

Exercices du chapitre 1-1

Exercice 1 :

Ecrire un programme qui demande à l'utilisateur de saisir 4 entiers A, B, C et D, puis qui indique quelle est l'intersection des intervalles [AB] et [CD].

Réponse :

```
#include <iostream>

using namespace std;

int main()
{
    int A,B,C,D,tmp,First,Last;

    cout << "Enter la valeur de A : " ;
    cin >> A;
    cout << "Enter la valeur de B : " ;
    cin >> B;
    cout << "Enter la valeur de C : " ;
    cin >> C;
    cout << "Enter la valeur de D : " ;
    cin >> D;

    First=C;
    Last=B;

    if(A>B){
        tmp=A;
        A=B;
        B=tmp;
    }

    if(C>D){
        tmp=C;
        C=D;
        D=tmp;
    }

    if(A>=C) First=A;
    if(B>=D) Last=D;
```

```

if(First<=Last)

    cout<< "L'intersection entre ["<<A<<","<<B<<"] et ["<<C<<","<<D<<"] est l'intervalle
    ["<<First<<","<<Last<<"]"<<endl;

else

    cout<< "Les deux intervalles ["<<A<<","<<B<<"] et ["<<C<<","<<D<<"] n'ont pas
    d'intersection."<<endl;

return 0;

}

```

Exercice 2 :

Ecrire un programme qui permet de calculer la fonction suivante, avec une précision jusqu'à 0.0001 :

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

Avec et Sans utiliser la bibliothèque <cmath>

Réponse :

```

#include <iostream>
#include <iomanip>
#include <cmath>

using namespace std;

/*Sans utiliser la biblio <cmath> on doit definir la fonction puissance()
double puissance(double x, int n){
    if(n==0)
        return 1;
    else
        return x*puissance(x,n-1);
}*/

double factoriel(double x){
    if(x==0 || x==1)
        return 1;
    else
        return x*factoriel(x-1);
}

double calculterm(double x, int n){
    double numerateur,denominateur;

    numerateur = pow(-1,n)* pow(x,2*n +1);

```

```

/*numérateur = puissance(-1,n)* puissance(x,2*n +1);*/
denominateur = factoriel(2*n +1);
return numérateur/denominateur;

}

double fonctionsin(double x){
    double somme=0;
    double term;
    int n=0;
    do{
        term = calculterm(x,n);
        somme = somme + term;
        n++;
    }while(abs(term)>= 0.0001);
    /*while(term >= 0.0001 || term <= -0.0001);*/
    return somme;
}

int main()
{
    double x;
    double resultat;
    cout<<"Donner la valeur de x : ";
    cin>> x;
    resultat = fonctionsin(x);
    double verif = sin(x);
    /*Sans <cmath> on peut pas verifier la valeur de la fonction*/
    cout<<"la valeur de la Serie pour x = "<<x<<" est : "<<resultat<<endl;
    cout<<"la valeur de sinus pour x = "<<x<<" est : "<<verif<<endl;
    return 0;
}

```

Exercices de Atelier 1 : Concepts de bases

Exercice 2 :

Quels résultats fournira ce programme :

```
#include <iostream>
using namespace std ;
main() {
    // cas 1
    int n=10, p=5, q=10, r ;
    r = n == (p = q) ;
    cout << "A : n = " << n << " p = " << p << " q = " << q << " r = " << r << "\n" ;

    // cas 2
    n = p = q = 5 ;
    n += p += q ;
    cout << "B : n = " << n << " p = " << p << " q = " << q << "\n" ;

    // cas 3
    q = n > p ? n++ : p++ ;
    cout << "D : n = " << n << " p = " << p << " q = " << q << "\n" ;

    // cas 4
    n = 5 ; p = 2 ;
    q = ++n == 6 && ++p == 3 ;
    cout << "D : n = " << n << " p = " << p << " q = " << q << "\n" ;
}
```

Réponse :

// cas 1

n=10 p=10 q=10 r=1

// cas 2

n=15 p=10 q=5

// cas 3

n=16 p=10 q=15

// cas 4

n=6 p=3 q=1

Exercice 3 :

Afficher un triangle isocèle formé d'étoiles. La hauteur du triangle (c'est-à-dire le nombre de lignes) sera saisie par l'utilisateur, comme dans l'exemple ci-dessous. On s'arrangera pour que la dernière ligne du triangle s'affiche sur le bord **gauche** de l'écran.

Combien de lignes ? 10

```
      *
     ***
    *****
   ********
  *********
 *****
*****
*****
*****
*****
*****
```

Réponse :

```
#include <iostream>

using namespace std;

int main()
{
    int i,j,l;

    cout << "Donner le nombre des lignes : ";

    cin >> l;

    for(i=1;i<=l;i++){
        for(j=1;j<=l-i;j++){
            cout<<"  ";
        }

        for(j=1;j<=(2*i-1);j++){
            cout<<"* ";
        }

        cout<<"\n";
    }

    return 0;
}
```

Exercice 4 :

On souhaite écrire un algorithme pour simuler un jeu de lancer de dés à deux joueurs. Voici les règles :

- A chaque tour les deux joueurs peuvent choisir de lancer un ou deux dés. Le score est la somme cumulée des lancers de tous les tours de jeu.
- Avant de lancer le dé, le joueur choisit s'il veut lancer 1 ou 2 dés. S'il choisit de lancer deux dés, son score augmentera du résultat de la somme des deux dés. Cependant si les deux dés ont le même résultat son score sera diminué du résultat de la somme des dés.
- Si à l'issue d'un tour, un joueur a le score de 30 ou au-dessus, il remporte la partie.

Q1. Comment simuler un lancer de dés en informatique ?

Q2. Écrire une fonction **throwDice()** permettant de communiquer au reste du programme le résultat d'un lancer de dés.

Q3. Écrire une fonction **playerTurn()** qui correspond au tour d'un seul joueur. Cette fonction va renvoyer l'augmentation ou la diminution du score du joueur à l'issue de ce tour. Elle doit :

- Afficher le message "**Voulez-vous lancer 1 ou 2 dés ?**" et récupérer la réponse de l'utilisateur. Il faudra s'assurer que l'utilisateur donne une valeur cohérente avant de passer à la suite.
- En fonction du choix du joueur simuler le lancer de dés en utilisant la fonction **throwDice()**.
- En fonction du résultat obtenu (cf les règles du jeu), renvoyer au reste du programme l'évolution du score du joueur.

Q4. Dans le programme principal **main()**, écrire le programme permettant le déroulement du jeu :

- Définir et initialiser les variables **score1** et **score2**, respectivement les scores du joueur 1 et 2.
- Le jeu continue jusqu'à ce que le score d'un joueur atteigne 30 à la fin d'un tour. Lors d'un tour, le joueur 1 joue puis le joueur 2 et le score de chaque joueur est affiché. Écrire l'algorithme permettant le déroulement complet d'une partie.
- Lorsque la partie se termine, afficher le joueur gagnant. Il est possible d'obtenir un match nul si les deux joueurs ont un score dépassant 30 à la fin du dernier tour.

Réponse :

```
#include <iostream>

#include <iomanip>

#include <cstdlib>

#include <ctime>

using namespace std;

int throwDice(int NumDice){
    int somme=0,resul;
    if(NumDice==1){
        srand(time(0));
        resul=(rand() % 6) + 1;

        return resul;
    }else{
        resul=(rand() % 6) + 1;
        somme+=resul;
        srand(time(0));
        resul=(rand() % 6) + 1;
        if(resul==somme){
            somme+=resul;
            return -somme;
        }else{
            somme+=resul;
            return somme;
        }
    }
}

int playerturn(int score){
    int N,somme;

    do{
        cout<<"Voulez-vous lancer 1 ou 2 des ? ";

        cin>>N;
    }while(N<1 || N>2);

    somme=throwDice(N);

    if(somme<0){
        cout<<"Malheureusement, vous avez obtenu deux des identiques, votre score va
etre diminue de "<<somme<<endl;

        score+=somme;
    }
```

```

        }else{
            cout<<"Bien joue, vous avez obtenu +"<<somme<<" additionne a votre score"<<endl;
            score+=somme;
        }
    }
    return score;
}

int main()
{
    int score1=0,score2=0,tourscore;
    cout<<"----- Bienvenue au jeu de des -----\n"<<endl;
    do{
        cout<<"Le score du joueur 1 : "<<score1<<endl;
        tourscore=playerturn(score1);
        score1=tourscore;
        cout<<"Le score du joueur 2 : "<<score2<<endl;
        tourscore=playerturn(score2);
        score2=tourscore;
    }while(score1<30 && score2<30);
    if(score1>=30 && score2<30){
        cout<<"\nLe joueur 1 gagnant d'un score : "<<score1<<" , felicitations!"<<endl;
    }else if(score2>=30 && score1<30){
        cout<<"\nLe joueur 2 gagnant d'un score : "<<score2<<" , felicitations!"<<endl;
    }else{
        cout<<"\nLe tour est nul d'un score : "<<score1;
    }
    return 0;
}

```