

Responsable : J. Mitéran - Durée : 2 h - Documents autorisés : cours et TD/TP. Pas d'ordinateur ni de téléphone ni de calculatrice.

Commentez vos programmes – rédigez vos réponses en les justifiant.

Question A : (12 points)(C/C++)

On veut réaliser un système d'intelligence artificielle (IA) simplifié. Ce système consistera à décider automatiquement de la mention attribuée à des étudiants à partir de leurs notes de programmation et d'anglais. Pour déterminer la règle d'attribution d'une mention à un étudiant, on utilisera des données provenant d'attribution de mentions à des étudiants des années antérieures (phase d'apprentissage).

Les notes de 500 étudiants ont été regroupées dans deux tableaux de la manière suivante. Le tableau **TabProg** contient les notes de programmation de 500 étudiants. Le tableau **TabAnglais** contient les notes d'anglais des 500 mêmes étudiants (dans le même ordre).

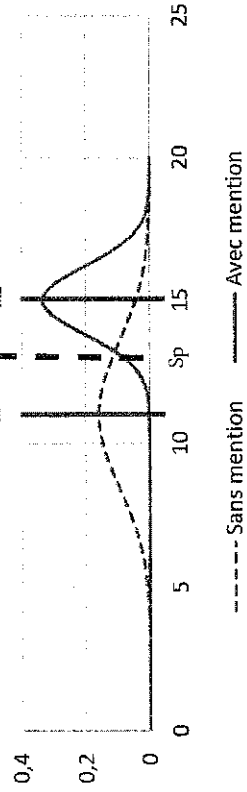
Le tableau **TabMention** contient 0 pour les étudiants avec mention et 1 pour les étudiants sans mention.

Voici un extrait du contenu de ces tableaux. Le premier étudiant a donc obtenu 12 en programmation, 15 en anglais, et une mention. Le second a obtenu 10 et 10, et pas de mention, et ainsi de suite. Ces tableaux sont supposés déjà déclarés et initialisés.

TabProg	12	10	11	8	17
TabAnglais	15	10	9	15	10
TabMention	1	0	0	0	1

Vous ne savez pas comment cette mention a été attribuée, mais vous allez déterminer automatiquement une règle mathématique qui permettra de reproduire automatiquement cette attribution de mention pour les futurs étudiants. Vous allez calculer les moyennes des deux distributions, puis la moyenne de ces deux moyennes (matérialisée par une ligne verticale pointillée sur la figure ci-dessous). Ceci donnera un seul Sp pour la note de programmation et Sa pour la note d'anglais. La règle d'attribution de la mention sera alors la suivante : si un étudiant a obtenu P en programmation et A en anglais, il aura une mention si (P>Sp et A>Sa).

Répartition des notes de programmation



1 - Ecrivez une fonction Moyenne qui reçoit en paramètre les pointeurs vers 2 tableaux : note et mention, ainsi que la taille de ces tableaux, et enfin un booléen qui, s'il vaut 0, permettra de calculer la moyenne des notes des étudiants sans mention, et s'il vaut 1, de calculer la moyenne des notes des étudiants avec mention.

Exemple d'appel pour calculer la moyenne des notes de programmation des étudiants sans mention :

```
M1=Moyenne(TabProg, TabMention, 500, 0);
```

2 - Ecrivez la fonction qui calcule les moyennes M1 et M2 pour les notes de programmation, ainsi que M3 et M4 pour les notes d'anglais. Déduisez-en les seuls Sp et Sa.

3 - Ecrivez la fonction qui reçoit en paramètre les notes de programmation et d'anglais d'un étudiant, ainsi que les seuls Sp et Sa, et retourne 1 si l'étudiant obtient une mention, et 0 sinon. Donnez un exemple d'appel.

4 - Ecrivez une fonction qui calcule le nombre d'erreurs commises par un tel système sur l'ensemble des étudiants des tableaux précédents (une erreur est une attribution de mention à un étudiant qui n'en avait pas, ou une non attribution à un étudiant qui aurait dû en avoir).

5 - Comment les données (notes, mention) auraient pu être mieux organisées que dans 3 tableaux séparés ? Proposez une variante de la fonction du 1 utilisant les structures.

Question B : (2 points)

2- Soit le code suivant extrait du main (on suppose que les includes sont intégrés correctement et qu'Unicode est désactivé).

Quel est le résultat de l'affichage ?

```
char Tab[10];
Tab[0]=7;
Tab[1]=1;
Tab[2]=5;
*(Tab+3)=6;
*(Tab+3)=*(Tab)+*(Tab+3));
Tab[9]=0;
int j=3;
char k=0;
k=2+*(Tab+);
printf ("k=%d",k);
```

Question C : (6 points)

On veut définir une classe CComplex permettant de manipuler les nombres complexes de type $C=a+ib$ ou a est la partie réelle et b la partie imaginaire. *Aucun affichage écran ou saisie clavier ne doit être fait dans les fonctions de cette classe.* On suppose que la classe complex du C++ n'est pas disponible.

- 1- En variables membres (attributs), la classe CComplex comportera les variables a et b . Donnez un constructeur avec initialisation des variables membres.
- 2- Ajouter à la classe CComplex une fonction membre (méthode) « **Produit** » qui permet de calculer et retourner le produit de deux nombres complexes reçus en paramètres. Cette fonction n'affecte pas les variables membre a et b .
- 4- Ajouter à la classe CComplex une fonction « **Angle** » qui permet de calculer et retourner l'argument du nombre complexe concerné (partie réelle strictement positive). Le calcul est donc fait à partir des variables membres a et b . Rappel : l'argument d'un complexe est l'angle entre l'axe des réels et le vecteur de coordonnées (a,b) .
- 5- Donnez le fichier header complet (.h) de la classe CComplex. Ecrire un programme principal (main), dans lequel vous déclarerez deux objets de type CComplex, en les initialisant avec les valeurs de votre choix. Vous calculerez le produit et l'argument de ces nombres, et vous afficherez les résultats à l'aide de printf ou cout.