

Contrôle individuel : Adaptation langage C++ -- 24 Octobre 2014

Nom et prénom : _____



1. Quel type utilise-t-on pour une fonction qui ne retourne pas de valeur ?
 - a. int
 - b. prototype
 - c. void
 - d. double
2. Les arguments d'une fonction et les paramètres dans sa déclaration doivent être :
 - a. Du même nombre seulement
 - b. Du même nombre et du même type
 - c. Du même type seulement
 - d. De nombre et de types différents.
3. Instructions **if-else** ;
 - a. Ré-écrire les instructions **if-else** imbriquées dans ce qui suit comme une seule instruction **if-else** (non-imbriquée) qui fait la même chose.
C'est-à-dire, on remplace les lignes entre ... et

```
bool bonnes_notes, felicitations, bon_etudiant;  
double notes ;
```

...

```
if (bonnes_notes)  
{  
    if (felicitations)  
    {  
        bon_etudiant = true;  
    }  
    else bon_etudiant = false;  
}  
else bon_etudiant = false;
```

...

```
if(bon_etudiant) cout << " Bravo :) "; else cout << " Bof :( ";
```

- b. Bonus : Est-ce que vous pouvez remplacer les lignes entre ... et ... par une seule instruction qui n'utilise pas une instruction **if-else** ?

4. Les éléments de programme ci-dessous comportent un tableau oct de dimension 31 contenant les températures maximales atteintes chaque jour du mois.
Ecrire une fonction `njours_chaud` qui reçoit 3 arguments : le tableau oct et les entiers `NJOURS` et `temp`. Elle retournera le nombre de jours dans le mois où la température maximale a atteint une température de `temp` ou plus.

```

const int NJOURS=31;

int day,temp,jours_chaud = 0;

int oct[NJOURS]={20,18,25,15,10,12,16,19,21,24,20,16,15,12,10,8,-1,-2,
                -1,0,1,2,4,5,8,10,12,14,16,18,19};

cout <<"Entrez une temperature " ; cin>> temp ;

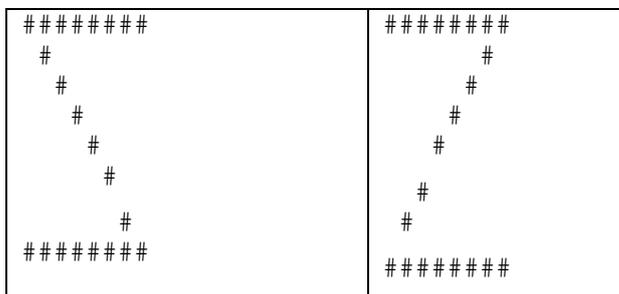
jours_chaud = njours_chaud(oct, NJOURS, temp);

cout <<"Il y avait " << jours_chaud << " jours dans le mois " << " ou il
faisait au moins " << temp << " degres" << endl;

... ..

```

5. Imprimer à l'écran les motifs suivants en utilisant des boucles imbriquées. (Essayer d'utiliser un nombre minimal de boucles).



6. Le code C++ ci-dessous utilise des variables de type **double** x, y, z. Il trouve la racine carrée d'un nombre val, en rapprochant de façon répétée x et y tels que $x*x < val$ et $y*y > val$ jusqu'à ce que les carrés de x et y soient si près de val que l'un comme l'autre puisse être utilisé comme la racine carrée de val.

- a. Compléter l'instruction composée {...} du code ci-dessous par une boucle **do-while** qui s'exécute jusqu'à ce que la différence y-x soit plus petite que $PREC = 0.001$.

```

const double PREC = 0.001 ;
double x = 0.0, y ;

cout <<"Entrez une valeur " ; cin >> val ;

y = val ;

{
  z = (x + y) / 2.0;
  if (z*z > val)
    y = z;
  else
    x = z;
}

```

- b. Utiliser ces lignes afin d'écrire une fonction **racine** qui retourne la valeur de la racine carrée.