbr==3 || nbr==5 || nbr==7)
        {
            printf("L'entier %d est premier \a\n",nbr);
        }

        else if(nbr%2==0 || nbr%3==0 || nbr%5==0 || nbr%7==0)
        {
            printf("L'entier %d n'est pas premier \a\n\n",nbr);
        }
        else if (nbr%2!=0 || nbr%3!=0)
        {
            printf("L'entier %d est premier \a\n\n",nbr);
        }
        else
        {
            printf("L'entier %d n'est premier \a\n\n",nbr);
        }
    }
    else
    {
        printf("cet entier n'est pas strictement positif \a\n\n");
    }
    return 0;
}
```

PROGRAMME 4 :

Le programme consiste à calculer la somme des entiers pairs entre 0 et 100 sauf les nombres divisibles par 10. En utilisant des structures, des étiquettes, le calcul s'interrompt dès que la somme dépasse 500. Pour ce faire nous avons déclarés et initialisés **somme** et **i** (Variables de

type entier **Int**) à 0. Nous avons créé deux étiquettes nommées **test** et **affiche**, **test** qui marque le début de la boucle et **affiche** qui met fin au calcul. La condition **if i>100** vérifie si la variable **i** dépasse 100. L'autre condition vérifie si la variable **i** est pair et non divisible par 10. Si la condition est vraie **i** est ajoutée à **somme** ensuite vérifie si ce dernier est supérieur à 500. Si c'est le cas le programme passe à l'étiquette **affiche** ainsi la somme des entiers pairs entre 0 et 100 est affichée sinon il y a itération de **i** par pas de 2 et la boucle recommence.

Illustration :

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

int main()
{
    int somme = 0;
    int i = 0;

    test:
    if (i > 100)
        goto affiche;
    if (i % 2 == 0 && i % 10 != 0) {
        somme += i;
        if (somme > 500)
            goto affiche;
    }
    i += 2;
    goto test;

    affiche:
    printf("La somme des entiers pairs entre 0 et 100 non divisibles par 10 est: %d\n\n", somme);
}
return 0;
```

PROGRAMME 5 :

Le programme présenté vise à résoudre un système d'équations linéaires à deux inconnues dans

R. Le système est représenté par les deux équations suivantes :

$$ax + by = c$$

$$dx + ey = f$$

Dans ce programme l'utilisateur est invité à entrer les valeurs des coefficients **a**, **b**, **c**, **d**, **e** et **f** qui apparaissent dans les deux équations le programme vérifie que l'utilisateur a bien saisi les six coefficients numériques. Si la saisie contient des caractères non numériques ou un nombre

incorrect de valeurs, un message d'erreur est affiché, et le programme se termine car ici le **scanf** est soumis à une condition dans la mesure où l'utilisateur ne doit pas saisir des valeurs différentes des valeurs des coefficients **a, b, c, d, e et f** et d'un saut de page(\n) pour chaque coefficient à l'aide de la variable **extra** (de type char). Le programme utilise les formules de résolution d'un système d'équations linéaires par la méthode d'addition. Il calcule les valeurs de **X** et **Y** (les solutions des équations) en multipliant la première équation par **d** et la deuxième par **-a** la solution obtenue est **Y** et en remplace **Y** dans la première équation pour obtenir **X**, la variable **Z** permet à réduire la longueur de l'affectation des valeurs à **X** et **Y**.

Illustration :

```

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

int main()
{
    float a,b,c,d,e,f,X,Y,Z;
    printf("\t RESOLUTION DANS R D'UN SYSTEME D'EQUATION A DEUX INCONNUES\n\n");
    printf("\t ax + by = c \n \t dx + ey = f \n\n");
    printf("Veuillez saisir svp a, b, c, d, e et f pour la resolution \n");
    char extra;
    if (scanf("%f%c%f%c%f%c%f%c", &a,&extra,&b,&extra,&c,&extra,&d,&extra,&e,&extra,&f,&extra) != 12 || extra != '\n') {
        printf("Erreur : Veuillez saisir seulement des nombres la prochaine fois. Au revoir \n\n");
        return 0;
    }
    Y=(d*c)-(a*f);
    Z=(d*b)-(a*e);
    Y=Y/Z;
    Z=c-(b*Y);
    X=Z/a;
    printf("\t x=%.2f et y=%.2f, l'ensemble de solution est (%.2f;%.2f) \n\n",X,Y,X,Y);
    return 0;
}

```

PROGRAMME 6 :

Ce programme a pour but de demander à l'utilisateur de saisir deux nombres réels et affiche le rapport entre ces deux réel et affiche le résultat de la division ainsi que l'arrondi supérieur et inférieur du résultat. De ce fait on a fait appel aux fonctions **ceil** (pour l'arrondi supérieur) et **floor** (pour l'arrondi inferieur) de la bibliothèque **math.h**. De plus le **scanf** est soumis à une condition dans la mesure où l'utilisateur ne doit pas saisir des valeurs différentes des deux réel et d'un saut de page(\n) pour chaque réel à l'aide de la variable **extra** (de type char). Si le deuxième nombre est zéro, il affiche un message d'erreur car la division par zéro n'est pas possible. Si tout est bon le programme affiche le résultat attendu.

Illustration :

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

int main()
{
    float a, b, R;
    char extra;
    double inf, sup;
    printf("Veuillez saisir deux reels: \n");
    if (scanf("%f%c%f%c", &a, &extra, &b, &extra) != 4 || extra != '\n') {
        printf("Erreur : Veuillez saisir seulement des nombres strictement positifs la prochaine fois. Au revoir \n\n");
        return 0;
    }
    if (b != 0) {
        R = a / b;
        sup = ceil(R);
        inf = floor(R);
        printf("Le rapport entre %.4f et %.4f est : %.4f \a\n", a, b, R);
        printf("L'entier superieur de %.4f est : %.4f \n", R, sup);
        printf("L'entier inferieur de %.4f est : %.4f \n\n", R, inf);
    }
    else
    {
        printf("Entree invalide.\a\n");
    }
    return 0;
}
```

II-TACHE 2

DESCRIPTION DE LA TACHE

On nous demande de Regrouper à l'aide d'un seul programme unique l'ensemble des programmes créés dans la Tache 1 sans faire usage des boucles while, do-while ou de la boucle for dans le cadre de nos travaux

1. Menu Principal :

Pour se faire nous avons fait usage de l'étiquetage en créant une étiquette au nom de **menu** où :

- Le programme affiche un menu avec les six options et un choix pour quitter.
- L'utilisateur est invité à entrer son choix.
- Une validation est effectuée pour s'assurer que l'utilisateur entre un nombre valide entre 0 et 6.

2. Choix de option:

- Nous avons utilisé la structure switch case pour diriger l'exécution vers l'option choisie à l'aide de **goto** pour une précision.
- En cas de saisie invalide, un message d'erreur est affiché, et le menu est réaffiché.

3. Les options:

Chaque programme a été affecté à une étiquète permettant ainsi une bonne fluidité et précision du code.

Illustration :

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
int main() {
    //debut du Menu
    goto menu;
    menu: {
        printf("\t VEUILLEZ CHOISIR LE PROGRAMME QUE VOUS VOULEZ EFFECTUER:\n\n");
        printf("1. Connaitre votre age\n");
        printf("2. Trouver le maximum et le minimum entre trois nombres\n");
        printf("3. Savoir si un entier est premier\n");
        printf("4. Connaitre la somme des entiers pairs entre 0 et 100 non divisibles par 10\n");
        printf("5. Resoudre dans R une equation a deux inconnues\n");
        printf("6. Calculer le rapport entre deux nombres et avoir leurs arrondis\n");
        printf("0. Quitter\n");
        printf(" Merci de faire votre choix\n");
        int choix;
        char extra;
        if (scanf("%d%c", &choix, &extra) != 2 || extra != '\n') {
            printf("Erreur : Veuillez saisir seulement des nombres dans l'intervalle de 1 a 6 la prochaine fois. Au revoir \n\n");
            return 0;
        }
        //fin du Menu
        //debut de la phase cas par cas
        switch (choix)
        {
            case 1:
                goto option1;
                break;
            case 2:
                goto option2;
                break;
            case 3:
                goto option3;
                break;
            case 4:
                goto option4;
                break;
            case 5:
                goto option5;
                break;
            case 6:
                goto option6;
                break;
            case 0:
                goto fin;
                break;
            default:
                {
                    printf("Choix invalide.\n");
                    goto menu;
                }
        }
    }
}
```

III-TACHE 3

DESCRIPTION DE LA TACHE

On nous demande de Regrouper à l'aide d'un seul programme unique l'ensemble des programmes créés dans la Tache 1 en faisant usage des boucles while, do-while ou de la boucle for dans le cadre de nos travaux

1. **Menu Principal :**

Pour se faire nous avons fait usage de la boucle while pour afficher un menu interactif, restant actif tant que l'utilisateur ne choisit pas de quitter avec 0. La condition **choix != 0** assure cette continuité :

- Le programme affiche un menu avec les six options et un choix pour quitter.
- L'utilisateur est invité à entrer son choix.
- Une validation est effectuée pour s'assurer que l'utilisateur entre un nombre valide entre 0 et 6.

2. **Choix de option:**

- Nous avons utilisé la structure switch case à l'intérieur de la boucle pour gérer les différentes options du menu, chaque case correspondant à une action spécifique. Cette structure évite les if imbriqués et rend la gestion des fonctionnalités plus claire et efficace
- En cas de saisie invalide, un message d'erreur est affiché, et le menu est réaffiché.

3. Les options:

Chaque programme a été affecté à une étiquettes permettant ainsi une bonne fluidité et précision du code.

Illustration :

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>

int main()
{
    int choix=1;
    char extra;
    while (choix!=0)
    {
        printf("\t VEUILLEZ CHOISIR LE PROGRAMME QUE VOUS VOULEZ EFFECTUE:\n\n");
        printf("1. Connaitre votre age\n");
        printf("2. Trouver le maximum et le minimum entre trois nombres\n");
        printf("3. Savoir si un entier est premier\n");
        printf("4. Connaitre la somme des entiers pairs entre 0 et 100 non divisibles par 10\n");
        printf("5. Resoudre dans R une equation a deux inconnues\n");
        printf("6. Calculer le rapport entre deux nombres et avoir leurs arrondis\n");
        printf("0. Quitter\n");
        printf(" Merci de faire votre choix\n");
        if (scanf("%d%c", &choix,&extra) !=2 || extra != '\n' ){
            printf("Erreur : Veuillez saisir seulement des nombres dans l'intervalle de 1 a 6 la prochaine fois. Au revoir\n\n");
            return 0;
        }

        switch (choix)
        {
            case 1:
            {
                system("cls");
                char prenom[15], nom[15];
                char car='/';
                int D = 2024, jour, mois, annee;
                float age;

                printf("Veuillez saisir votre prenom(Nous considerons qu'un seul prenom) :\n");
                scanf("%s", &prenom);
                if (isalpha(prenom[0]))
```

CONCLUSION

Ce projet nous a permis d'acquérir des compétences pratiques essentielles et de développer notre capacité à résoudre des problèmes de manière algorithmique. La présentation et la défense de nos programmes constitueront une occasion précieuse de démontrer les connaissances acquises et d'obtenir des retours constructifs pour notre développement continu en programmation.