|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| M:\ADMIN\henallux_montgolfiere.png | **UE 122 – Langage de programmation : Bases**  Informatique, bloc 1 — Interro formative — novembre 2023 | | | | |
| *Nom et prénom en MAJUSCULES* | | | | / 20 |
| **Michaux Dorian** | | | |
| *Groupe* | **H** | *Enseignant* |  |

**Consignes**

* Les cadres bleutés sont les cadres que vous devez compléter (en ajoutant des informations pertinentes, comme votre nom, ou en y plaçant vos réponses).
* Quand on vous demande un diagramme d’actions, copiez-collez dans le cadre bleuté correspondant la partie de droite (code formatté) de l’éditeur de diagrammes d’actions.

# Question 1

Quel affichage sera produit par le (pseudo-)code suivant ?

A computer screen shot of a number

Description automatically generated

|  |
| --- |
| Le pseudo-code affichera 5,6,4,9,10  mais n’affichera pas 7,11 et 12. |

Commentaires/correction :

|  |
| --- |
|  |

# Question 2

Le (pseudo-)code suivant est censé implémenter un module

A black and white text

Description automatically generated

interprétant une cote sur 20. On suppose que la cote reçue est un entier, et l’interprétation attendue en résultat est une chaîne de caractères correspondant à la grille suivante.

|  |  |
| --- | --- |
| **Cote reçue** | **Interprétation** |
| 0, 1, 2, …, 9 | Échec |
| 10, 11, 12, …, 17 | Réussite |
| 18, 19, 20 | Grande réussite |
| (autre valeur) | Cote invalide |

Barrez (en sélectionnant puis en cliquant sur ) toutes les parties à supprimer dans ce diagramme d’actions (soit parce qu’elles sont inutiles soit parce qu’elles ne devraient pas du tout apparaître).

┌─── \*

│ ~~Obtenir cote~~

│ ~~résultat = ""~~

│┌── if (cote < 0 OR cote > 20)

││ résultat = "Cote invalide"

│├── else

││┌── if (~~cote ≥ 0 AND~~ cote ≤ 9)

│││ résultat = "Échec"

││├── else

│││┌── if (~~cote ≥ 10~~ ~~AND~~ cote < 18)

││││ résultat = "Réussite"

│││├── else

││││┌─~~─ if (cote ≥ 18 AND cote ≤ 20)~~

│││││ résultat = "Grande réussite"

││││└~~──~~

│││└──

││└──

│└──

│ Sortir résultat

└──

Commentaires/correction :

|  |
| --- |
|  |

# Question 3

On désire écrire un diagramme d’actions pour un programme qui reçoit un nombre impair entre 5 et 25 (5 et 25 étant eux-mêmes des valeurs valides) et produit un affichage en forme de triangle, comme le suivant (pour le nombre 7).

\*

*1 étoile, puis 3, puis 5 puis 7 (le nombre entré)*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*

*7 étoiles, puis 6, 5, 4…  
jusqu’à ne plus en avoir qu’une*

\*\*\*\*

\*\*\*

\*\*

\*

a) Complétez le diagramme d’actions suivant, pour un module qui reçoit un nombre et indique s’il est valide (c’est-à-dire s’il s’agit bien d’un nombre impair entre 5 et 25 – comme indiqué au début de l’énoncé).

A black text and black text

Description automatically generated with medium confidence

Que faut-il mettre à la place des … ?

|  |
| --- |
| Valide = nombre >= 5 AND nombre <= 25 AND nombre % 2 == 1 |

Commentaires/correction :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRÈS INSUFFISANT | INSUFFISANT | JUSTE SUFFISANT | SATISFAISANT | BIEN ACQUIS |
|  | | | | |

b) Sachant qu’on dispose déjà du module

A black rectangle with black text

Description automatically generated

permettant d’afficher une ligne constituée d’un nombre donné d’étoiles, écrire le programme demandé.

Le programme doit effectuer les tâches suivantes :

- demander à l’utilisateur un nombre valide (et répéter la question jusqu’à ce que l’utilisateur entre un nombre valide) ;

- produire l’affichage triangulaire décrit plus haut.

Votre programme devra utiliser les modules estValide et afficheLigneÉtoiles.

|  |
| --- |
| ┌─── \*  │ obtenir nombre  │ o───────────o ↓ nombre  │ │ estValide │  │ o───────────o ↓ valide  │  │╔══ while (!valide)  │║ obtenir nombre  │║ o───────────o ↓ nombre  │║ │ estValide │  │║ o───────────o ↓ valide  │╙──  │  │ compteurCroissant = 1  │╔══ while (compteurCroissant ≤ nombre)  │║ o─────────────────────o ↓ compteurCroissant  │║ │ afficheLigneEtoiles │  │║ o─────────────────────o  │║ compteurCroissant = compteurBoucle + 2;  │╙──  │  │ compteurDecroissant = nombre - 1  │╔══ while (compteurDecroissant ≥ 1)  │║ o─────────────────────o ↓ compteurDecroissant  │║ │ afficheLigneEtoiles │  │║ o─────────────────────o  │║ compteurDecroissant--  │╙──  │  └────────── |

Commentaires/correction :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRÈS INSUFFISANT | INSUFFISANT | JUSTE SUFFISANT | SATISFAISANT | BIEN ACQUIS |
|  | | | | |

# Question 4

À l’approche de la fin de l’année 2023, un streamer décide d’encoder tous les jeux qu’il a testés en stream depuis janvier. Pour chaque jeu, il entre son nom (chaîne de caractères vide pour signifier la fin de l’encodage), le nombre d’heures qu’il y a consacrées et son appréciation du jeu (un code entier entre 1 et 3).

Il compte partager avec ses followers un classement où chacun de ces jeux se verrait attribuer une note calculée comme suit :

- La note correspond à un nombre de points dépendant du nombre d’heures de stream. Les 8 premières heures de stream comptent pour 2 points chacune. Les heures de stream à partir de la 9e ajoutent chacune 3 points.

- La note est ensuite modifiée en fonction de l’appréciation du streamer :

|  |  |  |
| --- | --- | --- |
| **Appréciation** | **Signification** | **Modification de la note** |
| 1 | Mauvais | - 10% |
| 2 | Moyen | Aucune modification |
| 3 | Bon | + 10% |

a) Écrivez un module permettant de calculer la note d’un jeu à partir du nombre d’heures de stream et de l’appréciation du streamer. N’oubliez pas de donner sa signature (la « boîte » avec son nom, les paramètres d’entrée et les paramètres de sortie). Utilisez ce module pour la question suivante.

|  |
| --- |
| o────────────────o ↓ nbHeures,appreciation  │ CalculeNoteJeu │  o────────────────o ↓ noteJeu  ┌─── \*  │ noteJeu = 0  │┌── if (nbHeures ≤ 8)  ││ noteJeu += 2 \* nbHeures  │├── else  ││ noteJeu += (8 \* 2) + ((nbHeures - 8) \* 3)  │└──  │  │┌── if (appreciation == 1)  ││ noteJeu -= noteJeu \* 0.10  │├── else  ││┌── if (appreciation == 3)  │││ noteJeu \*= 1.10  ││└──  │└──  │  │ sortir noteJeu  └────────── |

Commentaires/correction :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRÈS INSUFFISANT | INSUFFISANT | JUSTE SUFFISANT | SATISFAISANT | BIEN ACQUIS |
|  | | | | |

b) Écrivez un programme qui permettra au streamer d’encoder les données et produira les sorties suivantes :

- pour chaque jeu : son nom et sa note finale ;

- le nom du jeu possédant la meilleure note finale ;

- le pourcentage d’heures de stream consacrées à des jeux considérés comme « mauvais » (code d’appréciation égal à 1)

- la moyenne des notes des jeux que le streamer a considérés comme « bons » (code d’appréciation égal à 3).

|  |
| --- |
| ┌─── \*  │ nbJeuBon = 0  │ notesJeuBon = 0  │ nbHeuresTotal = 0  │ nbHeuresJeuMauvais = 0  │ meilleurNoteJeu = LV  │ nomMeilleurJeu = ""  │ obtenir nomJeu  │  │╔══ while (nomJeu ≠ "")  │║ obtenir nbHeures  │║ obtenir appreciation  │║  │║ nbHeuresTotal += nbHeures  │║  │║ o────────────────o ↓ nbHeures,appreciation  │║ │ CalculeNoteJeu │  │║ o────────────────o ↓ noteJeu  │║  │║┌── if (noteJeu > meilleurNoteJeu)  │║│ meilleurNoteJeu = noteJeu  │║│ nomMeilleurJeu = nomJeu  │║└──  │║  │║┌── if(appreciation == 1)  │║│ nbHeuresJeuMauvais += nbHeures  │║└──  │║  │║┌── if(appreciation == 3)  │║│ notesJeuBon += noteJeu  │║│ nbJeuBon++  │║└──  │║  │║ sortir nomJeu  │║ sortir noteJeu  │║  │║ obtenir nomJeu  │╙──  │  │ moyenneNotesJeuBon = noteJeuBon / nbJeuBon  │ pourcentageHeuresJeuMauvais = (nbHeuresJeuMauvais \* 100) / nbHeuresJeuTotal  │  │ sortir moyenneNotesJeuBon  │ sortir pourcentageHeuresJeuMauvais  │ sortir nomMeilleurJeu  └────────── |

Commentaires/correction :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRÈS INSUFFISANT | INSUFFISANT | JUSTE SUFFISANT | SATISFAISANT | BIEN ACQUIS |
|  | | | | |