

Nom et prénom : _____

1. (2pts) Quel type utilise-t-on pour une fonction qui retourne une valeur du type logique ?
 - a. **int**
 - b. **logic**
 - c. **double**
 - d. **bool**

2. (2pts) Le compilateur va lire votre programme ligne par ligne afin de générer un fichier .obj. Indiquer le nombre de fois le compilateur peut lire votre programme avant de pouvoir générer le fichier .obj.
 - a. 1 fois
 - b. 2 fois
 - c. 3 fois
 - d. autant de fois que nécessaire

3. (2pts) Les arguments d'une fonction et les paramètres dans sa déclaration doivent être :
 - a. du même type seulement
 - b. du même nombre seulement
 - c. de nombre et de types différents.
 - d. du même nombre et du même type

4. (2pts) On aimerait avoir un programme qui affiche les entiers impairs entre 2 et 15 (les bornes y comprises).

```

...
int i= 2, nmax=15;

do{
    rest = i%2;
    if(rest=1) cout << " i = " << i << endl;
    i++;
} while(i<nmax) ; ...
    
```

- a. Indiquer le/les fautes de programmation dans ce code.

- b. Réparer le code avec un minimum de changements afin que l'exécutable fonctionne correctement.

5. (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 ;
    
```

 - a. **Bonus** : 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)

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 calculer leur moyenne (sans compter les « -1 »). Elle doit également afficher la plus grande et plus petite des notes. Votre fonction `main()` devrait prendre la forme suivante :

```
{
    const int NDIM=21;
    int notes[NDIM]={13,-1,4,19,10,0,17,15,11,2,6,9,8,4,12,14,18,3, 5,6, -1} ;
    int valmin, valmax, nvalide,
    double moyenne;
    ....

    cout << "Le tableau a " << NDIM << " valeurs, mais il n y a que "
    << nvalide << " notes valides " << endl;
    cout << " La moyennes des notes valides = " << moyenne << endl;
    cout << " La plus petite note = " << valmin <<
    " et la plus grande note = " << valmax << endl ;
}
```

- Mettez le calcul de toutes les valeurs (`valmin`, `valmax`, `moyenne`, `nvalide`) dans une fonction `stat_tab` qui vous permettra de remplacer le «...» par :
`nvalide = stat_tab(notes,NDIM,valmin,valmax,moyenne);`
- Vérifier que votre programme fonctionne correctement avec le tableau suivant :
`notes[]={-1,-1,4,19,10,0,17,15,11,2,6,9,8,4,12,14,18,3, 6,5, -1} ;`
- Ajouter une fonction** `saisi_tab(notes[])` permettant un utilisateur de remplir le tableau de notes (si l'utilisateur entre moins de 21 notes, les éléments restants du tableau auront tous la valeur `-1`).
- Bonus** : Faites en sorte que `stat_tab` puisse également calculer l'écart type des notes.

7. (4pts) Ecrire une fonction « `void arrange_case(char ch[]){ ...}` » qui transformera tous les caractères majuscules en caractères minuscules (sauf le premier caractère de la phrase qui sera majuscule). Votre fonction `main` devrait avoir l'allure suivante :

```
{
    char phrase[]=" ceTTe PhRase AVAIt un GrAnD PrOBLeMe dE MiNISCuLes ET
    MaJuscULES .";
    cout << "Avant on avait : " << endl << phrase << endl ;
    arrange_case(phrase);
    cout << " Apres arrangecase on a : " << endl << phrase << endl;
    ...
}
```

- Bonus** : Ecrire une fonction `enleve_espace(char ch[])` qui va enlever les espaces en trop dans la phrase (telle qu'il n'existera plus qu'un seul espace entre les mots, et que le premier caractère de `phrase` soit une lettre alphanumérique).