

Exo 2 : IMPRIMANTE

```
//Evaluation 24/02/2023
//Quintero agustin
//Exercice 2 : imprimantes
#include <iostream>
using namespace std ;

float calculPrix(float imprimente, float prixPage, float nbPages){
    float total = 0;
    total = imprimente + (prixPage*nbPages);
    return total;
}

int main()
{
    //Déclaration des variables
    float loyerImprimant1, prixMonoImprimant1, prixCouleurImprimant1;
    float loyerImprimant2, prixMonoImprimant2, prixCouleurImprimant2;
    float nbPagesMono;
    float nbPagesCouleur;
    float result1, result2;
    //initialisation des prix
    loyerImprimant1 = 149;
    prixMonoImprimant1 = 0.0075;
    prixCouleurImprimant1 = 0.075;

    loyerImprimant2 = 89;
    prixMonoImprimant2 = 0.019;
    prixCouleurImprimant2 = 0.11;

    cout << "Entrez le nombre de page Monochrome a imprimer/mois : " << endl;
    cin >> nbPagesMono;
    cout << "Entrez le nombre de page Couleur a imprimer/mois : " << endl;
    cin >> nbPagesCouleur;

    result1 = calculPrix(loyerImprimant1, prixMonoImprimant1, nbPagesMono) +
    calculPrix(loyerImprimant1, prixCouleurImprimant1, nbPagesCouleur);

    result2 = calculPrix(loyerImprimant2, prixMonoImprimant2, nbPagesMono) +
    calculPrix(loyerImprimant2, prixCouleurImprimant2, nbPagesCouleur);

    if(result1 > result2){
        cout << "L'imprimante la plus interessante pour vous est la HP MFP M
575F, pour un total de : "<< result2 << " euros par mois." << endl;
    }else{
        cout << "L'imprimante la plus interessante pour vous est la HP MFP M
775F, pour un total de : "<< result1 << " euros par mois." << endl;
    }

    return 0;
}
```

Exo 2 : TIMER

```
//Evaluation 24/02/2023
//Quintero agustin
//Exercice 3 : horloge
#include <iostream>
#include <iomanip>
using namespace std ;

int main()
{
    int compteur=0; // 14400 + 420 + 35 = 14855, 1j = 86400 total : 101255
    int jour = 0;
    int time, heures, minutes, secondes;
    while (compteur <= 14855 && jour != 1){
        time = compteur;
        heures = time/3600; // pour trouver les heures
        minutes = (time % 3600) / 60; //les minutes, % pour enlever deja les
heures qui sont comptés
        secondes = time % 60;
        cout << setfill('0') << setw(2) << heures << ":" << setw(2) << minutes
<< ":" << setw(2) << secondes << endl; //setfil de la lib iomanip pour définir
la valeur par rapport aux setw(2) qui dit que si il n'y a qu'un seul nombre, il
formate
        compteur++;
        if (compteur > 86400) {
            compteur = 0; // réinitialiser le compteur à 0 après 24 heures
            jour++;
        }
    }
    return 0;
}
```

Exo 3 : GAME

```
//Evaluation 24/02/2023
//Quintero agustin
//Exercice 4 : game
#include <iostream>
#include <ctime>
#include <cstdlib>
using namespace std ;

int random(int a){
    int randomNb(0);
    srand(time(0));
    randomNb = rand()%a;
    return randomNb;
}

int displaySaisie(){
    int nombre;
    cout << "Veuillez saisir un nombre : " << endl;
    cin >> nombre;
    return nombre;
}

int displayReplay(){
    char choix;
    cout << "Voulez vous rejouer ? (y) ou autre touche" << endl;
    cin >> choix;
    if(choix == 'y' || choix == 'Y'){
        return 'Y';
    }
    return 'N';
}

int displayNbJoueurs(){
    int nombre;
    cout << "Veuillez saisir le nombre de joueur : " << endl;
    cin >> nombre;
    return nombre;
}

string displayNomJoueurs(int position){
    string nom;
    cout << "Veuillez saisir le nom du joueur "<< position+1 <<": " << endl;
    cin >> nom;
    return nom;
}

int findWineur(int nbJoueurs, int scores[100]){
    int min = scores[0];
    int position = 0;
    for(int i = 0; i<nbJoueurs; i++){
        if(scores[i] < min){
            min = scores[i];
            position = i;
        }
    }
    return position;
}

int main()
{
    //déclaration
    char replay;
```

```
do{
    //Déclaration plusieurs joueurs
    string joueurs[100];
    int scores[100];
    int nbParticipants;
    int wineur;
    nbParticipants = displayNbJoueurs();

    for(int i=0; i<nbParticipants; i++){
        joueurs[i] = displayNomJoueurs(i);
    }

    for(int i=0; i<nbParticipants; i++){
        //Déclaration a l'interier d'une partie
        int nombreAtrouver = random(1000);
        int nombreSaisie;
        int score = 0;

        cout << "Au tour du joueur " << i +1 << " de jouer !" << endl;
        do {
            nombreSaisie = displaySaisie();
            if(nombreSaisie < nombreAtrouver){
                cout << "Le nombre est plus grand" << endl;
            }
            if(nombreSaisie > nombreAtrouver){
                cout << "Le nombre est plus petit" << endl;
            }
            score ++;
        }while(nombreSaisie != nombreAtrouver);
        cout << "Bravo tu a gagné" << endl;
        scores[i] = score;
    }
    wineur = findWineur(nbParticipants, scores);
    cout << wineur << endl;
    cout << "Le winer est : " << joueurs[wineur] << " avec un score de " <<
scores[wineur] << endl;
    replay = displayReplay();
}while(replay != 'N');//question 2 replay

    return 0;
}
```