

Avec l'avènement de l'informatique dans notre quotidien, le métier de programmeur s'est considérablement développé. Celui-ci doit répondre à un problème en créant des algorithmes interprétables par un ordinateur qui réalise ensuite la tâche souhaitée.

Quels sont les différents outils qui permettent à l'humain de communiquer avec l'ordinateur et de le programmer ?

Document 1. Différents systèmes, différentes interfaces, différents fichiers.



Nous utilisons l'informatique quotidiennement sur différents supports (ordinateurs, smart-phones, tablettes...) qui sont de plus en plus compatibles entre eux mais qui gardent leur spécificité et leur propre mode de fonctionnement. On dit qu'ils fonctionnent sous *différents systèmes d'exploitation*.

Document 2. Différents formats de fichier.



Le *langage naturel* permet d'exprimer une pensée ou un objectif compréhensible par un humain mais ce mode de communication n'est pas celui d'un ordinateur qui fonctionne dans un langage binaire (uniquement des 0 ou des 1).

Par souci d'efficacité pour les programmeurs, des *langages intermédiaires* plus ou moins proches du *langage machine* ont été mis au point. Ils permettent

d'utiliser des concepts plus facilement manipulables (des fonctions, des opérateurs, des types, etc.) rendant plus aisé l'écriture d'*algorithmes* sous forme de *code*. CE code est ensuite traduit en une série d'instructions correspondant aux opérateurs informatiques à effectuer pour la tâche demandée (manipulation directe de successions de bits, gestion des registres et des adresses mémoire,...). Ces instructions sont spécifiques à chaque type de machine et de processeur.

Document 3. Erreurs de programmation.

Les algorithmes sont sujets à des erreurs de syntaxe ou de logique menant à des dysfonctionnements (bugs). La construction du raisonnement occupe en effet une place particulière dans l'écriture d'un programme, en particulier lorsqu'on considère des disjonctions de cas. L'objectif est par exemple ci-dessous d'écrire une fonction permettant de savoir si une année n est bissextile, sachant qu'une année est bissextile si elle est multiple de 4 mais pas de 100 ou bien si elle est multiple de 400.

```
1  def bissextile(n):
2      if n%4==0:
3          if n%100!=0:
4              return "année bissextile"
5          else:
6              return "année non bissextile"
7  elif n%400==0:
8      return "année bissextile"
9  else:
10     return "année NON bissextile"
```

1. À l'aide des documents 1 et 2, identifier les fichiers exécutables. Pour les autres, indiquer à quel format de fichier correspond leur extension¹ (texte, image, son ou vidéo).
2. En utilisant le document 2, tester si 2020 est bien une année bissextile.
3. Identifier le bug et proposer une correction du programme du document 3.

1. Extension d'un fichier : elle apparaît dans le nom du fichier après un point et indique au système d'exploitation comment le traiter. Les fichiers contenant seulement des données (textes, images, sons, vidéos...) sont en général lisibles sur la plupart des systèmes d'exploitation.