
Éléments d'informatique : partiel de mi-semestre

Durée : 3 heures.

Documents autorisés : Aucun.

Recommandations : Un barème vous est donné à titre indicatif afin de vous permettre de gérer votre temps. La notation prendra en compte à la fois la syntaxe et la sémantique de vos programmes, c'est-à-dire qu'ils doivent compiler correctement. Une fois votre programme écrit, il est recommandé de le faire tourner à la main sur un exemple pour s'assurer de sa correction.

1 Étude de programmes et questions de cours (7 points)

1.1 Trace d'un programme (4.5 points)

```
1  /* Déclaration de fonctionnalités supplémentaires */
2  #include <stdlib.h> /* EXIT_SUCCESS */
3  #include <stdio.h> /* printf */
4
5  /* Déclaration constantes et types utilisateurs */
6  #define N 2
7
8  /* Déclaration de fonctions utilisateurs */
9
10 /* Fonction principale */
11 int main()
12 {
13     /* Déclaration et initialisation variables */
14     int t[N] = {3,0}; /* donnees */
15     int s; /* commentaire supprimé */
16     int i; /* var. de boucle */
17     int j; /* var. de boucle */
18     for(i = 0; i < N; i = i + 1) /* commentaire supprimé */
19     {
20         /* commentaire supprimé */
21         s = 0;
22         for(j = 1; j <= t[i]; j = j + 1) /* commentaire supprimé */
23         {
24             /* commentaire supprimé */
25             s = s + j;
26         }
27         /* commentaire supprimé */
28         printf("%d\n", s);
29     }
30     /* commentaire supprimé */
31     return EXIT_SUCCESS;
32 }
33
34 /* implantation de fonctions utilisateurs */
```

Question A. À quoi sert la ligne 6 et à quelle étape de la compilation sera t-elle traitée?

0.5 pt
4 min

Question B. Simulez l'exécution du programme, en réalisant sa **trace** : l'exécution du programme est représentée par un tableau à $n+2$ colonnes ; la première colonne étant le numéro de la ligne exécutée, les n colonnes suivantes, les colonnes des n variables du programme (n à déterminer) et la dernière colonne, la colonne servant à l'affichage à l'écran du programme. Seules les lignes modifiant l'état mémoire du programme sont à reporter dans le tableau.

3 pt
27 min

Question C. Que calcule et affiche ce programme ?

1 pt
9 min

1.2 Une erreur classique (2.5 points)

Pippo a écrit un programme C. Celui-ci compile, mais une erreur survient à l'exécution, qu'il ne comprend pas.

```
$ gcc puissance.c -o puissance.exe
$ puissance.exe
Entrer un nombre reel : 2.3
Entrer son exposant (entier positif) : 2
Segmentation fault
```

Question D. Expliquer brièvement ce que signifie ce message d'erreur (dernière ligne).

1 pt
9 min

Voici quelques lignes choisies du programme.

```
10 int main()
11 {
12     /* Déclaration et initialisation des variables */
13     double x;
14     int n;
15
16     :
17
18     :
19
20     :
21
22     printf("Entrer un nombre reel : ");
23     scanf("%lg", x);
24     printf("Entrer son exposant (entier positif) : ");
25     scanf("%d", n);
```

Question E. Que faut-il corriger ?

1 pt
9 min

Question F. Quelle option de compilation aurait dû utiliser Pippo pour obtenir de l'aide ?

0.5 pt
4 min

2 Faut-il aller au cinéma ? (4 points)

Je viens de gagner une place de cinéma pour un séance ce soir. Je dispose des informations suivantes, codées dans des variables entières, pour décider si je vais me rendre à cette séance ou rester chez moi :

- critique une critique du film lui attribuant une appréciation parmi MAUVAIS, MOYEN, BON.

- **acteurs** mon appréciation personnelle du choix des acteurs : MOYEN ou BON.
 - **distance** la distance approximative (en kilomètres) qui me sépare du cinéma.
- Vous utiliserez des constantes symboliques pour coder les appréciations et vous initialiserez les trois variables, **critique**, **acteurs**, **distance** à des valeurs de votre choix. Pour prendre ma décision je dispose de l'arbre de décision suivant.

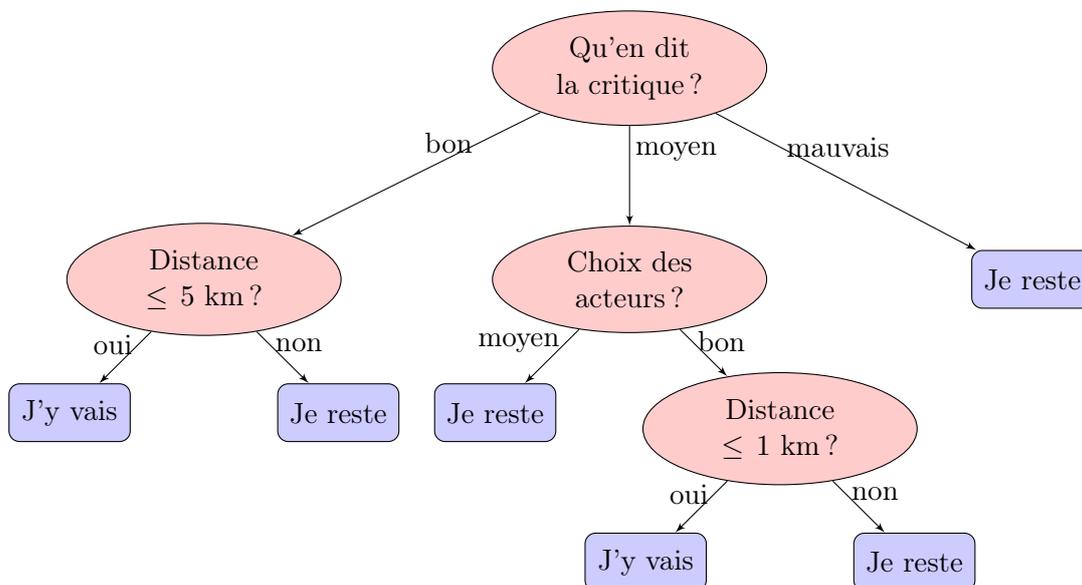


FIGURE 1 – Décider si je vais au cinéma

Question G. Écrire un programme implantant cet arbre de décision et affichant la décision à prendre (quel que soit le choix d'initialisation des variables).

4 pt
36 min

3 Puissance (4 points)

2 pt
18 min

Question H. Écrire un programme qui :

- demande à l'utilisateur d'entrer un nombre réel x ;
- calcule et affiche la valeur de x^2 .

Exemple d'exécution :

Entrer un nombre reel : 7.5e9
7.5e+09 exposant 2 = 5.625e+19

Question I. Réécrire ce programme (ou indiquer clairement les modifications à apporter) de manière à ce qu'il demande également à l'utilisateur l'exposant n (un entier positif) auquel doit être élevé x et affiche le résultat du calcul de x^n . Il faut programmer ce calcul (ne pas faire appel à une bibliothèque offrant la fonction ad-hoc). L'algorithme utilisé devra apparaître clairement.

2 pt
18 min

Exemple d'exécution :

Entrer un nombre reel : 12.35
Entrer son exposant (entier positif) : 10
12.35 exposant 10 = 8.2541e+10

4 Histogrammes (5 points)

Question J. Soit un entier n initialisé à une valeur positive de votre choix. Écrire un programme qui affiche une ligne contenant n fois le caractère '#' et se terminant par un saut de ligne. Pour cette question, vous pouvez répondre en ne donnant que le code de la fonction principale du programme (`main`).

1 pt
9 min

Soit un tableau d'entiers, initialisé à des valeurs de votre choix, toutes positives ou nulles, et dont la taille N sera définie par une constante symbolique. Dans les exemples suivants la taille du tableau est 10 et les valeurs contenues dans le tableau sont 3, 0, 2, 8, 9, 10, 3, 5, 5, 2.

On veut écrire un programme qui affiche sous forme d'histogramme (graphique en bâtons, ou graphique barres) les données du tableau : chaque barre de l'histogramme aura une longueur égale à la donnée représentée.

Question K. Écrire un programme qui affiche sous forme d'histogramme les données du tableau. Les barres seront dessinées horizontalement, à l'aide du caractère '#'.

3 pt
27 min

Exemple d'exécution :

Graphique barre 1 :

```
###  
  
##  
#####  
#####  
#####  
###  
#####  
#####  
##
```

Question L. Indiquer comment modifier le programme de la question précédente, de manière à demander à l'utilisateur le caractère qui sera utilisé pour dessiner les barres (à la place du #).

1 pt
9 min

Question bonus (plus difficile). Ajouter à votre programme un nouvel affichage en histogramme des données dont les barres progressent cette fois verticalement. Indication : avant cela, votre programme devra trouver le maximum parmi les données du tableau.

2 pt
18 min

Exemple d'exécution :

Graphique barre 2 :

```
      ##  
     ## ##  
    ## ## ##  
   ## ## ##  
  ## ## ##  
 ## ## ##  ## ##  
## ## ##  ## ##  
## ## ## ## ## ##  
## ## ## ## ## ## ##  
## ## ## ## ## ## ##
```