

# ESIR SYS 2012-2013

## Contrôle en Classe n.2

---

Nom: \_\_\_\_\_

Prénom: \_\_\_\_\_

**Durée: 1h**

**Notation: sur 20 points**

### Partie A (MCQ / QCM, 10 points) :

#### Instructions:

Entourez la lettre de la réponse que vous pensez être juste.

#### Barème:

+0.5 pour chaque réponse correcte

-0.5/m pour chaque réponse fausse (où m+1 est le nombre de réponses possibles)

#### Q1 : Si p est un pointeur qui pointe vers une chaîne de caractères

A : p contient une adresse

B : p contient un caractère

C : p contient un registre

#### Q2 : Les variables allouées dynamiquement sans que leur cycle de vie (création, destruction) ne soit synchronisées avec les appels de procédures sont :

A : allouées sur la pile d'appel

B : allouées sur le tas

C : allouées dans le segment de code

#### Q3 : Laquelle des fonctions suivantes permet-elle de réserver une zone mémoire en C ?

A : reserve(..)

B : allocate(..)

C : malloc(..)

**Q4 : Dans la notation du cours, considérons la définition de type suivante**

```
TYPE element =  STRUT
                  CAR c1, c2, c3 ;
                  POINTEUR element suiv ;
                  FSTRUCT ;
```

**Si elem est un pointeur vers une variable de type element, comment se traduit la ligne suivante en assembleur ?**

```
ec^.c3 = 0 ;
```

- A :   MOV BX,EC  
      MOV [BX+1],0
- B :   MOV EC,BX  
      MOV [BX+1],0
- C :   MOV BX,EC  
      MOV [BX+2],0

**Q5 : Toujours avec le type élément défini à la question 4, considérons le code suivant (dans la notation du cours) :**

```
VAR POINTEUR element x, y, z ;
x^.c1 = 1 ;
y^.c1 = 2 ;
z^.c1 = 4 ;
x = z
y = x ;
y^.c1 = x^.c1 + z^.c1 ;
```

**Que vaut y^.c1 à la fin de ce code ?**

- A : 7
- B : 8
- C : 5
- D : 3

**Q6 : Deux pointeurs peuvent-ils référencer la même variable ?**

- A : oui
- B : non
- C : pas en C, mais oui en Java

**Q7 : L'utilisation d'un ramasse miette (garbage collector)**

- A : évite d'avoir à allouer les variables dynamiques
- B : évite d'avoir à libérer les variables dynamiques
- C : évite d'avoir à transférer les informations du tas vers la pile

**Q8 : Que fait un éditeur de lien ?**

A : Il forme un fichier exécutable à partir des différents modules objets en résolvant les références entre modules.

B : Il implante le programme en mémoire et lance son exécution.

C : Il permet aux développeurs de modifier de façon ergonomique les liens binaires d'un processus en exécution.

**Q9 : Au moment de l'édition de lien l'information associée aux identificateurs de constantes dans la table d'identificateurs d'un programme :**

A : est traduite

B : reste inchangée

C : est modifiée en fonction de la nouvelle position du segment de donnée

**Q10 : À quoi sert le vecteur de translation au moment de l'édition de liens**

A : Le vecteur de translation est ajouté à la table des identificateurs

B : Le vecteur de translation permet de calculer l'image système de l'éditeur de lien par une addition vectoriel

C : Le vecteur de translation identifie les positions mémoires de l'exécutable qui doivent être modifiées au moment de l'édition des liens

**Q11 : Si le module A exporte la variable X, et utilise la variable S définie dans le module B**

A : X se trouve dans la table des points d'entrée de B

B : S se trouve dans la table des points d'entrée de A

C : S se trouve dans la table des externes de A

**Q12 : Dans l'exemple assembleur suivant**

**.DATA**

**X DB 5 DUP(?)**

**N EQU 10**

**Y DB 7 DUP(?)**

**N1 EQU N**

**Z EQU Y**

**X est associé à la valeur zéro dans la table des identificateurs du programme. Cette valeur associée est elle ?**

A : translatable

B : absolue

**Q13 : Si le module R exporte la variable S, qui est utilisé par le module T**

A : S se trouve dans le vecteur d'externalisation de T

B : S se trouve dans la table des points d'entrée de R

C : S se trouve dans la table des externes de R

**Q14 : Quelle est la commande Unix pour afficher le répertoire de travail courant**

A : cd

B : echo \$WD

C : pwd

**Q15 : Vers où pointe le chemin /private/student/5/25/2363425/../../25/test/./hop/**

A : /private/student/5/25/hop/

B : /private/student/5/2363425/25/test/hop/

C : /private/student/2363425/25/test/

**Q16 : Pour enlever les droits d'accès à un fichier à tous les utilisateurs en dehors de l'utilisateur courant et du groupe associé au fichier, quelle commande devez vous utiliser ?**

A : chmod a-rwe f

B : chmod a-rwx f

C : chmod o-rwx f

**Q17 : La ligne de commande « aaa < bbb | ccc » permet**

A : De diriger la sortie de la commande aaa vers la commande bbb, puis de sauvegarder le résultat dans le fichier ccc.

B : De diriger l'entrée de aaa vers la sortie de bbb, et de sauvegarder le résultat dans le fichier ccc.

C : D'utiliser le fichier bbb comme entrée de la commande aaa, et de diriger la sortie de la commande aaa vers l'entrée de la commande ccc.

**Q18 : Pour indiquer le programme de shell qui doit exécuter un script shell, on utilise sur la première ligne**

A : !#/bin/sh

B : #!/bin/sh

C : #/!bin/sh

**Q19 : Si on appelle une commande script de la façon suivante :**

command a1 a2 a3 a4

Que contiendra la variable \$0 ?

A : a1

B : a2

C : command

**Q20 : Quel commande Unix permet de rechercher une chaîne de caractère dans un ou plusieurs fichier ?**

A : find

B : grep

C : search

## Partie B Problèmes (10 points) :

### Question 1: (3 points)

Les deux modules M1 et M2 ci-dessous sont écrits dans le langage d'assemblage du 8086.

```
;Module M1                                ;Module M2
    .DATA                                    .DATA
    EXTRN  A:WORD,B:WORD,C:WORD            EXTRN  E:ABS
    PUBLIC D,E                              PUBLIC A,B,C

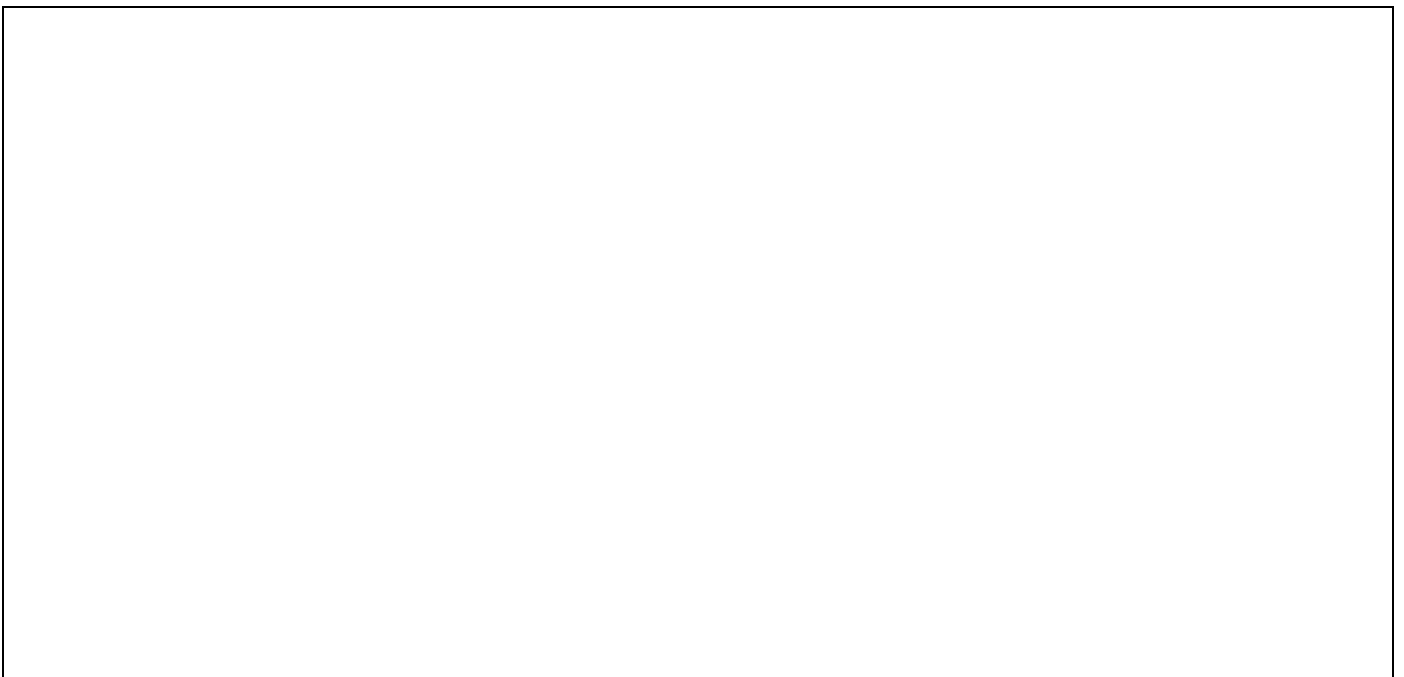
Z     EQU 5
V     DW  Z DUP (?)
      DW  OFFSET C
D     EQU  V
E     EQU  Z
      .CODE
      MOV  SI,Z
      MOV  AX,E
      ADD  AX,A
      MOV  BX,B
      MOV  V[SI],BX
      END

Z     EQU 5
A     DW  2 DUP (?)
C     DW  Z DUP (?)
B     DW  OFFSET C
      .CODE
DEB:  MOV  AX,Z
      MOV  BX,B
      ADD  AX,[BX]+E
      MOV  C,AX
      END  DEB
```

1.1) Donner les tables des identificateurs construites lors de l'assemblage de M1 et M2.



1.2) Donner les tables des points d'entrée et des externes de chaque module



## Question 2: (3 points)

Soit l'extrait de code C suivant:

```
typedef struct {
    int x;
    int y;
} vecteur;

typedef struct maillon_struct {
    vecteur v;
    struct maillon_struct * maillon_suiv;
} maillon;

int main(int argc, char *argv[]) {
    maillon* ptr_maillon;
    vecteur v1;

    v1.x = 1;
    v1.y = 2;

    ptr_maillon = malloc(sizeof(maillon));
    // [... à compléter ...]
}
```

Compléter le code de façon à :

- initialiser le maillon nouvellement créé avec la valeur de la variable v1 ;
- à faire pointer le maillon vers lui même ;
- et finalement à libérer la zone mémoire réservée pour le maillon.

On rappelle que l'opérateur de déréférencement se note '\*' en C (c.-à.-d. '\*p' dénote la variable pointée par p). Par ailleurs, dans le cas de variable structurées, la syntaxe 'p->a' peut être utilisée comme raccourci pour '(\*p).a'.

(Indiquer uniquement les lignes ajoutées.)

**Question 3: (4 points)**

Écrire un programme de commande : prog a b c qui réalise les traitements illustrés ci-dessous

