

Nom et prénom : _____

1. (2pts) Quels des types suivantes n'est pas un type valide en C++ ?
 - a. **float**
 - b. **real**
 - c. **double**
 - d. **bool**

2. (2pts) Lesquelles des déclarations de variables suivantes ne sont pas valides en C++ :
 - a. `int $main`
 - b. `double 2_fois`
 - c. `char _3type`
 - d. `bool *fois`

3. (2pts) Lequel des opérateurs est le « et » logique en C++ :
 - a. `&`
 - b. `||`
 - c. `&&`
 - d. `+`

4. (2pts) Ecrire une fonction *puiss* qui prend en argument un nombre réel x (à virgule flottant) et un entier n et calcul x^n , sans utiliser la fonction `cmath` (`pow`, `exp`, etc...) de la librairie. Afficher le résultat dans la fonction `main`.

5. (4pts) Ecrire une fonction « **bool** *triangle(...)* » qui prend en argument trois nombres réels, a , b , c et qui détermine si ces trois nombres peuvent correspondre aux trois côtés d'un triangle. La fonction va retourner une valeur de « true » s'il s'agit d'un triangle valide et « false » sinon. (La condition pour avoir un triangle valide est « $a+b>c$ et $a+c>b$ et $b+c>a$ »). Afficher dans la fonction `main` si les valeurs correspondent oui ou non à un triangle possible.
 - a. Faire en sorte que la fonction affiche la condition qui n'est pas satisfaite quand les conditions pour former un triangle ne sont pas remplies. (par exemple si $a = 1$, $b = 3$, $c = 0.5$) le programme affiche « $1 + 0.5$ n'est pas supérieur a 3 ».
 - b. Ajouter un argument à la fonction *triangle* du type caractère qui prendra la valeur 'e' s'il s'agit d'un triangle équilatéral, 'i' s'il s'agit d'un triangle isocèle, et 's' s'il s'agit d'un triangle scalène.

6. (4pts) Ecrire un programme pour traiter le tableau de « notes », ci-dessous. La note minimale possible est « 0 », mais une valeur de « -1 » correspond à une absence ou défaillance. Votre programme devrait déterminer le nombre de notes valides et remplir un tableau composé exclusivement des notes valides. Votre fonction `main()` devrait prendre la forme suivante :

```
{
    const int NDIM=21;
    int notes[NDIM]={13,-1,0,19,10,0,17,9,-1,2,6,9,0,4,12,14,18,3, 5,6, -1} ;
    int valides[NDIM];

    cout << "Le tableau a " << NDIM << " valeurs, mais il n y a que "
    << nvalide << " notes valides " << endl;
}
```

- Ecrire une fonction `affiche(NDIM, notes)` qui prend un tableau et la taille du tableau en argument et qui affiche les valeurs du tableau.
- Ecrire une fonction `nvalide = create_valide(NDIM, notes, valides)` qui va retourner la dimension des notes valides et qui va remplir le tableau *valides* constitué uniquement des notes valides. (par exemple nous aurons dans cet exemple `nvalide = 17` et `valides = {13,0,19,10,0,17,9,2,6,9,0,4,12,14,18,3,5,6}`)
- Ecrire une fonction `remplacer(NDIM, notes, val_vieux, val_nouv)` qui va remplacer chaque occurrence d'une valeur, `val_vieux`, et qui va la remplacer par une valeur `val_nouv`. (par exemple `remplacer(NDIM, notes, -1, 0)` produira un tableau : `notes = {13,0,0,19,10,0,17,9,0,2,6,9,0,4,12,14,18,3, 5,6,0}`);
- Bonus** : Ecrire une fonction `trinotes()` qui mettra les éléments d'un tableau dans un ordre croissant :

7. (4pts) Ecrire une fonction, « `int plus_grand_div(int){ ...}` », qui trouve le plus grand diviseur d'un entier. Votre fonction `main()` devrait prendre la forme suivante :

```
{ int nomb ,div;
    cout << "Entrez un nombre " << endl ; cin >> nomb ;
    div = plus_grand_div(nomb) ;
    cout << "Le plus grand diviseur de " << nomb << " = " << div << endl ;
```

- Ecrire une fonction, « `int premiers(int){ ...}` », qui décompose n'importe quel nombre en facteurs premiers et affiche le résultat. (Ex : $100=2*2*5*5$, $21=3*7$)