

Contrôle continu Bash / Python

Durée : 01h00.

En cas de réalisation par un binôme, on précisera les noms et prénoms des membres du binôme au début de chaque fichier.

L'énoncé présente le minimum attendu. La notation tiendra compte de l'effort fait pour écrire les programmes suivant les bonnes pratiques de développement (commentaires appropriés, découpage en fonctions, ...) et pour ajouter des fonctionnalités intéressantes, y compris pour la gestion des cas d'erreurs.

On veut réaliser un histogramme de la filmographie d'un acteur ou d'une actrice comme celui de la figure 1.

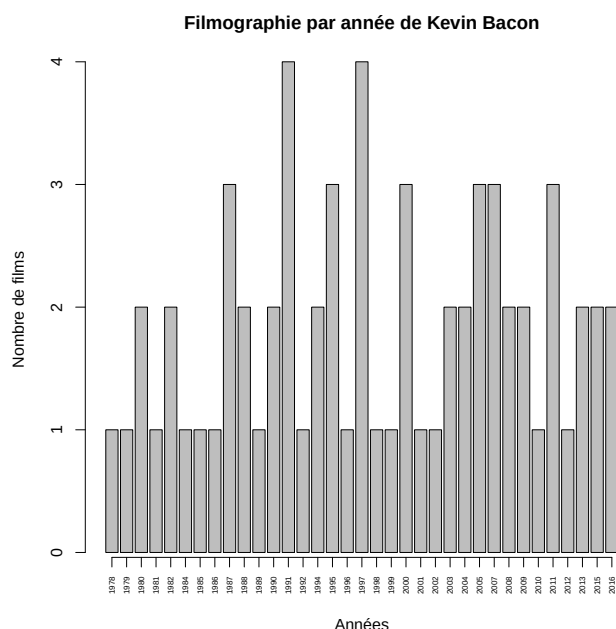


FIGURE 1 – Histogramme à produire pour Kevin Bacon.

On dispose pour cela dans le fichier `~goulard-f/X1BI030/dbmovies.json` d'une base de données de films au format JSON. Chaque entrée de la base a le format décrit dans la table 1.

1 Le programme Python casting.py

On va commencer par réaliser un programme Python prenant en entrée le chemin d'accès à la base de films au format JSON et un nom d'acteur ou d'actrice. Le programme doit retourner sur la sortie standard la liste des années des films dans lesquels l'acteur ou l'actrice joue. Exemple :

```
# ./casting.py ~goulard-f/X1BI030/dbmovies.json "Kevin BACON"
1991 1978 1995 1980 1979 1981 1998 1990 2003 1988 1990 2003 1996 1984 2000
```

TABLE 1 – Schéma d’une entrée JSON pour un film.

```
{
  "title": "Blade Runner",
  "cast": [
    "Harrison Ford",
    "Rutger Hauer",
    "Sean Young",
    "Edward James Olmos",
    "M. Emmet Walsh",
    "Daryl Hannah",
    "William Sanderson",
    "Brion James",
    "Joe Turkel",
    "Joanna Cassidy",
    "James Hong",
    "Morgan Paul",
    "Hy Pyke"
  ],
  "directors": ["Ridley Scott"],
  "producers": ["Michael Deeley"],
  "companies": [
    "The Ladd Company",
    "Shaw Brothers",
    "Blade Runner Partnership",
    "Warner Bros."
  ],
  "year": 1982
}
```

```
1982 2000 1987 1995 1995 1988 1999 2004 2005 2001 1987 1991 1989 2005 1994
1991 2002 2007 1994 1997 1997 2005 1997 1986 2007 2007 1980 2008 1991 2009
2000 1992 1997 1982 1987 2010 2011 2011 2011 2004 2012 2013 2009 2015 2013
2016 2015 2016 2008 1985
```

Le programme doit afficher un message d’aide sur la sortie d’erreur standard et retourner le code d’erreur 1 s’il est appelé avec le mauvais nombre de paramètres. Il doit aussi afficher un message d’erreur et retourner le code d’erreur 2 si le fichier JSON ne peut être lu. Le programme doit retourner le code d’erreur 3 si aucun film n’est trouvé.

Le script BASH casting.sh

Le programme casting.sh sert d’interface avec l’utilisateur. Il accepte le chemin d’accès à la base de films au format JSON ainsi que le nom d’un acteur ou d’une actrice en paramètres sur la ligne de commande. En retour, il crée un fichier PDF contenant l’histogramme pour l’acteur ou l’actrice en question. Exemple :

```
# ./casting.sh ~goulard-f/X1BI030/dbmovies.json "Kevin BACON"
# Création du fichier 'Kevin Bacon.pdf'
```

Le programme doit afficher un message d’aide sur la sortie standard et retourner le code d’erreur 1 si le nombre de paramètres est incorrect ; il doit afficher un message d’erreur et retourner le code d’erreur 2 si le fichier de films n’est pas accessible ; il doit afficher un message d’erreur et retourner le code d’erreur 3 si le programme casting.py ou le programme hist.R ne sont pas dans le répertoire courant. Il doit afficher un message d’erreur et retourner le code d’erreur 4 si le fichier PDF à créer existe déjà.

Le programme hist.R, [disponible sur madoc](#), accepte sur son entrée standard la liste des années des films où apparait l’acteur ou l’actrice dont le nom est donné en paramètre de la ligne de commande. Il crée en retour un fichier PDF portant le nom de l’acteur ou de l’actrice.

Si aucun film n’a été trouvé, le programme ne doit créer aucun fichier PDF et afficher un message. Exemple :

```
# ./casting.sh ~goulard-f/X1BI030/dbmovies.json "Jeannot Lapinou"
Aucun film trouvé pour Jeannot Lapinou.
```