

ICC (SMA/SPH) / DEVOIR NOTÉ :

La descente d'un plongeur

Introduction et instructions générales

Ce devoir noté consiste à écrire un programme permettant de calculer les paramètres de la plongée d'un plongeur en apnée.

Nous vous demandons d'écrire tout votre code dans un seul fichier nommé `plongee.cc` sur la base d'un fichier fourni, à compléter en utilisant le langage C++ et une approche « procédurale » avec le moins de duplication de code possible.

Instructions :

1. Cet exercice doit être réalisé **individuellement** ! L'échange de code relatif à cet exercice noté est **strictement interdit** ! Cela inclut la diffusion de code sur le forum ou sur tout site de partage.
Le code rendu doit être le résultat de *votre propre production*.
Le plagiat de code, de quelque façon que de soit et quelle qu'en soit la source, sera considéré comme de la tricherie.
En cas de tricherie, vous recevrez la note « NA » pour l'entièreté de la branche et serez de plus dénoncé(e) et puni(e) suivant l'ordonnance sur la discipline.
2. Vous devez donc également garder le code produit pour cet exercice dans un endroit à l'accès strictement personnel.
Le fichier (source !) `plongee.cc` à fournir comme rendu de ce devoir ne devra plus être modifié après la date et heure limite de rendu.
3. Veillez à rendre du code *anonyme* : **pas** de nom, ni de numéro SCIPER, ni aucun identifiant personnel d'aucune sorte !
4. Utilisez le site Moodle du cours pour rendre votre exercice.
Vous avez jusqu'au **mercredi 30 novembre, 23h59** (heure du site Moodle du cours faisant foi) pour soumettre votre rendu.
Aucun délai supplémentaire ne sera accordé.

Indication : Si un comportement ou une situation donnée n'est pas définie dans la consigne ci-dessous, vous êtes libre de définir le comportement adéquat. On considérera comme comportement adéquat toute solution qui ne viole pas les contraintes données et qui ne résulte pas en un crash du programme.

1 Modélisation du plongeur et affichage

Un plongeur est représenté par son nom, sa masse, son volume (qui changera lorsque le plongeur gonflera son ballon de remontée), son accélération, sa profondeur, sa vitesse et le temps de plongée.

Dans le fichier `plongee.cc` fourni :

1. commencez par définir un type `Conditions` pouvant représenter, dans cet ordre, une profondeur, une vitesse et un temps, tous `double`; ce type nous servira à représenter à la fois les conditions initiales et les conditions courantes d'un plongeur ;

2. définissez ensuite un type `Diver` correspondant à la description donnée ci-dessus (et en utilisant le type `Conditions`);
ce type est utilisé dans le `main()` fourni pour créer une variable s'appelant `jacques`, de masse 90 kg et de volume 0.075 m^3 à la profondeur -1.2 m de vitesse 0.8 ms^{-1} , et 0 pour l'accélération et le temps;
veuillez **ne pas modifier** cette ligne d'initialisation (déclarez vos types pour qu'ils soient compatibles);
3. définissez ensuite une fonction `display()` qui prend en paramètre un plongeur et affiche les informations suivantes en respectant *strictement* le format suivant :
temps, profondeur, vitesse, accélération
Avec les valeurs initiales données ci-dessus, la fonction `display()` affiche :
0, -1.2, 0.8, 0
4. définissez enfin une fonction `dive()` qui prend un plongeur et, pour le moment, ne fait qu'afficher les informations ci-dessus à l'aide de la fonction `display()`.

Testez votre programme.

2 La descente en profondeur

Nous nous intéressons maintenant à faire évoluer le plongeur. Définissez pour cela une fonction `evolve()` qui prend en paramètres (dans cet ordre) un plongeur et des conditions initiales, et qui :

1. ajoute 1 au temps de plongée du plongeur ;
2. puis fait évoluer son état suivant les équations données ci-dessous.

Pour calculer les équations de l'évolution du plongeur, nous avons besoin des expressions suivantes :

- la constante $g = 9.81 \text{ ms}^{-2}$;
- un coefficient $\mu = 1 - 1000 \cdot \frac{\text{volume}}{\text{masse}}$ (qui représente la réduction de poids due à la poussée d'Archimède; avec « masse » la masse du plongeur et « volume » son volume);
- un « terme α » valant $\exp(t_0 - t)$, où t représente le temps courant et t_0 le temps initial de plongée.

Les équations de l'évolution du plongeur s'expriment alors comme :

$$\begin{aligned} v(t) &= \alpha(t) \cdot v_0 + (1 - \alpha(t)) \cdot g \cdot \mu &= \alpha(t) \cdot (v_0 - g \cdot \mu) + g \cdot \mu \\ a(t) &= g \cdot \mu - v(t) &= -\alpha(t) \cdot (v_0 - g \cdot \mu) \\ p(t) &= p_0 + (v_0 - g \cdot \mu) \cdot (\alpha(t) - 1) + (t_0 - t) \cdot g \cdot \mu \end{aligned}$$

où $v(t)$ est la vitesse du plongeur, $p(t)$ sa profondeur, $a(t)$ son accélération, et p_0 , v_0 et t_0 correspondent aux conditions initiales (profondeur, vitesse et temps).

Complétez ensuite la fonction `dive()` de sorte à calculer l'évolution de la plongée : faites le calcul, pas à pas, tant que le plongeur n'a pas atteint la profondeur de -40 m .

Testez votre programme ; il devrait donner les résultats suivants :

```
0, -1.2, 0.8, 0
1, -2.30718, 1.32782, 0.307179
2, -3.748, 1.522, 0.113005
...
23, -37.97, 1.635, 8.56866e-11
24, -39.605, 1.635, 3.15224e-11
```

3 Profondeur désirée et vitesse limite

On vous demande maintenant d'étendre votre fichier `plongee.cc` de sorte que :

- dès que son accélération est inférieure à 10^{-5} m/s² (1e-5), le programme affiche (en plus, mais qu'une seule fois) le message au format suivant :

```
## (t = <TIME> s) <NAME> a atteint sa vitesse maximale (<ACCELERATION> ms-2)
```
- dès que sa profondeur est inférieure à -40 m, le programme affiche (en plus, mais qu'une seule fois) le message au format suivant :

```
## (t = <TIME> s) <NAME> a atteint la profondeur désirée (<DEPTH> m)
```

Nous vous demandons pour cela :

- de définir un type `Status` comprenant (dans cet ordre) de quoi représenter
 - si la profondeur désirée a été atteinte;
 - si la vitesse maximale (accélération inférieure à 10^{-5} m/s²) a été atteinte;
 - si le ballon (question suivante) a été gonflé;
- de définir une fonction `message()` qui reçoit en paramètres (et dans cet ordre) : un plongeur, un message (`string` ou `string_view` suivant votre niveau), une valeur (`double`) et des unités (`string` ou `string_view` suivant votre niveau), et qui affiche, sur une ligne, un message respectant *strictement* le format suivant :

```
## (t = <TIME> s) <NAME> <MESSAGE> (<VALUE> <UNITS>)
```

 (voir des exemples ci-dessus et ci-dessous);
- de définir une fonction `update_status()` qui reçoit en paramètres (et dans cet ordre) : un plongeur, des condition initiales et un `Status`, et qui fait toutes les mises à jour nécessaires, ainsi qu'afficher les messages qu'il se doit;
- de faire les modifications nécessaires à `dive()`.

Pour tester : avec les valeurs précédentes, la vitesse maximale est atteinte au bout de 12 s ($\simeq 1.6$ m/s) et la profondeur désirée au bout de 24 s :

```
11, -18.35, 1.63499, 1.39459e-05
## (t = 12 s) Jacques a atteint sa vitesse maximale (5.13042e-06 ms-2)
12, -19.985, 1.63499, 5.13042e-06
...
24, -39.605, 1.635, 3.15224e-11
## (t = 25 s) Jacques a atteint la profondeur désirée (-41.24 m)
```

4 Gonflage du ballon et remontée

On vous demande enfin d'étendre les fonctions `update_status()` et `dive()` de sorte que dès que la profondeur du plongeur est inférieure à -43 m, le programme augmente le volume de ce plongeur de 0.0375 m³ (gonflage de son gilet/ballon). Il faut alors aussi changer les « conditions initiales » t_0 et p_0 avec les valeurs courantes correspondantes et mettre v_0 à zéro (de sorte que les équations d'évolution soient correctes pour la remontée).

Il faudra de plus changer la condition d'arrêt dans `dive()` pour que la simulation termine dès que le plongeur a atteint la surface (c.-à-d. profondeur positive ou nulle).

Enfin, le programme doit afficher, une seule fois, au moment approprié, chacun des messages au formats suivants :

```
## (t = <TIME> s) Jacques s'arrête et gonfle son ballon (<DEPTH> m)
...
## (t = <TIME> s) Jacques est de retour à la surface (<DEPTH> m)
```

Pour tester : avec les valeurs précédentes, le ballon est gonflé au bout de 27 s et la simulation se termine au bout de 46 s :

```
...
26, -42.875, 1.635, 4.26618e-12
## (t = 27 s) Jacques s'arrête et gonfle son ballon (-44.51 m)
27, -44.51, 1.635, 1.56944e-12
...
46, -0.365, -2.4525, -1.37409e-08
## (t = 47 s) Jacques est de retour à la surface (2.0875 m)
```

Si nécessaire, nous donnons l'affichage complet attendu en page suivante.
Mais votre programme sera bien sûr aussi testé avec d'autres valeurs.

0, -1.2, 0.8, 0
1, -2.30718, 1.32782, 0.307179
2, -3.748, 1.522, 0.113005
3, -5.31157, 1.59343, 0.0415722
4, -6.92029, 1.61971, 0.0152936
5, -8.54563, 1.62937, 0.00562619
6, -10.1771, 1.63293, 0.00206976
7, -11.8108, 1.63424, 0.000761421
8, -13.4453, 1.63472, 0.000280111
9, -15.0801, 1.6349, 0.000103047
10, -16.715, 1.63496, 3.79089e-05
11, -18.35, 1.63499, 1.39459e-05
(t = 12 s) Jacques a atteint sa vitesse maximale (5.13042e-06 ms⁻²)
12, -19.985, 1.63499, 5.13042e-06
13, -21.62, 1.635, 1.88738e-06
14, -23.255, 1.635, 6.94326e-07
15, -24.89, 1.635, 2.55428e-07
16, -26.525, 1.635, 9.39669e-08
17, -28.16, 1.635, 3.45685e-08
18, -29.795, 1.635, 1.2717e-08
19, -31.43, 1.635, 4.67834e-09
20, -33.065, 1.635, 1.72106e-09
21, -34.7, 1.635, 6.33144e-10
22, -36.335, 1.635, 2.32921e-10
23, -37.97, 1.635, 8.56867e-11
24, -39.605, 1.635, 3.15224e-11
(t = 25 s) Jacques a atteint la profondeur désirée (-41.24 m)
25, -41.24, 1.635, 1.15964e-11
26, -42.875, 1.635, 4.26609e-12
(t = 27 s) Jacques s'arrête et gonfle son ballon (-44.51 m)
27, -44.51, 1.635, 1.56941e-12
28, -43.6078, -1.55028, -0.902224
29, -41.7256, -2.12059, -0.33191
30, -39.4829, -2.3304, -0.122103
31, -37.1076, -2.40758, -0.0449191
32, -34.6835, -2.43598, -0.0165248
33, -32.2414, -2.44642, -0.00607914
34, -29.7928, -2.45026, -0.00223639
35, -27.3417, -2.45168, -0.000822722
36, -24.8897, -2.4522, -0.000302663
37, -22.4374, -2.45239, -0.000111343
38, -19.985, -2.45246, -4.09609e-05
39, -17.5325, -2.45248, -1.50687e-05
40, -15.08, -2.45249, -5.54346e-06
41, -12.6275, -2.4525, -2.03932e-06
42, -10.175, -2.4525, -7.50225e-07
43, -7.7225, -2.4525, -2.75993e-07
44, -5.27, -2.4525, -1.01532e-07
45, -2.8175, -2.4525, -3.73515e-08
46, -0.365, -2.4525, -1.37409e-08
(t = 47 s) Jacques est de retour à la surface (2.0875 m)