

Option Programmation I

- Évaluation -

I – Variables et opérations

Évaluer les variables suivantes. Préciser les « . » et « f » au besoin :

```
int a = 3.1415f;  
a vaut :
```

```
float b = (int)4.2f;  
b vaut :
```

```
char c = 256;  
c vaut :
```

```
int d = 100;  
d += (50 + d);  
d vaut :
```

```
bool e = !(1 < 2);  
e vaut :
```

```
bool f = false || (true && (true || false) ) && true;  
f vaut :
```

```
bool g = ((5%3) == 1);  
g vaut :
```

```
int h;  
h = h * 2 - 5;  
h vaut :
```

```
int tab[5] = {1,2,3,4,5};  
tab[2] vaut :
```

```
std::string s = "a";  
s = s + "b" + "c\nd" + "ef\0g";  
std::cout << s; affiche :
```

II – Structures de contrôle

1 - Ecrire un code déclarant un tableau de 10 variables de type int et initialisant chaque sous-variable du tableau à sa valeur d'index correspondante (utiliser un for ou un while) :

2 – Déclarer deux variables, "somme" et "i", de type int initialisées à 0. Ecrire une structure de contrôle "while" qui incrémente "i" à chaque boucle puis qui additionne "i" à "somme" :
 $0+1+2+3+4+\dots$ La boucle s'arrête quand "somme" est strictement supérieure à 100.

III – Fonctions

1 - Ecrire une fonction "Abs" qui prend en paramètre une valeur de type int et qui a comme valeur de retour la valeur positive de ce paramètre. Par exemple : Abs(7) renvoie 7, Abs(-3) renvoie 3.

2 - Utiliser cette fonction dans une ligne d'instructions et stocker son résultat dans une variable.

IV – Exercice

Ecrire une fonction "Attack" qui calcule les dégâts d'une attaque d'un pokémon sur un autre.

- Paramètres en entrée : Type de l'attaquant (std::string), puissance de l'attaquant (int), type du défenseur (std::string)
- Valeur de retour : nombre de dégâts (int)

Le calcul de dégâts respecte la règle ci-dessous. Dans tous les autres cas, le nombre de dégâts vaut la puissance de l'attaquant. (Note : les std::string se comparent avec ==)

Attaquant	Défenseur	Dégats
"electric"	"water"	power * 2
"water"	"fire"	power * 2
"fire"	"water"	power / 2

Pour un index i, les tableaux ci-dessous donnent respectivement le nom, le type et la puissance d'un pokemon.

- Utiliser la fonction ci-dessus pour calculer les dégâts de Pikachu sur Carapuce.
- Afficher le résultat en console
- Ajouter le pokémon de votre choix en dernière position des tableaux

```
std::string names[4] = { "Pikachu", "Carapuce", "Salameche", "" };
std::string types[4] = { "electric", "water", "fire", "" };
int powers[4] = { 100, 40, 60, 0 };
```