

Final LO22 P2024 – Durée 1h30

L'examen se compose en trois parties.

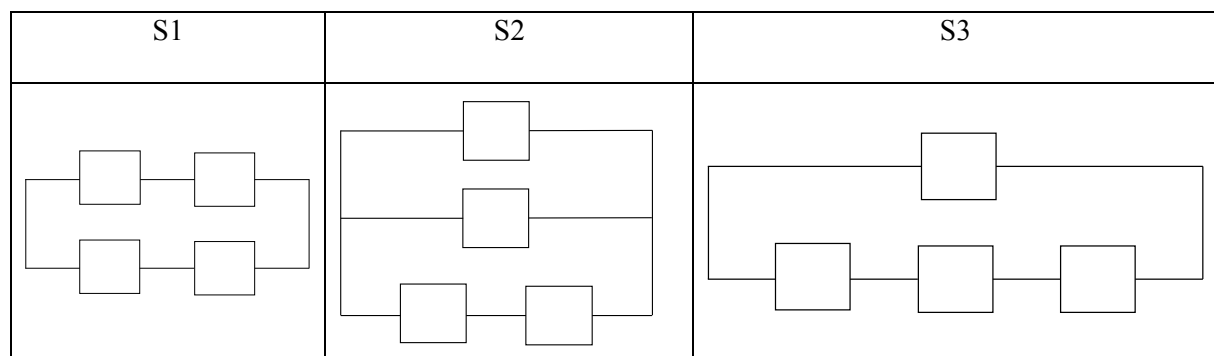
- *Matériel autorisé :*

- *Une feuille de note personnelles A4 recto-verso*
- *La fiche des symboles du langage B*
- *La calculatrice*

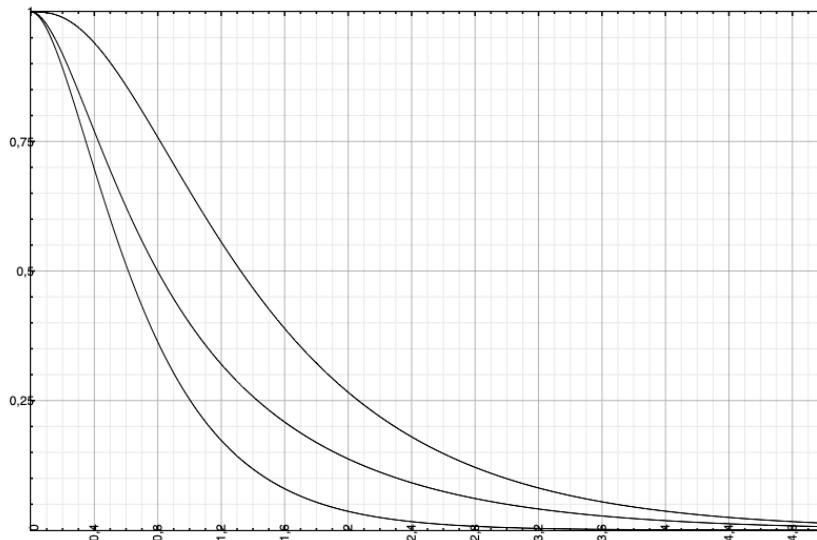
Partie I

Comparaison de trois architectures redondantes

On s'intéresse à trois architectures possibles d'un système à quatre composants identiques de même taux de défaillance λ .



- 1) Calculer les défiabilités (1-fiabilité) $U1, U2, U3$ des trois architectures en fonction de la fiabilité r d'un composant unique.
- 2) En déduire les fiabilités $R1, R2, R3$ en fonction de r , puis de λ et du temps t .
- 3) Utilisant le résultat de la question 1) donner le comportement aux temps courts ($\lambda t \ll 1$ on ne conserve que le premier monôme en λt de degré >0) des trois architectures, en déduire le classement des systèmes S1, S2, S3 dans l'ordre de leur fiabilité décroissante.
- 4) Utilisant le résultat de la question 2) donner le comportement aux temps longs ($\lambda t \gg 1$ donc $r \ll 1$ on ne conserve que le monôme de degré le plus bas en r) des trois architectures, en déduire le classement des systèmes S1, S2, S3 dans l'ordre de leur fiabilité décroissante. Conclure en identifiant les trois systèmes dans la figure ci-dessous (à recopier sur votre feuille de réponse).



Partie II

Soit le programme suivant :

```
int main(int argc, char *argv[])
{
    long nombreDAmis = 0, i = 0;
    long* ageAmis = NULL;

    printf("Combien d'amis avez-vous ? ");
    scanf("%ld", &nombreDAmis);

    if (nombreDAmis > 0)
    {
        ageAmis = malloc(nombreDAmis * sizeof(long));
        if (ageAmis == NULL)
        {
            exit(0);
        }

        for (i = 0 ; i < nombreDAmis ; i++)
        {
            printf("Quel age a l'ami numero %ld ? ", i + 1);
            scanf("%ld", &ageAmis[i]);
        }

        printf("\n\nVos amis ont les ages suivants :\n");
        for (i = 0 ; i < nombreDAmis ; i++)
        {
            printf("%ld ans\n", ageAmis[i]);
        }
    }
}
```

```

    free(ageAmis);
}

return 0;
}

```

- 1) Construisez le graphe de contrôle associé à ce programme ;
- 2) Calculez le nombre cyclomatique ;
- 3) Définissez les jeux de tests minimaux pour obtenir :
 - Une couverture des instructions,
 - Une couverture des branches,
 - Une couverture des i-chemins avec $i=1$.

Partie III

On propose de modéliser un ensemble de biens qui peuvent être proposés au prêt dans une machine **Prêt**. Pour cela on se donne deux ensembles abstraits : BIEN qui représente les biens et PERSONNE pour représenter les personnes qui peuvent prêter ou bien emprunter un bien. On introduit également un ensemble énuméré TYPE dont les valeurs représentent la nature du bien prêté, ici on se limitera à trois sortes de biens : voiture, livre et disque.

Les variables de la machine Prêt sont : biens qui représente l'ensemble des biens dans la base, personnes qui représente l'ensemble des personnes dans la base, type_de_bien qui associe à chaque bien de la base son type, appartient_à qui relie les biens à leur propriétaire et a_pour_emprunteur qui relie un bien à la personne qui l'emprunte. Un bien dans la base a toujours un unique propriétaire mais n'est pas forcément emprunté, s'il est emprunté, alors c'est au plus par une seule personne.

1. Écrire en langage B l'invariant de la machine **Prêt** en justifiant le type des variables (en particulier pour les relations, on précisera si ce sont des fonctions, si elles sont totales, partielles, injectives ou surjectives.)
2. Donnez en langage B les expressions pour représenter :
 - (a) l'ensemble **disques** des disques de la base;
 - (b) l'ensemble **déjà_loués** des objets qui sont déjà loués;
 - (c) la relation **loue_a** entre deux personnes p et p' qui est vraie lorsque la personne loue au moins un bien à la personne p' .
3. Donner les formules du langage B qui expriment les contraintes suivantes :
 - (a) une personne ne peut pas emprunter ce qui lui appartient ;
 - (b) une même personne ne peut emprunter strictement plus de 5 disques.
4. Écrire une opération location qui étant donné un bien qui n'est pas encore loué et une personne qui remplit les conditions pour louer ce bien, introduit la nouvelle location dans la base.
5. Énoncer et simplifier les obligations de preuve (POs) à vérifier pour que la condition énoncée à la question (3b) soit préservée par l'opération de location.