

Université **IBM i**

19 et 20 novembre 2024

IBM Innovation Studio Paris

S09 – RPG Battle Dev 2024 : Finale et corrections

19 novembre 13:30 - 14:30

Nathanaël Bonnet / Florian Gradot

COMMON France

nathanael.bonnet@gaia.fr ; florian.gradot@gaia.fr



uui2024

#ibmi

#uui2024

IBM

common
FRANCE

Université IBM i

19 et 20 novembre 2024

IBM i
continuous innovation
continuous integration

IBM

common

F R A N C E

Clubs des utilisateurs IBM i & Power

Objectifs

Partager la compétence
Favoriser l'échange d'informations
Mutualiser la veille technologique
Regrouper les utilisateurs pour défendre nos intérêts

Plus ...

Avantages des membres

Participer gratuitement aux événements
Profiter des contenus exclusifs réservés à nos adhérents
Accès à la base de connaissance Common
Tarif sur le Congrès Annuel de Common Europe

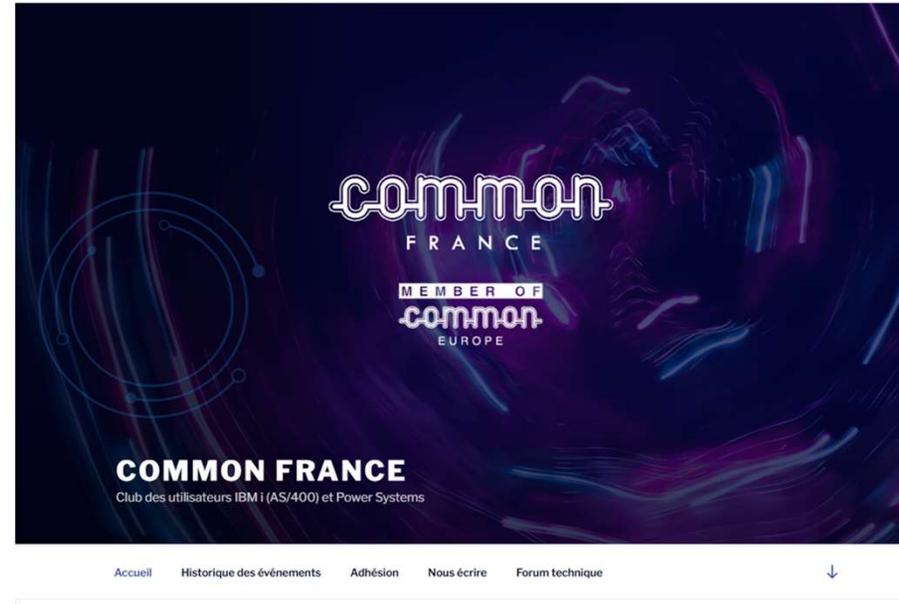
Plus ...

Adhérer ?

- Plot Common France dans le cœur de verre
- www.commonfrance.fr

Formules	
Prestation	Montant
Adhésion entreprise	450,00 €
Adhésion individuelle	150,00 €
Moins de 30 ans	Gratuit

- Le bureau



Université IBM i

19 et 20 novembre 2024



RPG Battle Dev 2024



PRENEZ VOTRE REVANCHE

La RPG Battle Dev 2024 en quelques chiffres

- 45 participations
 - 22 équipes
 - 23 participants individuels
 - Plus de 79 développeurs !
- De 4 pays
 - France
 - Belgique
 - Italie
 - Sénégal

400 Team INFITEX
Altinate
Armonie - Les Mixes du sud
Armonie - Equipe A
Armonie - Les Barbus sauf une
Armonie - Les pacificateurs
CGI
Chaussonours
CoCoDaDaNiNi
Equipe MBE
GCF-OCSI
GIEPS
GO GLS
Green Code Force
GuildeIBMi
Les Développeuses Chaussonnées
Les invincibles
Noobs & I
Pépinère14
RMM
AGIPI
Rocket By Hardis

BENERGY
CFD
Clément GIRARD
CLU
Colibri
CSD68
DSR
EXICIEL
Georges Kopp
GFP
JoeDiha
Le LAC du 400
LP consultant
Marion
Maxou
MK1
Romain DULOQUIN
Torbel
tricoptops
vzm

MERCI!

Et merci également...



Philippe
Bourgeois



Jérôme
Clément

Université IBM i

19 et 20 novembre 2024

IBM i
continuous innovation
continuous integration

IBM

Restitution des corrections



PRENEZ VOTRE REVANCHE

Nos critères de correction

- Programme rendu dans les temps ?
 - Bon résultat ?
 - Code simple et moderne ?
 - Code lisible ?
 - Code optimisé ?
-
- Nous étions 3 à nous partager les corrections de vos 315 programmes ... avec le plus d'objectivité possible ... mais aussi le souci du détail pour vous départager

Manche 1 – 6 Novembre – Epreuve E1 – Affichage des films

- Enoncé
 - Afficher pour chaque catégorie le film ayant le plus de vues

```
// Objectif :  
// Afficher pour chaque catégorie le film ayant le plus de vues et les afficher (DSPLY)  
// par ordre alphabétique.  
// Mise en forme du résultat :  
// DSPLY Comédie  
// DSPLY Les bronzes  
// DSPLY 0094558100  
// DSPLY Fantastique  
// DSPLY Avatar  
// DSPLY 0985452365
```

Formatage sur 10 positions !

Manche 1 – 6 Novembre – Epreuve E1 – Affichage des films

- Enoncé
 - Afficher pour chaque catégorie le film ayant le plus de vues
- Proposition de correction

```
ctl-opt actgrp(*new) alwnull(*no);
ctl-opt extbinint(*yes) ;

dcl-f film rename(film:ffilm);

// Pour lecture et stockage des données
dcl-ds curFilm likerec(ffilm) inz ;
dcl-ds films likerec(ffilm) dim(*auto:10) inz ;
```

- Résultat attendu

```
DSPLY Catastrophe
DSPLY Titanic
DSPLY 0085452365
DSPLY Comedie
DSPLY Les bronzes
DSPLY 0094558100
DSPLY Dessin Anime
DSPLY Shrek
DSPLY 0998545236
DSPLY Fantastique
DSPLY Avatar
DSPLY 0985452365
```

Manche 1 – 6 Novembre – Epreuve E1 – Affichage des films

- Proposition de correction (suite)

```
// Lecture du fichier des films pour chargement du tableau
read film curFilm;
dow not %eof;
    addFilm(curFilm);
    read film curFilm;
Enddo;

// Affichage du résultat
displayResult();

return ;
```

Manche 1 – 6 Novembre – Epreuve E1 – Affichage des films

- Proposition de correction (fin)

```
// procédure de chargement du tableau
dcl-proc addFilm;
  dcl-s l_index uns(3) inz(0);

  dcl-pi *n;
  | p_film likerec(ffilm) const ;
  End-pi;

  l_index = %lookup(p_film.categorie:films(*).categorie);

  select;
  | // si la catégorie est dans la liste et que le film a plus de vues
  | When l_index > 0 and films(l_index).vues < p_film.vues;
  | | films(l_index) = p_film ;
  | // si la catégorie n'existe pas on la rajoute
  | When l_index = 0;
  | | films(*next) = p_film;
  Ends;
End-proc;
```

```
// Procédure d'affichage
dcl-proc displayResult;

  // trier par catégorie
  sorta films(*).categorie;

  // Mise en forme et affichage
  for-each curFilm in films;
  | DSPLY curFilm.categorie;
  | DSPLY curFilm.libelle;
  | DSPLY %editc(curFilm.vues:'X') ;
  endfor;
End-proc;
```

Manche 1 – 6 Novembre – Epreuve E2 – Affichage des logs

- Enoncé
 - Afficher les logs du fichier du plus récent au plus ancien
- Proposition de correction

```
ctl-opt actgrp(*new) alwnull(*no);

dcl-f log rename(log:flog);

// Lecture et stockage
Dcl-Ds log_cur likerec(flog);
Dcl-Ds logs likerec(flog) dim(*AUTO:100);

// Pour formatage date et heure
dcl-pr formatage varchar(20) overload(format_date : format_time);
dcl-s msg varchar(52) inz ;
```

- Résultat attendu

```
DSPLY 11 Novembre 2024 - 17h12mn00s - Avertissement
DSPLY 10 Novembre 2024 - 11h12mn12s - Suspect
DSPLY 09 Novembre 2024 - 17h12mn00s - Avertissement
DSPLY 08 Novembre 2024 - 00h01mn01s - Critique
DSPLY 01 Novembre 2024 - 23h52mn10s - Critique
DSPLY 31 Octobre 2024 - 09h45mn25s - Suspect
DSPLY 07 Octobre 2024 - 20h15mn01s - Avertissement
```

Manche 1 – 6 Novembre – Epreuve E2 – Affichage des logs

- Proposition de correction (suite)

```
// Lecture du fichier des logs
read log log_cur;
dow not %eof;
  // Exclusion des logs <> de la liste
  if log_cur.logtype in %list('Critique' : 'Avertissement' : 'Suspect');
  | // Stockage
  | logs(*next) = log_cur ;
  endif;
  read log log_cur;
Enddo;

// Tri
sorta(d) logs %fields(logdate : logtime ) ;
// Affichage
for-each log_cur in logs;
  msg = formatage(log_cur.logdate) + ' - ' +
  | formatage(log_cur.logtime) + ' - ' +
  | log_cur.logtype ;
  dsply msg;
endfor ;

return ;
```

Manche 1 – 6 Novembre – Epreuve E2 – Affichage des logs

- Proposition de correction (suite)

```
// Formatage de la date au format 11 Novembre 2024
dcl-proc format_date;
  dcl-s l_datechar varchar(10);
  dcl-s l_mois varchar(15) dim(12) inz static ;
  dcl-s l_inz ind inz(*off) static ;

  dcl-pi *n varchar(20);
  | p_date date const;
end-pi;

// chargement des libellés, au premier appel
if not l_inz ;
  l_mois = %list('Janvier' : 'Fevrier' : 'Mars'      : 'Avril'   : 'Mai'       : 'Juin' :
                'Juillet' : 'Aout'      : 'Septembre' : 'Octobre' : 'Novembre' : 'Decembre');
  l_inz = *on ;
endif ;

l_datechar = %char(p_date:*iso);
// Retourner la date sous le format demandé
return %subst(l_datechar:9:2) + ' ' + l_mois(%subdt(p_date:*months)) + ' ' +
       | %subst(l_datechar:1:4);

End-proc;
```

Manche 1 – 6 Novembre – Epreuve E2 – Affichage des logs

- Proposition de correction (fin)

```
// Formatage de l'heure au format 14h30mn45s
dcl-proc format_time;

  dcl-pi *n varchar(20);
  | p_time time const ;
end-pi;
return %editw(%dec(p_time) : '0 h mn s') ;

End-proc;
```

Manche 2 – 13 Novembre – Epreuve E3 – Calcul clé IBAN

- Enoncé
 - Calculer les clés des IBAN fournis

```
// Consignes :  
// 1. Utiliser uniquement des instructions RPG natives (pas de SQL, pas d'appels d'API ...)  
// 2. Les déclarations ne doivent pas être modifiées  
// 3. Les lignes déjà présentes dans le code ne doivent pas être supprimées  
  
// Règles de calcul :  
// Cf https://fr.wikipedia.org/wiki/International\_Bank\_Account\_Number  
  
// Exemple :  
// FR{00}4255910000080016689317  
// doit afficher : DSPLY FR7642
```

Algorithme de vérification de l'IBAN [\[modifier \]](#) [\[modifier le code \]](#)

1. Enlever les caractères indésirables (espaces, tirets),
2. Déplacer les 4 premiers caractères à la fin du compte,
3. Remplacer les lettres par des chiffres au moyen d'une table de conversion (A=10, B=11, C=12 etc.),
4. Diviser le nombre ainsi obtenu par 97,
5. Si le reste n'est pas égal à 1 l'IBAN est incorrect : modulo de 97 égal à 1.

Manche 2 – 13 Novembre – Epreuve E3 – Calcul clé IBAN

- Proposition de correction

```
ctl-opt actgrp(*new) alwnull(*no);

dcl-f iban rename(iban:fiban);

// Parcours des IBAN et calcul des clés
read iban ;
dow not %eof ;
  if %Left(valeur:2) IN %list('FR':'GR':'IT':'MC') ;
    dsply calculer_cle(valeur);
  endif ;
  read iban ;
enddo ;

return ;
```

- Résultat attendu

```
DSPLY  FR7630001007941234567890185
DSPLY  IT4311808009101234567890147
DSPLY  GR6742559000011234567890121
DSPLY  MC5810011000201234567890188
DSPLY  FR7630056009271234567890182
```

Manche 2 – 13 Novembre – Epreuve E3 – Calcul clé IBAN

- Proposition de correction (suite)

```
// Calculer la clé
dcl-proc calculer_cle ;
  dcl-pi *n char(27) ;
    p_iban char(27) value ;
  end-pi ;

  // Clé
  dcl-s l_cle      packed(2:0)  inz ;
  // Parcours de l'IBAN
  dcl-s l_pos      uns(3)  inz ;
  // Pour constitution IBAN numérique
  dcl-s l_iban     like(p_iban) inz ;
  dcl-s l_iban_char varchar(54) inz ;
  // Lettres à remplacer
  dcl-s LETTRES char(26) const inz('ABCDEFGHIJKLMNOPQRSTUVWXYZ');
```


Manche 2 – 13 Novembre – Epreuve E4 – Gestion des stocks

- Enoncé
 - Afficher le statut de chaque ligne de stock de la zone A
- Proposition de correction

- Résultat attendu

```
DSPLY POWER10 STOCK CRITIQUE!  
DSPLY ECRAN 17" A REAPPROVISIONNER  
DSPLY CLAVIER STOCK OK
```

```
// Stockage des éléments XML lus  
dcl-ds stock dim(*auto :10) qualified ;  
  zone          varchar(1);  
  dcl-ds article ;  
    qte         packed(5:0) ;  
    libelle     varchar(20);  
  end-ds ;  
end-ds ;  
// Pour parcours  
dcl-ds curStock likeds(stock) ;
```

```
<entrepot>  
  <Header>  
    <Description>Interface stock entrepot 1</Description>  
  </Header>  
  <stock>  
    <zone>A</zone>  
    <article qte="0">POWER10</article>  
    <description>Le meilleur système</description>  
  </stock>  
  <stock>  
    <zone>B</zone>  
    <article qte="10">DRIVE LTO 7</article>  
    <description>Lecteur externe</description>  
  </stock>
```

Manche 2 – 13 Novembre – Epreuve E4 – Gestion des stocks

- Proposition de correction (suite)

```
// initialisation
clear stock;

// lecture données
read(e) interface;
if %error or %eof(interface) ;
    dsply 'Problème d'accès au données' ;
    return ;
endif ;
```

```
// Extraction des infos :
xml-into stock %xml(ligne:'case=any path=entrepot/stock trim=all ' +
    'allowextra=yes datasubf=libelle' );
// On parcourt, on affiche. Pas de contrainte de tri exprimée
for-each curStock in stock ;
    if curStock.zone = 'A' ;
        select curStock.article.qte ;
            when-is 0 ;
                dsply (curStock.article.libelle + ' STOCK CRITIQUE!' ) ;
            when-in %range(1:4) ;
                dsply (curStock.article.libelle + ' A REAPPROVISIONNER' ) ;
            other ;
                dsply (curStock.article.libelle + ' STOCK OK' ) ;
        endsel ;
    endif ;
endfor ;

return ;
```

Manche 3 – 19 Novembre – Epreuve E5 – Erreur d’affichage

- Enoncé
 - Afficher RPG BATTLE DEV 2024!
- Proposition de correction

- Résultat attendu

DSPLY RPG BATTLE DEV 2024!

```
ctl-opt actgrp(*new) alwnull(*no);

dcl-s contenu varchar(100) inz('IBM:COMMON FRANCE::2024::rpg battle dev 2024!:Edition:2');

dsply procsplit(contenu);

return ;

dcl-proc procSplit;
  dcl-s resSplit varchar(20) dim(10);
  dcl-c ind const(6);

  dcl-pi *n varchar(20);
  | p_entry varchar(100);
  End-pi;

  resSplit = %upper(%split(contenu : ':' : *allsep));
  return resSplit(ind);
End-proc;
```

Manche 3 – 19 Novembre – Epreuve E6 – Combien de Battle?

- Enoncé
 - Afficher le résultat attendu
- Proposition de correction

- Résultat attendu

```
DSPLY Et 1
DSPLY Et 2
DSPLY BATTLE DEV!
```

```
ctl-opt actgrp(*new) alwnull(*no);

dcl-s cpt packed(1:0) inz(1);

dow not compteurTest and cpt < 6;
  dsply ('Et ' + %char(cpt));
  cpt += 1;
enddo;
dsply 'BATTLE DEV!';

return ;

dcl-proc compteurTest;
  dcl-s compteur packed(1:0) static;

  dcl-pi *n ind;
  End-pi;

  compteur += 1;
  return compteur = 3;

End-proc;
```

Manche 3 – 19 Novembre – Epreuve E7 – Liste des épreuves

- Énoncé
 - Modifier le programme pour afficher le résultat attendu
- Proposition de correction

```
ctl-opt actgrp(*new) alwnull(*INPUTONLY);  
  
dcl-f epreuve rename(epreuve:fepreuve);  
  
read epreuve;  
dow not %eof;  
  dsply Nom;  
  read epreuve;  
enddo;  
  
return ;
```

- Résultat attendu

```
DSPLY  E1  
DSPLY  E2  
DSPLY  E3  
DSPLY  E4  
DSPLY  E5  
DSPLY  E6  
DSPLY  E7
```

MEERCS