

Corrigé de Devoir d'algorithmique et programmation C++

Exercice 1: Question de cours

1. C'est quoi un code ASCII?
2. Convertir en binaire le nombre -180 en utilisant la méthode de complément à 2.
3. Expliquer le rôle des technologies Intel Hyper-threading et Turbo boost
4. C'est quoi la différence entre SRAM et DRAM?

Corrigé:

1.
Le code ASCII (American Standard Code for Information Interchange) est un système de codage qui permet de représenter les caractères (lettres, chiffres, symboles, commandes) par des nombres.
2.
Convertir en binaire le nombre -180 en utilisant la méthode du complément à 2
 - Conversion de 180 en binaire :
 $180_{10} = 10110100_2$
 - Complément à 1 (inversion des bits) :
 $10110100 \rightarrow 01001011$
 - Complément à 2 (ajout de 1) :
 $01001011 + 1 = 01001100$
 - -180 en binaire (complément à 2 sur 8 bits) = 01001100_2
3.
 - Hyper-threading : Cette technologie permet à un seul cœur de processeur physique de gérer deux flux d'instructions (threads) simultanément, améliorant ainsi l'efficacité du multitâche.
 - Turbo Boost : Elle permet au processeur de fonctionner à une fréquence supérieure à sa fréquence de base de manière automatique lorsque la charge de travail l'exige et que les conditions thermiques le permettent.
4.
La **SRAM** (Static RAM) est très rapide, plus chère et utilisée principalement pour la mémoire cache du processeur. À l'inverse.
la **DRAM** (Dynamic RAM) est moins rapide, moins coûteuse, nécessite un rafraîchissement électrique constant et constitue la mémoire vive principale du système.

Exercice 2 :

Qu'affiche le programme C++ suivant:

```
#include <iostream>
using namespace std;
int main() {
    int i=1, j;
    while(i <=5){
        j= i%3;
        switch(j) {
            case 0: cout<<++j*3<<endl; break;
            case 1: cout<<(i+j++) %2 <<endl;
            case 2: cout<<(i+j) % 4 <<endl;
        }
        i++ ;
    }
    return 0;
}
```

Corrigé:

→ La variable i varie de 1 à 5

À chaque tour :

- on calcule $j = i \% 3$
- on exécute le switch(j)

→ Le programme affiche une valeur par ligne

Détail des itérations

- $i=1 ; j=1$
case 1: affiche $(1+1)\%2 = 0$
j devient 2 Pas de break, à cause de j++
case 2: affiche $(1+2)\%4 = 3$
- $i=2 ; j=2$
case 2: affiche $(2+2)\%4 = 0$
- $i=3 ; j=0$
case 0: affiche $(1 * 3) = 3$.
S'arrête à cause de break
- $i=4 ; j=1$
case 1: affiche $(4+1)\%2 = 1$
j devient 2 Pas de break, à cause de j++
case 2 affiche $(4+2)\%4 = 2$

- $i=5 ; j=2$
case 2 affiche $(5+2)\%4 = 3$

→ affichage final:

```
0
3
0
3
1
2
3

-----
Process exited after 0.2872 seconds with return value 0
Appuyez sur une touche pour continuer... |
```

Exercice 3 :

Écrire un programme C++ qui demande à l'utilisateur de saisir une chaîne de caractères alphabétiques et d'au moins dix lettres. Le programme force l'utilisateur de continuer à refaire la saisie si la chaîne ne respecte pas les deux conditions. Votre programme affiche ensuite le nombre de caractères majuscules et minuscules de la chaîne.

Corrigé:

```
#include<iostream>
#include<cctype>
using namespace std;
int main() {
    string n;
    int maj,min;
    maj=min=0;
    do{
        cout<<"entrez une chaine: ";
        cin>>n;
    }while(n.size()<10);
    for(int i=0;i<n.size();i++)
    {
        char c=n[i];
```

```

    if(isupper(c)) maj++;
    if(islower(c)) min++;
}
cout<<"Majuscules = "<<maj<<endl;
cout<<"Minuscules = "<<min<<endl;
return 0;
}

```

Résultat:

```

entrer une chaine: MohamedMoktarSidi
Majuscules = 3
Minuscules = 14

-----
Process exited after 15.7 seconds with return value 0
Appuyez sur une touche pour continuer... |

```

Exercice 6:

Soit U_n la suite comme suit

- $U_0 = \sqrt{2}$
- $U_1 = 1$
- $U_n = 2 * U_{n-1} + U_{n-2}$, pour tout $n > 1$

1. Donner le prototype de la fonction (U_n) qui retourne U_n
2. Ecrire le code C++ de la fonction l

Corrigé:

```

#include <iostream>
#include <cmath>
using namespace std;
double U(int n) {
    if(n==0) {
        return sqrt(2);
    }
    if(n==1) {
        return 1;
    }
}

```

```

    }
    return 2*U(n-1)+U(n-2);
}

```

Exercice 5 :

Écrire une fonction qui prend comme paramètre un entier n et qui affiche le triangle de Pascal d'ordre n (voir ci-dessous le résultat d'affichage pour $n=5$).

Chaque ligne commence par 1 et se termine par 1. Un élément interne d'une ligne i est obtenu par la somme deux éléments de la ligne $i-1$: l'élément au-dessus et celui qui est à sa gauche. Indication : utiliser une matrice carrée d'entiers d'ordre n .

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

```

Corrigé:

```

#include<iostream>
using namespace std;
void pascal(int n) {
    int T[n][n];
    for(int i=0; i<n; i++){
        for(int j=0; j<=n; j++){
            if(j==0 || j==1){
                T[i][j]=1;
            }
            else
                T[i][j]=T[i-1][j-1]+T[i-1][j];
        }
    }
    for(int i=0; i<n; i++) {
        for(int j=0; j<=i; j++)
            cout<<T[i][j]<<" ";
        cout<<endl;
    }
}

```

```

}
int main() {
    int n;
    cout<<"entrez nombre: ";
    cin>>n;

    pascal(n);

    return 0;
}

```

Résultat:

```

Entrez un nombre: 5
1
1 1
1 1 12113794
1 1 12113795 40591243
1 1 12113796 52705038 94699060

-----
Process exited after 5.605 seconds with return value 0
Appuyez sur une touche pour continuer... |

```