

3ETI, Examen [CSC2] Développement Logiciel en C CPE Lyon

2013-2014 (1ere session)
durée 3h

Tous documents et calculatrices autorisés.

Répondez aux questions sur une copie séparée

Le sujet comporte 12 pages

Le temps approximatif ainsi que le barème sont indiqués pour les grandes parties. Notez que le barème est donné à titre purement indicatif et pourra être adapté par la suite.

En cas de doute sur la compréhension de l'énoncé, explicitez ce que vous comprenez et poursuivez l'exercice dans cette logique.

Note préliminaire:

- Toutes les questions concernent le langage de programmation C dans le cadre du développement de logiciels sur architecture PC standard.
- Nous vous invitons à commenter le code et les réponses. En particulier, en cas d'ambiguïté de compréhension d'une question, ajoutez toutes remarques et illustrations supplémentaires permettant d'expliquer votre démarche.
- Sauf mention contraire explicite, on supposera que l'on dispose d'un système Linux standard fonctionnant correctement sur un PC récent 32 ou 64 bits (identiques aux conditions de TP des PC de CPE: *int* étant encodé sur 4 octets, pointeurs étant encodés respectivement sur 4(/8) octets sur 32(/64) bits).
- On supposera que le code C est compilé avec une version récente de gcc sous la norme C99 (ou ultérieure) identique aux conditions de TP des PC de CPE.
- On supposera dans chaque cas que les `#include` des en-tête standards nécessaires à la bonne compilation et exécution des programmes décrits sont correctement installés et appelés (ex. `stdio`, `stdlib`, `string`, `math`, etc).

1 Exercice d'application de développement logiciel

[1h max] 5 points

On souhaite réaliser une fonction permettant de calculer les racines réelles d'un polynôme d'ordre 2 à coefficients réels. On supposera que le polynôme p s'écrit sous la forme

$$p(x) = ax^2 + bx + c,$$

avec (a, b, c) les variables du polynôme.

On rappelle que les racines (ou zéros) de ce polynôme sont données par les nombres r tels que

$$r = (-b \pm \sqrt{\Delta}) / (2a),$$

avec $\Delta = b^2 - 4ac$ appelé discriminant. Cette relation est valable lorsque $\Delta \geq 0$. Si $\Delta < 0$, le polynôme possède deux racines imaginaires qui sont hors du cadre de notre fonction.

On définit les structures de données suivantes

```
//polynome du type: ax^2+bx+c
struct polynome
{
    float a,b,c;
};

//racines du polynome d'ordre 2
struct racine
{
    float r0,r1;
};
```

On pourra supposer que l'on initialise ces structures de la manière suivante dans le programme principal (les nombres choisis sont tout à fait arbitraires).

```
int main()
{
    struct polynome p={1,2,-1}; //un polynome quelconque
    struct racine r={0,0}; //initialisation par default
    ...
}
```

1.1 Fonction d'évaluation

On souhaite créer une fonction permettant d'évaluer la valeur du polynôme en une valeur x flottante donnée (c.a.d calculer la valeur de $p(x)$). Cette fonction s'appellera `eval`.

Question 1 *Un collègue vous propose la signature suivante pour cette fonction:*

```
float eval(struct polynome* p, float x);
```

Qu'en pensez-vous?

On supposera dans la suite des questions que le corps de la fonction `eval` est le suivant:

```
{
  assert (p!=NULL);
  return p->a*x*x + p->b*x + p->c;
}
```

Question 2 *Écrivez le contrat correspondant à la fonction `eval`.*

1.2 Recherche des racines

Nous souhaitons désormais coder la fonction `calcul_racine` permettant de trouver les deux racines du polynôme. Pour cela, on passera en paramètre (par adresse) de cette fonction le polynôme et une structure *racine* qui sera mise à jour dans la fonction.

Question 3 *Écrivez la signature (en tête) de la fonction `calcul_racine`.*

Question 4 *Pour traiter le cas où le discriminant est négatif, un collègue vous propose de placer les valeurs des deux racines à -1 pour indiquer qu'il s'agit d'un cas particulier. Qu'en pensez-vous?*

Vous décidez finalement que la fonction ne devra jamais être appelée si le discriminant est négatif. Dans le cas contraire, on considère qu'il s'agit d'une erreur de programmation.

Question 5 *Écrivez le contrat de la fonction `calcul_racine` satisfaisant ces contraintes. Quelle garantie pouvez vous avoir en sortie en vous servant de la fonction `eval` ?.*

Question 6 *Écrivez le corps de la fonction `calcul_racine` satisfaisant ce contrat.*

1.3 Garantie d'existence

Pour vous assurer que la fonction `calcul_racine` ne soit jamais appelée lorsque le discriminant du polynôme est négatif, vous souhaitez coder la fonction `discriminant_est_positif` qui retourne 1 lorsque $\Delta \geq 0$, et 0 sinon.

Question 7 *Écrivez la signature (en tête) de la fonction `discriminant_est_positif`.*

Question 8 *Écrivez un exemple concret d'utilisation de la fonction `discriminant_est_positif` et `calcul_racine` en venant compléter l'initialisation proposée précédemment dans la fonction `main` (un début possible de la fonction `main` est montré sur la p.2).*

Question 9 *Dans une démarche de développement par tests, écrivez quelques tests possibles permettant de valider le fonctionnement de la fonction `discriminant_est_positif`.*

2 Connaissances générales du développement logiciel

[30min max] 4 points

Question 10 *Pour chacune des propositions suivantes, dites si celle-ci est vraie ou fausse en justifiant ou en expliquant votre réponse. (Une réponse sans explication n'apporte pas de points).*

1. On inclut une en-tête de la bibliothèque C standard (ex. `#include <stdio.h>`) dans un fichier `f.c`. Lors de la compilation de `f.c`, seules les signatures des fonctions de cette bibliothèque qui sont effectivement utilisées par le programme sont copiées dans le fichier `f.c`.
2. Il est possible d'utiliser une énumération en tant qu'indice dans un tableau C.
3. Placer plusieurs instructions C sur une même ligne permet de rendre l'exécution du code plus rapide.
4. `git` est un logiciel de tests unitaires.
5. Un freeware est un logiciel libre.
6. Mettre beaucoup de fonctions `assert` dans un programme peut rendre l'exécution du programme plus lent chez le client.
7. Il est préférable d'écrire les tests d'une fonction avant même d'écrire le code de la fonction.
8. Il est possible (et parfois utile) de convertir une variable de type `int**` (pointeur avec 2 indirections) en un pointeur de type `void*`.
9. Lors de l'exécution, l'intégralité d'un programme est chargée dans la mémoire cache (cache de type L1, L2, ou L3).
10. Un fichier objet (`.o`) correspond au fichier obtenu après passage du préprocesseur sur le fichier `.c` correspondant.

3 Structs, adressage et portée des variables en C

[20min max] 3 points

Soit le code suivant (pour information, ce programme compile sans Warnings et 5 lignes sont affichées à son exécution).

```

struct vecteur
{
    int x;
    int y;
};
int main()
{
    struct vecteur v0={0,1};
    struct vecteur v1={4,7};

    struct vecteur v3;
    v3=v1;
    printf("%d,%d\n",v3.x,v3.y);

    struct vecteur* p0=&v0;
    struct vecteur* p1=&v1;
    p1=p0;
    printf("%d,%d , %d,%d\n",v1.x,v1.y,p1->x,p1->y);

    p0=&v0;
    p0->x=7;
    printf("%d,%d\n",p0->x,v0.x);

    struct vecteur* p4=NULL;
    int k=0;
    for (k=0;k<2;++k)
    {
        struct vecteur v4={k,2*k};
        p4=&v4;
    }
    printf("%d,%d\n",p4->x,p4->y);

    int *p=&v0.x;
    printf("%d\n",p[1]);

    return 0;
}

```

Question 11 *Qu'affiche ce programme? Commentez chacune des lignes affichées (si elle n'est pas triviale). Si un affichage est associé à un comportement indéterminé, indiquez directement indéterminé et expliquez pourquoi.*

4 Bonnes pratiques de programmation

[50min max] 5 points

Soit la définition des structures de données suivantes:

```
#define NBR_MAX_PAYS 5
enum type_pays {France,Allemagne,Angleterre,Suisse,Norvege};

struct temperature_info
{
    float min;
    float max;
    float moyenne;
};
struct type_climat
{
    struct temperature_info temperature;
    float pluviometrie_moyenne;
    int jour_enseillement;
};
struct pays_info
{
    enum type_pays pays;
    struct type_climat climat;
    int population;
    int PIB;
};
```

De plus, on suppose que l'on dispose d'une fonction `initialise_pays_info` déjà codée permettant d'initialiser un tableau de `pays_info`. La signature ainsi que le contrat de cette fonction est le suivant

```
/** Initialise pays info
 *
 * Initialise l'ensemble des donnees des pays.
 *
 * Prerequis:
 * - Un pointeur non NULL et non constant vers un tableau de pays_info de
 *   taille NBR_MAX_PAYS
 *
 * Garanties:
 * - L'ensemble des champs du tableau sont mis a jour avec des valeurs
 *   correctes.
 */
void initialise_pays_info(struct pays_info* pays);
```

4.1 Auto-documentation du code

Ces structures de données sont utilisées avec les codes suivants. Chaque code compile sans Warnings et est fonctionnel, cependant ils ne satisfont pas intégralement les règles de bonne programmation concernant la **lisibilité et l'auto-documentation du code**.

Question 12 Indiquez comment modifier les 4 programmes suivants (tout en préservant leurs comportements) pour qu'ils satisfassent au mieux les règles de lisibilité. Pour chacun des programmes, vous désignerez la ou les lignes à modifier, et expliquerez la modification à appliquer avec le code correspondant.

4.1.1 Programme 1

```

1 int main()
2 {
3     struct pays_info pays[NBR_MAX_PAYS];
4     initialise_pays_info(pays);
5
6     //temperature min de la France
7     float min_france=pays[0].climat.temperature.min;
8     //temperature max de la Suisse
9     float max_suisse=pays[3].climat.temperature.max;
10
11    printf(" min France: %f \n max Suisse: %f \n",min_france,max_suisse);
12
13    return 0;
14 }
```

4.1.2 Programme 2

```

1 float calcul_enseillement_moyen(struct pays_info* pays)
2 {
3     int enseillement_total=0;
4
5     int k=0;
6     for (k=0;k<NBR_MAX_PAYS;++k)
7     {
8         enseillement_total += pays[k].climat.jour_enseillement;
9     }
10    float enseillement_moyen=(float)enseillement_total/NBR_MAX_PAYS;
11    return enseillement_moyen;
12 }
13
14 int main()
15 {
16    struct pays_info pays[NBR_MAX_PAYS];
17    initialise_pays_info(pays);
18
19    float enseillement_moyen=calcul_enseillement_moyen(pays);
20    printf("enseillement moyen %f jours \n",enseillement_moyen);
21
22    return 0;
23 }
```

4.1.3 Programme 3

```

1 int main()
2 {
3     int t=0;
4     int k=0;
5
6     struct pays_info pays[NBR_MAX_PAYS];
7     initialise_pays_info(pays);
8
9     //calcul la population totale
10    for (k=0;k<NBR_MAX_PAYS;++k)
11        t+=pays[k].population;
12
13    printf("Population totale: %d \n",t);
14
15    return 0;
16 }
```

4.1.4 Programme 4

```

1 int main()
2 {
3     struct pays_info pays[NBR_MAX_PAYS];
4     initialise_pays_info(pays);
5
6     //calcul le nombre de jour moyen d'ensoleillement pour les pays ayant une
7     //temperature moyenne superieur a 12 degres.
8     int k=0;
9     float jour_ensoleillement=0.0f;
10    int nbr_pays_concernes=0;
11    for (k=0;k<NBR_MAX_PAYS;++k)
12    {
13        if (pays[k].climat.temperature.moyenne>12)
14        {
15            jour_ensoleillement += pays[k].climat.jour_ensoleillement;
16            nbr_pays_concernes++;
17        }
18    }
19    jour_ensoleillement/= (float) nbr_pays_concernes;
20
21    printf("Nombre de jour ensoleillement moyen: %f\n",jour_ensoleillement);
22
23    return 0;
24 }
```

4.2 Correction de codes

Certains des programmes suivants ont un comportement potentiellement **indéterminés** (lié à des erreurs de mémoire, ou à d'autres types de comportement).

Question 13 Pour chacun des 5 programmes suivants, désignez si celui-ci possède un comportement déterministe (c.a.d correct, prévisible et non dépendant du système) ou non. Dans le cas où ce comportement est non prévisible, désignez clairement la (ou les) ligne(s) fautive(s), et proposez une solution permettant de corriger le problème.

4.2.1 Programme 1

```

1 void affiche_population(const struct pays_info* pays)
2 {
3     int k=0;
4     for (k=0;k<NBR_MAX_PAYS;++k)
5         printf("%d\n",pays[k].population);
6 }
7 int main()
8 {
9     struct pays_info pays[NBR_MAX_PAYS];
10    initialise_pays_info(pays);
11    affiche_population(pays);
12    return 0;
13 }
```

4.2.2 Programme 2

```

1 int main()
2 {
3     struct pays_info pays[NBR_MAX_PAYS];
4     initialise_pays_info(pays);
5
6     if(pays[France].climat.temperature.max==24.3)
7         puts("Temperature est de 24.3 degres");
8
9     return 0;
10 }
```

4.2.3 Programme 3

```

1 struct tableau_pays
2 {
3     struct pays_info pays[NBR_MAX_PAYS];
4 };
5
6 int main()
7 {
8     struct tableau_pays tableau;
9     initialise_pays_info(tableau.pays);
10    initialise_pays_info(&tableau.pays);
11    return 0;
12 }
```

4.2.4 Programme 4

```

1 void affiche_temperature_min(const struct temperature_info* temp)
2 {
3     printf("temperature min: %f\n",temp->min);
4 }
5 int main()
6 {
7     struct pays_info pays[NBR_MAX_PAYS];
8     initialise_pays_info(pays);
9     affiche_temperature_min(pays[France].climat.temperature);
10    return 0;
11 }

```

4.2.5 Programme 5

```

1 void affiche_temperature_min(const struct temperature_info* temp)
2 {
3     printf("temperature min: %f\n",temp->min);
4 }
5
6 int main()
7 {
8     struct temperature_info *temp;
9     temp->min=-8.0f;
10    temp->max=25.1f;
11    temp->moyenne=11.2f;
12
13    affiche_temperature_min(temp);
14
15    return 0;
16 }

```

4.3 Méthodologie de programmation

On souhaite réaliser la fonction `climat_froid` qui retourne un pointeur sur une structure `type_climat` pour laquelle la **température min est la plus basse** par rapport à l'ensemble des autres pays. Cette fonction prendra en tant qu'argument la variable `pays` désignant le tableau de `pays_info` initialisé de cette manière

```

int main()
{
    struct pays_info pays[NBR_MAX_PAYS];
    initialise_pays_info(pays);

    ...
}

```

Question 14 Écrivez le contrat correspondant à la fonction `climat_froid`.

Question 15 Écrivez la signature (en-tête) de cette fonction.

Question 16 Écrivez l'algorithme que doit suivre le corps de cette fonction.

Question 17 Écrivez le corps (code de l'implémentation) de cette fonction.

5 Makefile et compilation

[20min max] 3 points

Dans un **même répertoire**, on dispose de 9 fichiers: p0.c, p1.c, p1.h, p2.c, p2.h, p3.c, p4.c, p4.h, p5.c. Les codes de ces fichiers sont les suivants (les en-têtes de type `#include <stdio.h>` ne sont pas montrées, et les codes n'ont pas d'erreurs de syntaxe):

p0.c

```
#include "p1.h"
#include "p2.h"
int main()
{
    p1(); p2();
    return 0;}
```

p1.h

```
void p1 ();
```

p1.c

```
#include "p1.h"
void p1 () {puts ("p1");}
```

p3.c

```
#include "p1.h"
#include "p4.h"
int main()
{
    p1(); p4();
    return 0;}
```

p2.h

```
void p2 ();
```

p2.c

```
#include "p2.h"
void p2 () {puts ("p2");}
```

p5.c

```
int main()
{
    puts ("Bonjour");
    return 0;}
```

p4.h

```
void p4 ();
```

p4.c

```
#include "p4.h"
void p4 () {puts ("p4");}
```

On souhaite compiler et créer l'exécutable du programme correspondant à p0. On suppose pour cela que l'on dispose d'une ligne de commande dans le répertoire courant où est l'ensemble des fichiers.

Question 18 Écrivez la ou les ligne(s) de commande(s) à taper (sans utiliser de Makefile) pour créer l'exécutable correspondant à p0. On appellera le fichier exécutable simplement p0. (Si certains fichiers n'ont pas besoin d'être compilés pour créer l'exécutable p0, alors on ne les compilera pas).

On considère désormais le fichier Makefile suivant, également présent dans ce même répertoire.

```
CFLAGS=-Wall -Wextra -g

all: p0

p0: p0.o p1.o p2.o
p3: p3.o p1.o p4.o
p5: p5.o p6
    gcc p5.o
p6: p6.o

p0.o: p0.c p1.h p2.h
p1.o: p1.c p1.h
p2.o: p2.c p2.h
p3.o: p3.c p1.h p4.h
p4.o: p4.c p4.h
p5.o: p5.c
p6.o: p5.o
    cp p5.o p6.o

clean:
    rm *.o
```

On tape en ligne de commande

```
$ make clean
$ make
```

Après exécution de cette commande, un ou plusieurs fichiers objets (.o) sont créés dans le répertoire. De même, un ou plusieurs fichiers exécutables sont créés dans ce répertoire.

Question 19 *Quels sont les fichiers .o et fichiers exécutables présents dans le répertoire.*

On tape désormais en ligne de commande

```
$ make clean
$ make p3
```

Question 20 *(Même question): Quels sont les fichiers .o et fichiers exécutables présents dans le répertoire.*

On tape désormais en ligne de commande

```
$ make clean
$ make p5
```

Question 21 *(Même question): Quels sont les fichiers .o et fichiers exécutables présents dans le répertoire.*