

But du TD

Comprendre l'utilisation des pointeurs et comment faire le prototype de fonctions en utilisant le passage par pointeur et le passage par référence.

Exercice 1

Comprendre l'utilisation des pointeurs.

Selon l'exemple suivant, indiquez quelles seront les différentes choses affichées par le programme. De plus indiquez quelles sont les lignes qui pourraient poser problème à la compilation ou à l'exécution du programme.

```
int main(){
int *ptrint;
float *ptrfloat;
float *ptrfloattest;

ptrfloattest= new float;
int i=1;
float f=0.5;
ptrfloattest=f; // ne fonctionne pas car f n'est pas un pointeur.
*ptrfloattest=f;
int T[10];
T[0]=1;
float Tf[10];
Tf[0]=0.5;
*ptrfloat=1; // Probleme à la compilation car la case n'est pas allouée
ptrfloat = new float;
ptrfloat= Tf[0]; //Erreur car Tf[0] n'est pas un pointeur.
ptrfloat= &Tf[0];
ptrint= &Tf[0]; // faux car ptrint est de type int et non de type float.
ptrint= &T[0]; //ceci marche même si ptrint n'est pas alloué

for (int j=0;j<10;j++)
{
    ptrfloat= ptrfloat+f; //ne fonctionne pas car f est un float;
    ptrfloat= ptrfloat+1;
    *(ptrfloat)= *(ptrfloat-1)+j;// ne marchera pas à la dixième itération car
ptrfloat sort du tableau. Il faut faire attention a aller jusqu'à j<9.
}

for (j=0;j<10;j++)
{
    *(ptrfint+j)= *(ptrfint-j); //pas de problème car on peut toujours accéder en
lecture mais on ne saura jamais quelles valeurs ont été copiées.
}
*ptrfloat =-10 ;// erreur car maintenant ptrfloat est sorti du tableau;
```

```

for (int j=0;j<10;j++)
{
    // Ici il faudra faire attention de bien faire un # include<iostream>; using
    namespace std; sinon ca ne marchera pas
    cout << Tf[j]; //devrait afficher 0.5 0.5 1.5 3.5 6.5 10.5 15.5 21.5 28.5 36.5
    cout << T[j]; // On ne peut pa savoir ce qui sera affiché
}

cout <<*(ptrfloat+1); // il va afficher la valeur de la case a coté de T[9] mais on
ne sait pas ce que ce sera.
cout <<ptrint; //affiche l'adresse vers laquelle point ptrint qui correspond
toujours a l'adresse &T[0]
cout <<*ptrint; //affiche 1
cout <<*ptrfloattest; //affiche 0.5
delete ptrfloat; // ne fonctionne pas car ptrfloat ne pointe plus sur sa case de
départ. Si on lan ce le programme trop de fois, alors il y aura un problème de
mémoire car on alloue une case mémoire pour ptrfloat mais on ne peut jamais la
désallouer.

delete ptrint; // ne fonctionne pas car ptrint n'a pas été alloué.

delete ptrfloattest; // ne fonctionne pas car ptrint n'a pas été alloué. }

```

Correction exercice 1

```
int main(){
    int *ptring;
    int j;
    float *ptrfloat;
    float *ptrfloattest;

    ptrfloattest= new float;

    float f=0.5;

    *ptrfloattest=f;

    int T[10];
    T[0]=1;
    float Tf[10];
    Tf[0]=0.5;

    ptrfloat= &Tf[0];
    ptring= &T[0];

    for (j=0;j<9;j++)
    {ptrfloat= ptrfloat+1;
    *(ptrfloat)= *(ptrfloat-1)+j;
    }

    for(j=0;j<10;j++)

    {
        *(ptring+j)= *(ptring-j);
    }

    for (int j=0;j<10;j++)
    { cout << Tf[j]<<"\t"<< T[j]<<endl;
    }
    cout <<endl<< "\t"<< *(ptrfloat+1)<<"\t"<<ptring<<"\t" <<*ptring<<"\t"<<*ptrfloattest<<endl;

    delete ptrfloattest;
    return 0;
}
```

0.5 1
0.5 0
1.5 0
3.5 1108475904
6.5 1105461248
10.5 1101791232
15.5 1098383360
21.5 1093140480
28.5 1087373312
36.5 1080033280

0 0x7fff5fbff680 1 0.5
Program ended with exit code: 0

Exercice 2

Cet exercice sert à comprendre le passage par pointeur et par référence.
Dans cet exercice, il est demandé:

- Réaliser la déclaration des fonctions et du main pour une fonction échange avec passage par paramètre et référence qui échange une variables avec une autre.

Solution Exercice 2

```
void echange1(int a, int b)
{ int temp;
  temp=a;
  a=b;
  b=temp;
}
void echange2(int* a, int* b)
{ int temp;
  temp=*a;
  *a=*b;
  *b=temp;
}
void echange3(int &a, int &b)
{ int temp;
  temp=a;
  a=b;
  b=temp;
}

int main(){
  int a=4;
  int b=0;
  int *ptrb;

  ptrb=&a;

  int *ptrb ;
  ptrb=new int;
  *ptrb=0;

  cout<<"a=" <<a<<endl;
  cout<<"b="<<b<<endl;

  echange1(a,b);
  cout<<"Echange sans passage par parametre ne fonctionne pas"<<endl;
  cout<<"a=" <<a<<endl;
  cout<<"b="<<b<<endl;

  echange3(a,b);
  cout<<"Passage par reference"<<endl;
  cout<<"a=" <<a<<endl;
  cout<<"b="<<b<<endl;
  cout<<"Passage par pointeur 1"<<endl;
  echange2(&a,&b);
  cout<<"a=" <<a<<endl;
  cout<<"b="<<b<<endl;

  echange2(ptra,ptrb);
  cout<<"Passage par pointeur 2: attention ptrb ne pointe pas sur b "<<endl;
  cout<<"a=" <<a<<endl;
  cout<<"b="<<b<<endl;
  cout<<"ptrb="<<*ptrb<<endl;

  delete ptrb;
```

```
a=4
b=0
Echange sans passage par parametre ne fonctionne pas
a=4
b=0
Passage par reference
a=0
b=4
Passage par pointeur 1
a=4
b=0
Passage par pointeur 2: attention ptrb ne pointe pas sur b
a=0
b=0
ptrb=4
Program ended with exit code: 0
```

Exercice 3

Cet exercice sert à comprendre le passage par pointeur et par référence ou par fonction.
Dans cet exercice, il est demandé de

- Réaliser la déclaration et le main pour une fonction qui renvoie le min et le max d'un tableau de taille N avec passage par référence, par pointeur et par fonction.

```

-----
void computeminmax(int tab[10], int* vmax,int* vmin)
{int i;
  *vmax=tab[0];
  *vmin=tab[0];
  for(i=0;i<10;i++)
  {
    if(*vmax<tab[i]){*vmax=tab[i];}
    if(*vmin>tab[i]){*vmin=tab[i];}
  }
}

```

//ici on pourra déclarer un tableau dynamiquement quelque soit la taille
 //de plus on peut surcharger la fonction computeminmax sachant que les par

```

void computeminmax(int *tab,int taille, int& vmax,int& vmin)
{int i;
  vmax=*tab;
  vmin=*tab;
  for(i=0;i<taille;i++)
  {
    if(vmax<*(tab+i)){vmax=*(tab+i);}
    if(vmin>*(tab+i)){vmin=*(tab+i);}
  }
}

```

//on peut ici déclarer computemax comme un int car on ne retourne qu'une s

```

int computemax(int* tab,int taille)
{int i;
  int res=*tab;
  for(i=1;i<taille;i++)
  {
    if(res<*(tab+i)){res=*(tab+i);}
  }
  return res;
}

```

```

int main(){

    int T[10]; //T sera quand même un pointeur
    int *Tptr;
    int n;
    int i;

    int Vmax;
    int *Vmin;
    Vmin=new int;
    cout<<"n?"<<endl;
    cin>>n;
    Tptr=new int[n];

    for(i=0;i<10;i++)
    {
        T[i]=2*i;
    }
    for(i=0;i<n;i++)
    {
        *(Tptr+i)=-3*i;
    }

    //la bonne fonction est appelée grace à la surcharge
    computeminmax(T,&Vmax,Vmin); //pointeur
    cout << "T fixe, passage par pointeur:"<<Vmax<<","<<*Vmin<<endl;
    computeminmax(Tptr,n,Vmax,*Vmin); //reference
    cout <<"T dynamique,passage par reference:"<<Vmax<<","<<*Vmin<<endl;
    computeminmax(T,10,Vmax,*Vmin);
    cout <<"T dynamique,passage par reference:"<<Vmax<<","<<*Vmin<<endl;
    Vmax=computemax(T,10);
    cout <<"T fixe,passage par fonction:"<<Vmax<<endl;
    Vmax=computemax(Tptr,n);
    cout <<"T dynamique,passage par fonction:"<<Vmax<<endl;

    delete [] Tptr;
    //delete T; pas besoin de désallouer T
    delete Vmin;
    return 0;
}

n?
6
T fixe, passage par pointeur:18,0
T dynamique,passage par reference:0,-15
T dynamique,passage par reference:18,0
T fixe,passage par fonction:18
T dynamique,passage par fonction:0
Program ended with exit code: 0

```