



Programmation en C (EPINF2E1)

Nom et prénom :

.....

- ▶ *Durée : 2 heures.*
- ▶ *Aucun document n'est autorisé. L'usage de la calculatrice ou de tout équipement électronique est interdit.*
- ▶ *Le sujet qui vous a été attribué est unique et il ne vous sera fourni que cet exemplaire.*
- ▶ *Cochez ou noircissez l'intérieur des cases choisies. Vous pouvez utiliser un correcteur liquide pour faire disparaître une case noircie par erreur (**mais ne redessinez pas la bordure de la case**).*
- ▶ *Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse. Il n'y a pas de points négatifs entre les questions.*

Codez votre numéro d'étudiant ci-dessous :

<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0
<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1
<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2
<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3
<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4
<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5
<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6
<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7
<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8
<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9

1 Questionnaire à choix multiples (12 points)

2 Questions ouvertes (8 points)

Nous souhaitons manipuler des notes d'étudiants. On utilisera pour cela les définitions suivantes :

```
#define NBMAXETU 500
#define LGMAX 127
struct etudiant {
    int num_etu; /* numéro de l'étudiant */
    char nom[LGMAX+1];
    char prenom[LGMAX+1];
};
struct note {
    int num_etu; /* numéro de l'étudiant */
    float valeur; /* note sur 20 */
};
```

Le nom et le prénom d'un étudiant étant des chaînes de caractères, vous pourrez vous aider de la fonction : `int strcmp(const char *ch1, const char *ch2)` qui retourne 0 si les deux chaînes sont identiques ou un entier non nul si elles sont différentes.

On suppose par ailleurs les deux fonctions suivantes déjà écrites :

- `void lire_classe(char *nom_fich, struct etudiant etudiants[], int *p_nb_etu)` qui lit les informations sur les étudiants stockées dans le fichier dont le nom est passé en premier paramètre, les range dans le tableau `etudiants` et délivre en sortie leur nombre grâce au paramètre `p_nb_etu` ;
- `void lire_notes(char *nom_fich, struct note notes[], int *p_nb_notes)` qui effectue la même opération mais pour les notes obtenues à un examen.

On ne sait pas dans quel ordre sont stockées les informations sur les étudiants. Il en est de même pour les notes. Enfin, pour la question 28, nous supposons que la fonction principale commence par les déclarations suivantes :

```
int main(void) {
    struct etudiant etudiants[NBMAXETU];
    struct note notes[NBMAXETU];
    int nb_etu, nb_notes, numero_etudiant;
    float min, max, moy;
    char nom[LGMAX+1], prenom[LGMAX+1];
```



Q.1) ♣ Écrivez la fonction d'en-tête : `float note(int numero_etudiant, struct note notes[], int nb_notes)` qui retourne la note de l'étudiant de numéro `numero_etudiant`. Si l'étudiant n'est pas présent dans le tableau `notes`, alors cette fonction doit retourner -1.

☐ 0 ☐ 1 ☐ 2 ☐ 0.25 ☐ 0.5 *Réservé au correcteur*

Q.2) ♣ Écrivez la fonction d'en-tête : `int numero(char *nom, char *prenom, struct etudiant etudiants[], int nb_etu, int *p_num_etu)` qui recherche dans la tableau `etudiants` l'étudiant dont le nom et le prénom sont passés en paramètres et délivre en sortie son numéro grâce au dernier paramètre. Cette fonction doit retourner 1 si l'étudiant a été trouvé ou 0 dans le cas contraire.

☐ 0 ☐ 1 ☐ 2 ☐ 0.25 ☐ 0.5 *Réservé au correcteur*



Q.3) ♣ Écrivez la fonction d'en-tête : `void statistiques(struct note notes[], int nb_notes, float *p_min, float *p_max, float *p_moy)` qui détermine la plus petite, la plus grande et la moyenne des notes stockées dans le tableau `notes` et les transmet à l'appelant grâce aux trois derniers paramètres.

☐ 0 ☐ 1 ☐ 2 ☐ 0.25 ☐ 0.5 *Réservé au correcteur*

Q.4) ♣ En utilisant toutes les fonctions précédentes, écrivez la suite de la fonction principale (dont le début se trouve dans l'énoncé) afin de : lire les informations sur les étudiants se trouvant dans le fichier `11.txt` ; lire dans le fichier `notes_progc.txt` les notes de ceux qui ont passé l'examen ; lire le nom et le prénom d'un étudiant tapés au clavier par l'utilisateur ; afficher à l'écran la note qu'il a obtenue (vous supposerez que cet étudiant existe et fait bien partie de ceux qui ont passé l'examen) ; afficher à l'écran les statistiques sur l'ensemble des notes.

☐ 0 ☐ 1 ☐ 2 ☐ 0.25 ☐ 0.5 *Réservé au correcteur*