



Département Mathématiques et Informatique

Concours d'accès en 1^{ère} Année du cycle d'ingénieur

Session Juillet 2022

Génie du Logiciel et Systèmes Informatiques Distribués : GLSID

Ingénierie Informatique, Big Data et Cloud Computing : II-BDCC

Durée de l'épreuve : 2H

Consignes importantes à respecter :

- 1- L'épreuve est un QCM dont les réponses doivent être reportées dans la grille de réponses fournie.
- 2- Dans la grille de réponses, il est nécessaire de préciser votre numéro d'examen dans la grille relative au numéro d'examen en remplissant les cases correspondantes aux 4 chiffres composant à votre numéro d'examen. Exemple : pour un numéro d'examen **0129**, il sera représenté dans la grille comme suit :

Num Examen			
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9

- Le **premier** chiffre **0** du numéro de l'examen (0129) sera représenté en remplissant la case portant le chiffre **0** de la **première** colonne de grille
- Le **deuxième** chiffre **1** du numéro de l'examen (0129) sera représenté en remplissant la case portant le chiffre **1** de la **deuxième** colonne de grille
- Le **troisième** chiffre **2** du numéro de l'examen (0129) sera représenté en remplissant la case portant le chiffre **2** de la **troisième** colonne de grille
- Le **quatrième** chiffre **9** du numéro de l'examen (0129) sera représenté en remplissant la case portant le chiffre **9** de la **quatrième** colonne de grille

- 3- Pour toutes les questions du QCM, une seule réponse est correcte parmi 4 choix possibles : A, B, C, et D
- 4- Pour répondre aux questions, dans la grille de réponses, Il faut remplir la case correspondante à la bonne réponse correspondante à chaque question. Exemple : pour un QCM à 3 Questions Q1, Q2 et Q3 dont les bonnes réponses respectives sont B, D et C, la grille sera remplie de la manière suivante :

Q - 1	A	B	C	D
Q - 2	A	B	C	D
Q - 3	A	B	C	D

- 1- Le barème appliqué dans la correction est le suivant :
 - Bonne réponse : 2 Pts
 - Mauvaise réponse : - 1
 - Aucune réponse : - 0,5
 - Réponse incohérente (plusieurs réponses choisies) : - 1
- 2- Cocher la case qui correspond à la réponse correcte sur la fiche de réponses.
- 3- L'utilisation des documents, téléphones portables, calculatrices, appareils électroniques, correcteur « blanco », sont strictement interdits
- 4- Aucune explication supplémentaire ne vous sera donnée pendant l'examen. Il vous appartient de prendre vos décisions selon votre compréhension et votre analyse.
- 5- A la fin de l'examen, il faut rendre : l'énoncé de l'épreuve, la grille de réponse. Ces deux documents doivent être enveloppés dans une double, qui vous sera fournie, pourtant votre nom, votre prénom et votre numéro d'examen.

QCM :

1. Qu'est-ce qu'un algorithme :

- A) Un organigramme
- B) Pseudo-code
- C) Un ensemble d'instructions pour résoudre un problème
- D) Une décision

2. Qu'est-ce que la compilation ?

- A) Exécution d'un programme
- B) Traduction d'un code source en un code machine
- C) Edition d'un programme
- D) Correction des erreurs d'un programme

3. Soit l'extrait de code suivant :

```
int A=0, B=2, C=5, AIDE;  
AIDE=A;  
A=C;  
C=B;  
B=AIDE;  
printf("A = %d\tB = %d\tC = %d\n", A, B, C);
```

L'exécution de ce code produit l'affichage :

- A) A = 0 B = 2 C = 5
- B) A = 2 B = 5 C = 0
- C) A = 2 B = 0 C = 5
- D) A = 5 B = 0 C = 2

4. Soit l'extrait de code suivant :

```
int A=5, B=2;  
printf("Division:%d, Reste: %d, Quotient:%f\n", A/B, A%B,(float)A/B);  
return 0;
```

L'exécution de ce code produit l'affichage :

- A) 0
- B) Erreur d'exécution
- C) Division:2,Reste: 1,Quotient:2.500000
- D) Division:2, Reste:1,Quotient:2

5. Soit l'extrait de code suivant :

```
int N,I,S;
double P;
printf("Donnez un nombre: ");
scanf("%d", &N);
S=0;
P=1;
I=1;
do
{
    S += I;
    P*= I;
    I++;
}
while(I<=N);
printf("S=%d, P=%.0f, M= %.2f\n", S, P, (float)S/N);
return 0;
```

L'exécution de ce code pour N=4 produit l'affichage:

- | | |
|--------------------------|-------------------------|
| A) S=10, P=24, M= 2.50 | B) S=15, P=120, M= 3.00 |
| C) Erreur de compilation | D) 0 |

6. On considère le programme suivant :

```
void f8(int a, int *b)
{
    a++;
    *b=-a++;
    a=*b;
}

int main(){
    int a=8, b=10;
    f8(a,&b);
    printf("%d %d",a,b);
    return 0;
}
```

Quel est le résultat affiché par le programme ? :

- | | |
|---------|----------|
| A) 8 -8 | B) 9 -9 |
| C) 8 -9 | D) 8 -11 |

7. Soit le code suivant :

```
int main()
{
    char tab[] = "en-set-mohammedia";
    int i = 0;
    while (tab[i] != '\0') i++;
    printf("i = %d", i);
    return 0;
}
```

L'exécution de ce code produit l'affichage:

- | | |
|-------|---|
| A) 16 | B) 17 |
| C) 18 | D) Rien du tout, il y a une boucle infinie, '\0' n'est pas présent dans "en-set-mohammedia" |

8. Soit le code suivant :

```
int main()
{
    int i, somme, tab[5];
    tab[0] = 1;
    somme = 0;
    for( i=1 ; i < 5 ; i++ ) {
        tab[i] = tab[i-1] * 2;
    }
    i = 0;
    while(tab[i] < 16) {
        somme = somme + tab[i];
        i++;
    }
    printf("somme = %d - ",somme);
    printf("tab = %d",tab[0]);
    for( i=1 ; i < 5 ; i++ ) {
        printf(",%d",tab[i]);
    }
    return 0;
}
```

L'exécution de ce code produit l'affichage:

- A) somme = 15 - tab = 1,2,4,8,16
- B) somme = 31 - tab = 1,2,4,8,16
- C) somme = 31 - tab = 1,2,3,4,5
- D) somme = 15 - tab = 1,2,3,4,5

9. On considère la fonction fibo de la suite de fibonnaci suivante :

```
int fibo(int n){  
    if (n<=1) return n;  
    else  
        return fibo(n-1)+fibo(n-2);  
}
```

Elle a une complexité temporelle:

- A) Logarithmique
- B) Linéaire
- C) Quasi-linéaire
- D) Exponentielle

10. Quelle est le retour de l'appel de la fonction fibo(7) :

```
int fibo(int n){  
    if (n<=1) return n;  
    else  
        return fibo(n-1)+fibo(n-2);  
}  
int main()  
{  
    printf("%d", fibo(7));  
    return 0;  
}
```

- A) 8
- B) 13
- C) 21
- D) la fonction ne s'arrête pas

11. Lequel des algorithmes de tri suivants est un algorithme du paradigme « diviser pour régner »?:

- A) Tri par sélection
- B) Tri par fusion
- C) Tri bulle
- D) Tri par insertion

12. Soit l'extrait de code suivant :

```
int a=17,b=3,r=1;  
r = a - a / b * 4 % 3 + r;
```

Lequel des choix suivants est le bon ordre d'évaluation pour l'expression ci-dessus :

- A) / * % + - =
- B) / * % - + =
- C) = / * % - +
- D) / % * - + =

13. Soit le code suivant :

```
int i=0;  
while(++i <= 6);  
printf("%d ",i);
```

L'exécution de ce code produit l'affichage:

- A) 0 1 2 3 4 5 6
- B) 1 2 3 4 5 6 7
- C) 7
- D) 6

14. On considère le programme suivant:

```
#include <stdio.h>  
void multiplier(int *,int *);  
int main()  
{  
    int a=3,b=4;  
    multiplier(&a,&b);  
    printf("%d %d",a,b);  
    return 0;  
}
```

```
void multiplier(int *a,int *b){
    int *i=a,*j=b;
    (*i)++;
    (*j)++;
    *i=*i**i;
    *j=*j**j;
}
```

L'exécution de ce code produit l'affichage:

A) 3 4

B) 6 16

C) 4 5

D) 16 25

15. On considère le code suivant:

```
#include <stdio.h>
void afficher(int);

int main()
{
    int nb=0;
    afficher(nb);
    return 0;
}

void afficher(int nb){
    if(nb==6)
        return;
    else
        printf("%d ",nb);
        afficher(nb++);
}
```

L'exécution de ce code produit l'affichage:

A) 0 1 2 3 4 5

B) Boucle infinie

C) 0 1 2 3 4 5 6

D) 1 2 3 4 5 6

16. Quel sera le résultat de ce programme ?

```
#include <stdio.h>
int maFonction(int);
int main()
{
    int a,b;
    a=maFonction(543);
    b=maFonction(234);
    printf("%d %d",a,b);
    return 0;
}
int maFonction(int n){
    static int s=0;
    int d;
    while(n!=0){
        d=n%10;
        n=n/10;
        s+=d;
    }
    return s;
}
```

- A) 12 21 B) 12 9
C) 5 2 D) 12 12

17. Quel sera le résultat de ce programme ?

```
int a=2,b=1,c=3;
a+=c=b=10;
printf("%d %d %d",a,b,c);
```

- A) 10, 3, 10 B) 12, 10, 10
C) Erreur de syntaxe D) 10, 12, 10

18. Quel sera le résultat de ce programme ?

```
int i,nb;
char *ch="ENSET";
nb=strlen(ch);
for(i=0;i<nb;i++){
    printf("%s ",ch++);
}
```


21. On considère le programme suivant :

```
int main()
{
    int M[3][3]={3,6,4},
                {10,2,3},
                {7,5,1}
                };

    int i,j,s;
    for(i;i<3;i++){
        s=0;
        for(j=0;j<3;j++){
            s+=M[j][i];
        }
        printf("%d ",s);
    }
    return 0;
}
```

Quel est le résultat affiché par le programme :

- A) 13 15 13 B) 20 13 8
C) 3 2 1 D) 13 28 41

22. On considère le programme suivant :

```
#include <stdio.h>
int maFonction(int,int []);
int main()
{
    int tab[6]={3,7,8,10,11};
    int i,j,a=9,p;
    p=maFonction(a,tab);
    printf("%d ",p);
    return 0;
}

int maFonction(int a,int tab[]){
    int p=0;
    while(p<6 && a>=tab[p]) p++;
    return p;
}
```

Quel est le résultat affiché par le programme ?

A) 0 B) 6

C) 2 D) 3

23. Quelle est la fonction qui reçoit en paramètres une chaîne de caractères et retourne le nombre de lettres alphabétiques dans cette chaîne ?

<p>A)</p> <pre>int maFonction(char ch[]){ int n=0,i; for(i=0;i<strlen(ch);i++){ if((ch[i]>='a' && ch[i]<='z') (ch[i]>='A' && ch[i]<='Z')){ n++; } } return n; }</pre>	<p>B)</p> <pre>int maFonction(char ch[]){ int n=0,i; for(i=0;i<strlen(ch);i++){ if((ch[i]>='a' && ch[i]<='z')){ n++; } } return n; }</pre>
<p>C)</p> <pre>int maFonction(char ch[]){ int n=0,i; for(i=0;i<strlen(ch);i++){ if((ch[i]>='a' && ch[i]<='z') && (ch[i]>='A' && ch[i]<='Z')){ n++; } } return n; }</pre>	<p>D)</p> <pre>int maFonction(char ch[]){ int n=0,i; for(i=0;i<strlen(ch);i++){ if((ch[i]>='a' ch[i]<='z') && (ch[i]>='A' ch[i]<='Z')){ n++; } } return n; }</pre>

24. On considère la fonction suivante :

```
void f1(int a)
{
if(a>0) f1(a-1);
printf("%d ",a);
}
```

Quel est le résultat qui sera affiché suite à l'appel de f1(5):

A) 1 2 3 4 5 B) 0 1 2 3 4 5

C) 5 4 3 2 1 D) 5 4 3 2 1 0

25. Soit les structures de données suivantes :

<pre>typedef struct Element{ int data; struct Element *next; }Element;</pre>	<pre>typedef struct{ Element *First; Element *Last; }File;</pre>
--	--

On dispose aussi des deux fonctions suivantes pour manipuler une file d'attente :

- La fonction **enfiler** permet d'ajouter un élément à la fin de la file d'attente
- La fonction **defiler** permet d'enlever le premier élément de la file d'attente et retourner sa valeur

On considère la fonction suivante :

```
void f3(int n)
{
    int i ,a,b;
    File fil ;
    fil.Last=fil.First=NULL ;
    enfiler(&fil,0);
    enfiler(&fil,1);
    for (i = 0; i< n; i++){
        a=defiler(&fil);
        b=defiler(&fil);
        enfiler(&fil,b);
        enfiler(&fil,a + b);
        printf("%d",a);
    }
}
```

Quel est le résultat affiché suite à l'appel de f3(7):

- | | |
|--------------------------|------------------|
| A) 0 1 1 2 2 3 3 5 5 6 6 | B) 0 1 2 3 5 6 |
| C) 0 1 1 2 3 5 8 | D) 0 1 2 3 4 5 6 |

26. On considère la structure de données suivante :

```
typedef struct element{
    int *data;
    int size;
    struct element *next;
}Element;
```

La structure de données ci-dessus donne-t-il lieu à ?:

- | | |
|--|---|
| A) Une liste chaînée de tableaux devant avoir la même taille | B) Une liste chaînée de tableaux avec des tailles différentes |
| C) Une liste chaînée d'entiers | D) Erreur de compilation |

27. On considère le programme suivant :

```
int main() {
    int *T;
    int n=6;
    int i;
    T=(int*)malloc(sizeof(int)*(n+1));
    T[0]=T[1]=1;
    for(i=2;i<=n;i++){
        T[i]=T[i-1]+T[i-2];
    }
    printf("%d",T[n]);
    return 0;
}
```

Quel est le résultat affiché par ce programme :

- | | |
|-------|-------|
| A) 12 | B) 14 |
| C) 15 | D) 13 |

28. Soit la structure de données suivante :

```
typedef struct Element{
    int data;
    struct Element *next;
}Element;
Soit addData une procédure qui permet d'ajouter un entier à la fin de la liste.
Soit la procédure displayList définie par le code suivant :
void displayList(Element *p){
    while(p!=NULL){
        printf("%d ",p->data);
        p=p->next;
    }
}
```

Et on considère le programme suivant :

```
int main(){
    Element *L;L=NULL;
    addData(1,&L); addData(2,&L);
    addData(3,&L); addData(4,&L);
    displayList(L);
    return 0;
}
```

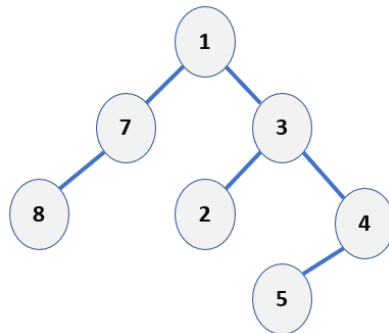
Quel est la proposition correcte après l'appel de **displayList(L)**:

- A) L est NULL
- B) Problème de mémoire
- C) Erreur d'exécution
- D) L n'est pas NULL

29. Soit la structure de données d'un arbre binaire suivante :

```
typedef struct element{
    int data;
    struct element *right;
    struct element *left;
}Arbre;
```

A l'aide de la structure précédente, nous avons créé l'arbre binaire T suivant :



- T représente la racine de l'arbre.
- Pour afficher le contenu de l'arbre, nous avons utilisé la procédure suivante :

```
void display(Element *T){
    if(T!=NULL){
        display(T->left);
        display(T->right);
        printf("%d ",T->data);
    }
}
```

Quel est la proposition correcte après l'appel de **display(T)**:

- A) 8 7 2 5 4 3 1 B) 8 7 1 2 3 5 4
- C) 1 7 8 3 2 4 5 D) Erreur d'exécution

30. Les deux programmes suivants calculent la suite de fibonnaci:

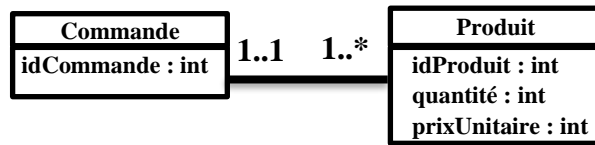
<pre> long program1(int n){ if(n<=1) return n; else return program1(n-1)+program1(n-2); } </pre>	<pre> long program2(int n) { int a,b,i; long c; a = 0; b = 1; if (n <= 1) return n; i = 1; while (i< n) { c = a + b; a = b; b = c; i = i + 1; } return c; } </pre>
---	--

Pour n=10,

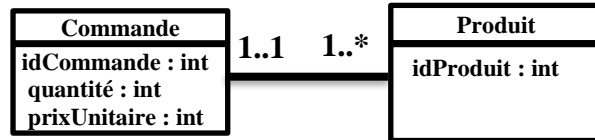
- A) L'espace mémoire utilisé par program1 est plus grand que celui utilisé par program2 B) L'espace mémoire utilisé par program2 est plus grand que celui utilisé par program1
- C) Les deux programmes utilisent le même espace mémoire D) Aucune des propositions n'est correcte

31. Quel diagramme de classes est correcte pour l'énoncé suivant : Une commande contient plusieurs produits avec une quantité et un prix unitaire. Un produit peut appartenir à plusieurs commandes :

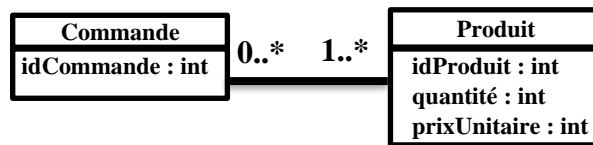
A. Modèle 1



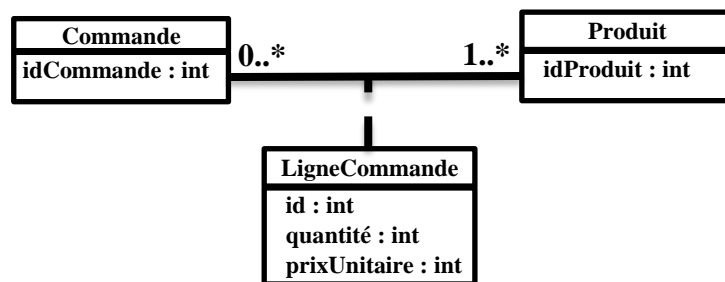
B. Modèle 2



C. Modèle 3

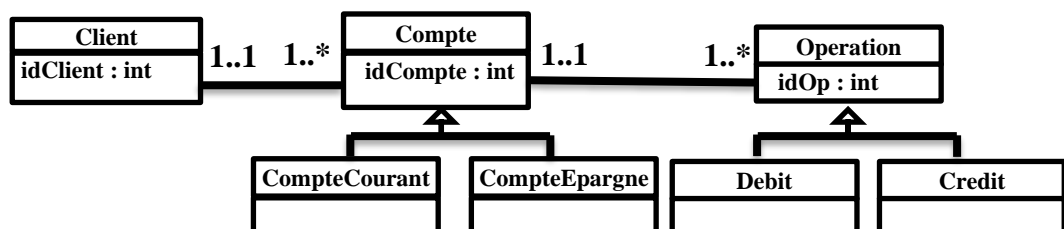


D. Modèle 4

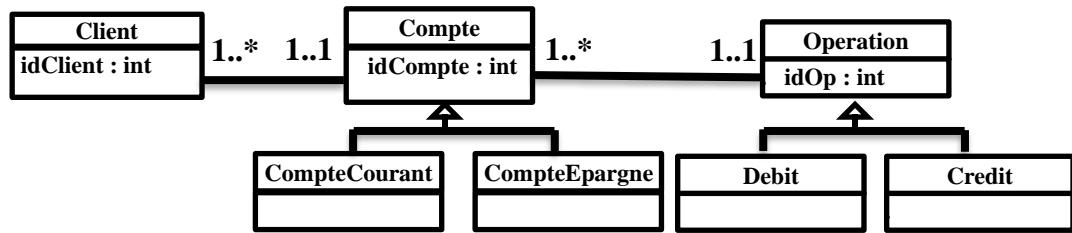


32. Quel diagramme de classes est correcte pour l'énoncé suivant : Un client peut avoir plusieurs comptes courants ou épargnes. Chaque compte peut subir des opérations de débit ou crédit

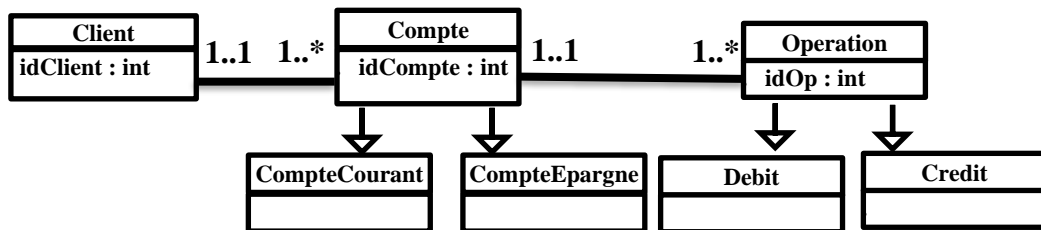
A. Modèle 1



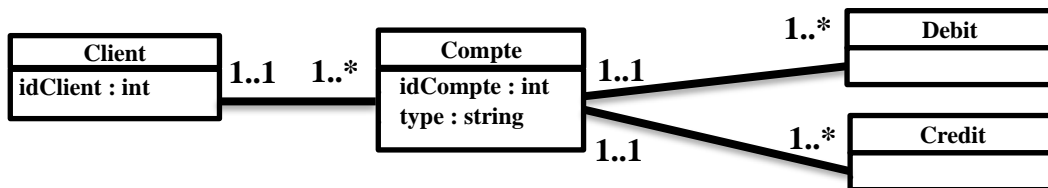
B. Modèle 2



C. Modèle 3



D. Modèle 4



33. On considère le modèle d'une base de données relationnelle qui permet de gérer les commandes d'un client. Cette base de données se compose des tables suivantes :

- CLIENT (ID_CLIENT, NOM_CLIENT)
- PRODUIT (ID_PRODUIT, NOM_PRODUIT, QUANTITE_STOCK)
- COMMANDE (ID_CMD, DATE_CMD, #ID_CLIENT)
- LIGNE_COMMANDE (ID_LC, #ID_COMMANDE, #ID_PRODUIT, QUANTITE, PRIX_UNITAIRE)

Quelle est la requête SQL qui permet de chercher les commandes du client dont le nom est « ALPHA »

- A. `SELECT CMD.ID_CMD, CMD.DATE_CMD, CLI.NOM_CLIENT FROM commande as CMD, client AS CLI WHERE (CLI.NOM_CLIENT='ALPHA')`
- B. `SELECT CMD.ID_CMD, CMD.DATE_CMD, CLI.NOM_CLIENT FROM commande as CMD, client AS CLI WHERE (CMD.ID_CLIENT=CLI.ID_CLIENT) and (CLI.NOM_CLIENT='ALPHA')`

- C. `SELECT CMD.ID_CMD, CMD.DATE_CMD, CLI.NOM_CLIENT FROM commande as CMD, client AS CLI WHERE (CLI.CLI.NOM_CLIENT='ALPHA')`
- D. `SEARCH CMD.ID_CMD, CMD.DATE_CMD, CLI.NOM_CLIENT FROM commande as CMD, client AS CLI WHERE (CLI.CLI.NOM_CLIENT='ALPHA')`

34. On considère le modèle d'une base de données relationnelle qui permet de gérer les commandes d'un client. Cette base de données se compose des tables suivantes :

- CLIENT (ID_CLIENT, NOM_CLIENT)
- PRODUIT (ID_PRODUIT, NOM_PRODUIT, QUANTITE_STOCK)
- COMMANDE (ID_COMMANDE, DATE_COMM, #ID_CLIENT)
- LIGNE_COMMANDE (ID_LC, #ID_COMMANDE, #ID_PRODUIT, QUANTITE, PRIX_UNITAIRE)

Quelle est la requête SQL qui permet de calculer le montant total de chaque commande. On ne retiendra que les commandes dont le total est supérieur à 9000

- A. `SELECT LC.ID_CMD, SUM(LC.QUANTITE*LC.PRIX_UNITAIRE) AS TOTAL FROM ligne_commande as LC GROUP BY LC.ID_CMD WHERE TOTAL>9000`
- B. `SELECT LC.ID_CMD, SUM(LC.QUANTITE*LC.PRIX_UNITAIRE) AS TOTAL FROM ligne_commande as LC GROUP BY LC.ID_CMD HAVING TOTAL>9000`
- C. `SELECT LC.ID_CMD, SUM(LC.QUANTITE*LC.PRIX_UNITAIRE) AS TOTAL FROM ligne_commande as LC WHERE TOTAL>9000`
- D. `SELECT LC.ID_CMD, SUM(LC.QUANTITE*LC.PRIX_UNITAIRE) AS TOTAL FROM ligne_commande as LC WHERE LC.TOTAL>9000`

35. Sachant que les codes respectifs des caractères ASCII « A », « B » et « C » en binaire sont 01000001, 01000010 et 01000011, donnez le code du caractère « D » en décimal.

- A) 67 B) 70
- C) 69 D) 68

36. Convertir en octal ce nombre hexadécimal $A1F_{16}$:

- A) 4037_8 B) 7037_8
C) 6037_8 D) 5037_8

37. Effectuer l'opération suivante en binaire $111_2 + 101_2$:

- A) 1000_2 B) 1100_2
C) 1110_2 D) 1111_2

38. Sachant que sur un réseau informatique une machine utilise cet adresse IP 192.168.0.160 avec comme masque de sous réseau 255.255.255.192. Calculer selon l'opération ET logique bit par bit l'adresse réseau au quelle appartient cette machine:

- A) 192.168.0.128 B) 192.168.0.138
C) 192.168.0.148 D) 192.168.0.158

39. Convertir en décimal ce nombre fractionnaire binaire $101,11_2$:

- A) $3,75_{10}$ B) $4,75_{10}$
C) $5,75_{10}$ D) $6,75_{10}$

40. Convertir en binaire selon le codage en complément à deux sur huit bits ce nombre signé décimal, -90_{10} :

- A) 10100110_2 B) 10100111_2
C) 10101111_2 D) 10111111_2