

Les fonctions

1. Traitement sur les points

Le but de ce programme est de trouver parmi un nuage de point celui qui est le plus proche du barycentre des points, au sens de la norme euclidienne.

- Créez une fonction qui génère un ensemble de point
- Créez une fonction qui trouve le barycentre des points
- Affichez les coordonnées du point le plus proche du barycentre des points

Voir

[http://fr.wikipedia.org/wiki/Norme_\(mathematiques\)](http://fr.wikipedia.org/wiki/Norme_(mathematiques))

2. Calcul Intégral

Le but de cet exercice est de calculer une approximation de l'intégrale de la fonction

$$f: R \Rightarrow R$$

$$f(x) = x^2 + 3$$

en utilisant la méthode des rectangles.

Cette méthode est basée sur une approximation de f par une fonction en escalier, avec n 'marches' de longueur h . La valeur approchée R de l'intégrale est

$$R = h * \sum_{i=0}^{n-1} f(x_i)$$

avec $h = \frac{b-a}{n}$ et $x_i = a + i * h$

Calculez R pour x variant de 1 à 4.

3. Suite numérique linéaire à coefficients constants

On considère la suite u_n défini ci-dessous, pour $n > 0$

$$\begin{cases} u_0 = 0 \\ u_n = \frac{1}{2}u_{n-1} + 3 \end{cases}$$

1°) Écrire une méthode prenant en entrée un indice n et retournant le terme $u(n) = u_n$

2°) Écrire une méthode calculant le plus petit indice n tel que

$$u_n - u_{n-1} \leq 10^{-4}$$

4. Automate

Vous aurez besoin d'utiliser les conditions pour cet exercice,

```
age := 9
if age < 17 { // vérifier si l'utilisateur à au moins 18 ans
    fmt.Println("Sortez !")
} else { // si ce n'est pas le cas alors on l'accepte pas
    fmt.Println("Entrez :)")
}
```

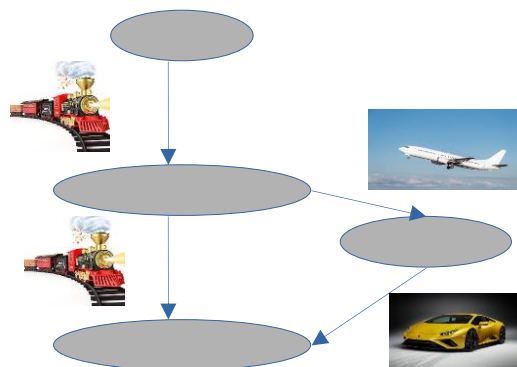
ainsi que la lecture d'un caractère au clavier

```
var character rune
fmt.Scanf("%c", &character)
fmt.Println(character)
```

Il faudra aussi rajouter le fichier launch.json dans le répertoire .vscode

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Launch Package",
      "type": "go",
      "request": "launch",
      "mode": "auto",
      "program": "${workspaceFolder}",
      "console": "integratedTerminal"
    }
  ]
}
```

Nous allons reprendre l'automate du cours



Comme dans la machine de Turing, nous allons demander au programme d'analyser la séquence de voyage :

train, avion, train
que nous allons modéliser par la chaîne de caractère

tat
Le programme devra nous dire si cette séquence est valide, ou bien si un moyen de transport n'est pas adapté au lieu où se trouve le voyageur. Par exemple

vt
est invalide, car à ailes, je ne peux pas prendre la voiture.

Il existe plusieurs méthodes pour implanter ce type de logiciel. Le plus simple est d'avoir des fonctions qui portent le nom de chaque ville. Une fonction associée à la ville va vérifier que le moyen de transport est valide, et nous fera transiter vers une autre ville en appelant la fonction qui porte le nom de la prochaine ville.