



Informatique Examen - Durée 1H30

Session normale - Automne 2018-2019

Documents non autorisés

Consignes importantes

- Une écriture lisible, la clarté des raisonnements ainsi que la qualité de la rédaction interviendront pour une part importante dans l'appréciation des copies,
- L'utilisation de tout appareil électronique est interdite.

Questions de cours (4.5 pts)

1. les affirmations suivantes sont-elles vraies ou fausses?

- a) Toute boucle **for** peut être réécrite en une boucle **while** équivalente. b) Une « chaîne de caractères » est un tableau de **char**.
c) L'expression **!(a || b)** est équivalente à **(!a && !b)**.

2. Soit les déclarations suivantes :

```
int n = 5, p = 9 ;
int q ; float x ;
```

Donner le type et la valeur de chacune des expressions suivantes :

- a) $q = p \% 3$; d) $x = p / n$;
b) $q = n == p$; e) $x = (\text{float}) p / n$;
c) $q = p \% n + (p > n)$; f) $q *= --n + (p/2)$

3. Quelles erreurs ont été commises dans les groupes d'instructions suivants :

a)

```
int a,b;
...
if(a<b)
printf("ascendant")
else printf("descendant") ;
```

b)

```
#define LIMITE 100
int n, var;
...
switch (var) {
case 1 : printf("petit");
case n : LIMITE++;
}
```

Exercice 1 : Résolution d'une équation du 2^{ème} degré (5 pts) 4

Écrire un algorithme de résolution de l'équation du second degré $ax^2+bx+c=0$ qui affiche les racines (solutions) en tenant compte des cas particuliers suivants : (a,b, et c sont saisies au clavier)

- Si $a = 0$, l'équation est ramenée à une équation du 1^{er} degré.
- Si $a = 0$ et $b = 0$, il y a une infinité de solutions, pour autant que $c = 0$. Si c n'est pas égal à 0, il y a impossibilité.
- Si le discriminant (delta) est inférieur à 0 : l'équation n'a pas de solutions réelles.
- Si le discriminant est égal à 0 : l'équation a une racine double ($x = -b/2a$).
- Si le discriminant est supérieur à 0 : l'équation a deux racines (solutions) réelles.

Exercice 2: Triangle (4.5 pts)

Écrire un programme en C demandant à l'utilisateur un entier n et affiche un triangle vide formé d'étoiles (*) d'une hauteur n.



Exercice 3 : Tableaux (6pts)

Soit la chaîne de caractère suivante : S = "Cette ligne contient quelques lettres e".

Écrire un programme en C qui permet de :

- 1- Déclarer les variables nécessaires et initialiser la chaîne de caractère S.
- 2- Compter le nombre d'occurrences (nombre d'apparition) de la lettre 'e' dans la chaîne S
- 3- Parcourir la chaîne et donner la position de la première occurrence de la lettre 'e'
- 4- Enlever toutes les apparitions de la lettre 'e' et donner le résultat en affichant de nouveau le tableau

CNE : *****

Durée : 1h 30

Selon Cas

☐ **Si... Sinon**

- ☒ `printf("%d", nbr);`
- ☐ `scanf("%d", &nbr);`
- ☒ `scanf("%d", nbr);`

```
i=3;
while (i<=3){
    printf("i = %d",i);
}
i++;
```

☒ 0 fois
☐ 4 fois
☐ à l'infini

```
int x=1, y=1, z=1,  
    y=++x;  
    z=y++ ;
```

- ☐ $x=1, y=1, z=1$;
- ☐ $x=2, y=3, z=2$;
- ☐ $x=2, y=2, z=3$;

```
1. int main {
2. int i,S=0;
3. for(i=1;i<=6;i++)
4.     S=S + i
5. printf("%d\n",S);
6. return 0;
7. }
```

☐ 1, 2, 5
☐ 1, 4, 5
☐ 2, 3, 4
☐ 2, 4, 5

☐ 1 2 3 4 5 6

☐ 15

☐ 21

1. Traduire l'algorithme suivant en langage C

1. Réécrire l'algorithme précédent en utilisant la boucle tant que
2. Donner les valeurs affichées par le programme si $N=12$
2 3 4 6
3. Que représentent ces valeurs? : les diviseurs de N
4. Ecrire un algorithme (ou un programme en langage C), qui permet de lire un entier n strictement supérieur à 2 et d'afficher son plus petit diviseur strictement supérieur à 1 ;

Exemple d'exécution :

→ Entrer un entier strictement supérieur à 2 : 12

→ Le plus petit diviseur (strictement supérieur à 1) de 12 est : 3

```
#include <stdio.h>
main() {
    int i, N;
    printf("Entrer un entier strictement supérieur à 2:");
    scanf("%d",&N);
    i=2;
    while (N%i!=0)
        i++;
    printf("le plus petit diviseur (strictement supérieur à 1 ) de %d est: %d\n",N,i);
}
```

Exercice 3 (7 points) :

Ecrire un programme en C, qui permet de :

- Saisir un tableau **tab** d'entiers de taille N=7 ;
- Afficher les éléments du Tableau **tab** ;
- Afficher le plus grand élément **g** de **tab** ;
- Afficher le nombre de fois que **g** est répété dans le tableau **tab** ;
- Afficher le plus petit et le plus grand des indices correspondant à **g**.

Exemple d'exécution :

indice	0	1	2	3	4	5	6
Tab =	12	25	9	25	17	25	15

- Les éléments de tab sont : 12 25 9 17 25 15
- Le plus grand élément est : 25
- Le nombre de fois que 25 est répété dans le tableau est : 3
- le plus petit indice correspondant à 25 est : 1
- le plus grand indice correspondant à 25 est : 5

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int tab[7], i, g, nbr_g, pp, pg;
    /*Saisie du tableau*/
    for(i=0; i<7; i++)
    {
        printf("Entrer l'element num %d du tableau:", i);
        scanf("%d", &tab[i]);
    }
    /*Affichage du tableau*/
    printf("Les éléments de tab sont:");
    for(i=0; i<7; i++)
        printf("\t %d ", tab[i]);
    /* Le plus grand élément du tab*/
    g=tab[0];
    for(i=1; i<7; i++)
        if (g<tab[i])
            g=tab[i];
    printf("\nle plus grand élément est: %d\n", g);
    /* Le nombre de fois que g est répété dans le tab*/
    nbr_g=0;
    for(i=0; i<7; i++)
        if (g==tab[i])
            nbr_g++;
```

```
printf("\n le nombre de fois que %d est répété dans le
tableau est : %d\n", g, nbr_g);
/* Le plus petit et le plus grand indice correspondant à g*/
g=tab[0];
pp=0;
pg=0;
for(i=1; i<7; i++)
{
    if (g<tab[i]) /* plus petit indice correspondant à g*/
    {
        g=tab[i];
        pp=i;
    }
    if (g<=tab[i]) /*Le plus grand indice correspondant à g*/
    {
        pg=i;
    }
}
printf("Le plus petit indice correspondant à %d
est: %d\n", g, pp);
printf("Le plus grand indice correspondant à %d
est: %d\n", g, pg);
system("PAUSE");
return 0;
}
```



Examen de l'informatique
Session normale- Automne 2017/2018

Durée : 1h 30

Exercice 1 (4 points) :

1. Qu'affiche le programme C suivant :

```
#include <stdio.h>
#include <stdlib.h>
void main(){
    int x=1, y=1, z=1, t=1;
    char mot[10]="FPT";
    y=++x;
    z=t++;
    printf("x=%d, y=%d, z=%d\n",x, y, z);
    z+=y--x;
    printf("x=%d, y=%d, z=%d\n",x, y, z);
    x=strlen(mot);
    y++;
    printf("x=%d, y=%d\n", x, y);
    system("PAUSE");
}
```

x=2, y=2, z=1

x=1, y=1, z=2

x=3, y=2

Exercice 2 (4,5 points)

Ecrire un programme en langage C qui demande successivement 10 nombres positifs à l'utilisateur, et qui lui dise ensuite quel était le plus grand et le plus petit parmi ces 10 nombres :

Exemple :

Entrez le nombre numéro 1 : 13
Entrez le nombre numéro 2 : 34
:
Entrez le nombre numéro 10 : 6

Le plus grand de ces nombres est : 34

Le plus petit de ces nombres est : 6

```
#include <stdio.h>
#include <math.h>
void main() {
    int nb, i, max, min;
    printf("Entrez le nombre numéro 1 : ");
    scanf("%d",&nb);
```

```
max=min=nb;
for(i=2;i<=10;i++)
{
    printf("Entrez le nombre numéro %d : ",i);
    scanf("%d",&nb);
    if(nb>max)
        max=nb;
    if(nb<min)
        min=nb;
}
printf("Le plus grand de ces nombres est %d\n",max);
printf("Le plus petit de ces nombres est %d\n",min);
}
```

Exercice 3 (7 points) :

1. Traduire l'algorithme suivant en langage C

Algorithme	Programme en langage C	Algorithme avec Tant que
Algorithme sommel Variables S, i : entier Debut S ← 0 i ← 0 Répéter S ← S + i i ← i + 1 Jusqu'à (i = 10) Ecrire (" S = ", S) Fin	<pre>#include <stdio.h> void main() { int S, i; S=0; i=0; do{ S+=i; i++; }while(i!=10); printf("S= %d\n",S); }</pre>	Algorithme sommel Variables S, i : entier Debut S ← 0 i ← 0 Tant que (i!=10) S ← S + i i ← i + 1 Fin tant que Ecrire (" S = ", S) Fin

2. Est-ce qu'on peut réécrire l'algorithme précédent en utilisant la boucle pour ?, justifier votre réponse
Oui, car le nombre de répétitions est connu.
3. Réécrire l'algorithme précédent en utilisant la boucle Tant que
4. Donner la valeur affichée par le programme est 45. Que représente cette valeur ?
La somme des nombres de 1 à 9

Exercice 4 (4,5 points) :

Ecrire un algorithme (ou un programme), qui permet de :

1. Saisir deux tableaux d'entiers **T1** (de taille N=3) et **T2** (de taille N=7) ;
2. Afficher les éléments du Tableau **T1**, puis les éléments du Tableau **T2** ;
3. Afficher le message :
 - « **T1** est inclus dans **T2** » → si tout élément de **T1** est aussi élément de **T2**
 - « **T1** n'est pas inclus dans **T2** » → s'il existe un élément n de **T1**, tel que n n'appartient pas à **T2**

• Exemple :

➤ **T1** =

3	2	5
---	---	---

T2 =

0	16	5	7	2	4	8
---	----	---	---	---	---	---

 → **T1** n'est pas inclus dans **T2**

➤ **T1** =

1	4	7
---	---	---

T2 =

9	1	6	7	16	4	8
---	---	---	---	----	---	---

 → **T1** est inclus dans **T2**

```
#include <stdio.h>
#include <math.h>
void main() {
  int T1[3], T2[7];
  int i,j, inc=0;
  /* Saisie du tableau T1*/
  for(i=0;i<3;i++)
  {
    printf("donner l'element num %d de T1:",i);
    scanf("%d",&T1[i]);
  }
  /* Saisie du tableau T2*/
  for(i=0;i<7;i++)
  {
    printf("donner l'element num %d de T2:",i);
    scanf("%d",&T2[i]);
  }
  /*Affichage du tableau T1*/
  for(i=0;i<3;i++)
    printf("%d \t",T1[i]);

  printf("\n");
  /*Affichage du tableau T2*/
  for(i=0;i<7;i++)
    printf("%d \t",T2[i]);

  /*Affichage du message*/
```

```
}
```



Correction d'Examen de l'informatique
Session normale / Automne 2015/2016

Exercice 1 : (5 points)

Qu'affiche le programme C suivant :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int x=0, y=0, T[]={0,0,0,0};
    char mot[10]="SABE3";
    T[0]=++x;
    printf("x=%d, T[0]=%d\n", x, T[0]);
    T[1]=x++;
    printf("x=%d, T[1]=%d\n", x, T[1]);
    T[2]+=T[1]-=x;
    printf("T[1]=%d, T[2]=%d\n", T[1], T[2]);
    T[++y]=5;
    printf("y=%d, T[0]=%d, T[1]=%d\n", y, T[0], T[1]);
    x=strlen(mot);
    strrev(mot);
    printf("x=%d, %s, %s\n", x, strlwr(mot), strrev(mot));
    system("PAUSE");
    return 0;
}
```

Le programme affiche:

```
x=1, T[0]=1
x=2, T[1]=1
T[1]=-1, T[2]=-1
y=1, T[0]=1, T[1]=5
x=5, sabe3, sabe3
```

Exercice 2 : (10,5 points) Soit l'algorithme suivant :

Algorithme	Algorithme en utilisant la boucle pour (question 3)
Algorithme Calcul Variables S, N, i : Entier Debut i ← 1 S ← 0 Ecrire ("donner la valeur de N") Lire (N) Tant que (i ≤ N) S ← S + i * i i ← i + 1 Fin Tantque Ecrire (" la valeur de S est :", S) Fin	Algorithme Calcul Variables S, N, i : Entier Debut S ← 0 Ecrire ("donner la valeur de N") Lire (N) Pour i allant de 1 à N S ← S + i * i Fin pour Ecrire (" la valeur de S est :", S) Fin

- 1) Faire le déroulement (l'exécution) de l'algorithme donné pour N=3 et déduire la valeur finale de S.
 i= 1 -----> S= 0
 i= 2 -----> S= 1
 i= 3 -----> S= 5
 i= 4 -----> S= 14
 La valeur finale de S est 14
- 2) Déduire l'expression générale calculée par l'algorithme en fonction de N.

$$S = \sum_{i=1}^N (i^2)$$

- 3) Réécrire l'algorithme en remplaçant la boucle Tant que par la boucle Pour.
- 4) Traduire l'algorithme donné en programme C.
- 5) Réécrire le programme en remplaçant la boucle Tant que par la boucle Do while

Programme en langage C (question 4)	Programme en utilisant la boucle Do while (question 5)
<pre>#include <stdio.h> #include <stdlib.h> int main(){ int S=0, N, i=1; printf("donner la valeur de N:"); scanf("%d",&N); while (i<=N) { S+=i*i; i++; } printf("la valeur de S est %d\n",S); system("PAUSE"); return 0; }</pre>	<pre>#include <stdio.h> #include <stdlib.h> int main(){ int S=0, N, i=1; printf("donner la valeur de N:"); scanf("%d",&N); do { S+=i*i; i++; } while (i<=N); printf("la valeur de S est %d\n",S); system("PAUSE"); return 0; }</pre>

Exercice 3 : (4,5 points)

1) Les habitants d'une ville paient l'impôt selon les règles suivantes :

- les hommes de plus de 20 ans **paient l'impôt** ;
- les femmes entre 18 et 35 ans **paient l'impôt** ;
- les autres **ne paient pas d'impôt**.

Ecrire un algorithme qui demande l'âge et le sexe de l'habitant, et affiche **s'il paye l'impôt ou non**.

Algorithme Impot

Variables age: Entier
Sexe : caractère

Debut

Ecrire ("donner l'âge")
Lire (age)
Ecrire ("donner le sexe")
Lire (sexe)

If (sexe='H') alors

If (age>20) alors

Ecrire ("il paye l'impôt ")

Else

Ecrire ("il ne paye pas l'impôt ")

End if

Else

If (age>=18) et (age<=35) alors

Ecrire ("il paye l'impôt ")

Else

Ecrire ("il ne paye pas l'impôt ")

End if

End if

Fin

2) Ecrire un programme en langage C qui affiche, selon un numéro entre 1 et 12, le mois de l'année correspondant

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    int nombre;
    printf("Entrez un nombre:");
    scanf("%d",&nombre);
    switch (nombre) {
        case 1: printf(" Janvier ");
            break;
        case 2: printf("Fevrier");
            break;
        case 3: printf("Mars");
            break;
        case 4: printf("Avril");
            break;
        case 5: printf("Mai");
            break;
        case 6: printf("Juin");
```

```
        break;
        case 7: printf("Juillet");
            break;
        case 8: printf("aout");
            break;
        case 9: printf("Septembre");
            break;
        case 10: printf("Octobre");
            break;
        case 11: printf("Novembre");
            break;
        case 12: printf("decembre");
            break;
        default: printf("le nombre doit être entre 1 et 12");
    }
    system("PAUSE");
    return 0;
}
```

Examen de l'informatique - Durée : 1h 30
Session de Rattrapage / Automne2014/2015

Programme non corrigé	Programme corrigé
#include <stdio.h>	
#include <stdlib.h>	
void main(){	
int A, B, D, R; /* déclaration des variables	int A, B, D, R; /* déclaration des variables*/
printf("donner A et B:");	
scanf("%d%d",A,B);	
D=A/B	scanf("%d%d",&A,&B);
R=A%B;	D=A/B ;
if B!=0	
printf("le quotient est: d\n",D);	if (B!=0){
printf("le reste est: d",R);	printf("le quotient est: %d\n",D);
else	printf("le reste est: %d\n",R);
printf(Resultat impossible\n);	}else
system("PAUSE");	printf("Resultat impossible\n");
}	

- Ce programme contient des erreurs; Corrigez les instructions qui représentent des erreurs dans la colonne "programme corrigé".
- Après la correction et l'exécution du programme, donner le résultat qui sera affiché par ce programme si A=10 et B=3:

Exercice 3 :

Ecrire un algorithme qui demande à l'utilisateur de saisir un nombre entier de départ, et qui calcule et affiche la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer: $1+2+3+4+5 = 15$

Exercice 4 : Soit l'algorithme suivant

- 1) Traduire cet algorithme en langage C.

Algorithme	Programme en langage C
Algorithme CALCUL; Variables N, I, Q, S: entier; DEBUT Lire(N); $I \leftarrow N$; $S \leftarrow 0$; TantQue $I \neq 0$ faire $Q \leftarrow I \% 10$; $S \leftarrow S * 10 + Q$; $I \leftarrow I / 10$; FinTantque ; Ecrire (S) FIN	

- 2) Dérouler cet algorithme pour $N = 52$ et puis pour $N = 123$ et donner la valeur de S pour chacune des 2 valeurs de N.

N = 52			N = 123		
Q	S	I	Q	S	I
Pour N= 52 $\rightarrow S = \dots\dots$			Pour N= 123 $\rightarrow S = \dots\dots$		

Que fait cet algorithme ? :

Examen de l'informatique - Durée : 1h 30
Session de Rattrapage - Printemps 2014

Fin

Soit **T** un tableau qui contient les notes de 20 étudiants. Ecrire un Algorithme qui permet de :

1. Saisir toutes les notes du tableau.
2. Afficher les notes du tableau.
3. Calculer et afficher la moyenne des notes du tableau.
4. Compter et afficher le nombre d'étudiants ayant une note supérieure à 10.
5. Déterminer et afficher la note minimale et la note maximale du tableau T.



TRCS3

Module 15 : Informatique

Nom et prénom :

CNE :

Filière :

CIN :

Examen de l'informatique - Durée : 1h 30
Session normale / Automne 2014/2015

Exercice 1 : (6 points)

Choisissez la bonne réponse (**une seule**) pour chaque question parmi la liste des réponses proposées

Q1 : À quoi sert un langage de programmation ?

- ☐ À écrire un document texte.
- ☐ À produire des organigrammes corrects.
- ☐ À traduire un algorithme de manière compréhensible par un ordinateur.

Q2 : Que permet de faire une boucle ? (Une boucle « **pour** » par exemple)

- ☐ Attribuer une valeur à une variable.
- ☐ Améliorer la vitesse d'exécution d'un programme mais augmente le risque d'erreurs.
- ☐ Initialiser les constantes.
- ☒ Répéter certaines actions plusieurs fois.

Q3 : Sachant qu'un traitement peut être répété **1 à n fois**, quelle est la meilleure structure pour ce traitement ?

- ☒ Répéter.... Jusqu'à
- ☐ Si... Sinon
- ☐ Tant que
- ☐ Selon Cas

Q4 : Peut-on insérer une boucle dans une autre boucle ?

- ☒ Oui.
- ☐ Non, jamais.

Q5 : Laquelle des instructions suivantes est juste ?

- ☐ scanf("%d", nbr);
- ☒ scanf("%d", &nbr);
- ☐ scanf(%d, &nbr);
- ☐ scanf(nbr);

Q6 : Soit la boucle suivante :

```
i=3 ;
while (i>3){
    printf("la valeur de i est :%d",i) ;
    i++;}
```

Le corps de cette boucle s'exécute :

- ☒ 0 fois
- ☐ 4 fois
- ☐ Erreur de compilation
- ☐ à l'infini

Exercice 2 : (4,5 points)

1) Sur quelles bases nous choisissons l'utilisation de la boucle **POUR** ou de la boucle **TANT QUE** ? (précisez la différence entre l'utilisation des deux boucles)

POUR	TANT QUE
.....	
.....	
.....	
.....	

2) Ecrire un algorithme qui demande à l'utilisateur de saisir une température de l'eau, et qui ensuite affiche son état suivant sa température :

Température <=0 " c'est de la glace "

0 < Température <=100 " c'est du liquide "

Température >100 " c'est de la vapeur "

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....