

Cet examen contient 4 pages.

Les Documents, calculatrices, téléphone portable sont interdits. Veuillez rendre une copie propre et claire. La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.

► **Exercice 1. Appel fonction et procédure (5 Points)**

Soit le programme suivant :

Algorithme Essai

```
r,s, c : entier;  
Début  
r ← 5;  
s ← 4;  
Un(r,s);  
afficher("r= ", r );  
afficher("s= ", s );  
c ← Trois(r);  
afficher("r= ", r );  
afficher("c= ", c );  
Fin
```

Procédure Deux(d/r A :entier)

```
Début  
repeter  
    A ← A + 1;  
jusqu'à (A ≥ 10);  
afficher(A);  
Fin
```

Procédure Un(a :entier, d/r b :entier)

```
i :entier;  
Début  
pour i ← 1 à 4 Faire  
    a ← a+i;  
    afficher(a);  
FinPour  
b ← i;  
Deux(a);  
afficher("a= ", a);  
Fin
```

Fonction Trois(C :entier) :entier

```
i :entier;  
Début  
pour i ← 4 à C Faire  
    afficher(i);  
FinPour  
Retourne(i);  
Fin
```

1. Exécutez le et donnez la séquence des affichages produits
2. Traduire la **fonction Trois()** en langage C



6 (0.25pt)
8 (0.25pt)
11 (0.25pt)
15 (0.25pt)
16 (0.25pt)
a= 16 (0.25pt)

r= 5 (0.25pt)
s= 5 (0.25pt)
4 (0.25pt)
5 (0.25pt)
r= 5 c= 6 (0.5pt)

```
int Trois(int C){ //(0,5pt)  
int i; //(0,25pt)  
for(i= 4 ; i<=C;i++) //(0,5pt)  
printf("%d\n",i); //(0,25pt)  
return(i) ; //(0,5pt)  
}
```



► Exercice 2. Langage C (7 Points)

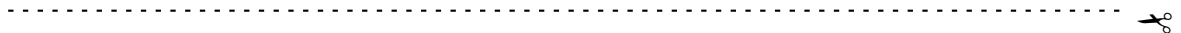
1. Écrire une fonction en algorithmique **Fonction Nbchiffres(N : entier) :entier** et en langage C **int Nbchiffres(long N)** qui prend un paramètre de type long et qui retourne le nombre de chiffres de N comme résultat.

Exemple :l'appel de Nbchiffres(236457) retournerait 6



Fonction Nbchiffres(N : entier) :entier
 cpt :entier;
Début
 cpt←0;
Tant que N > 0 **Faire**
 N←N div 10;
 cpt←cpt+1;
FinTQ
 Retourne(cpt);
Fin

```
int Nbchiffres(long N){
int cpt=0; //(0,25pt)
while(N!=0) //(0,25pt)
{ N=N/10; //(0,25pt)
  cpt++; //(0,25pt)
}
return cpt; //(0,5pt)
}
```

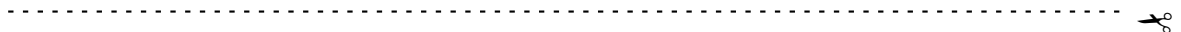


2. Qu'affiche le code C suivant :

```
char s [] = "EXAMEN";
int d=0;
while(s[d++]!=0){
if((d%2)==0) printf("%c",s[d]);
}
```



AE (1pt)

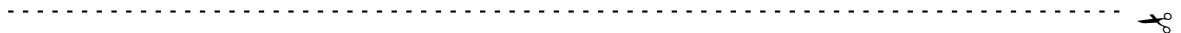


3. Qu'affiche le code C suivant :

```
int x = 5;
int y = x++ + 2;
x+=--y;
printf ("%d %d", x, y) ;
```



12 6(1pt)



4. Qu'affiche le code C suivant :

```
int a = 10, b=6, c=2;
c += (a>0 && a<=10) ? b++ : a/b;
printf ("%d %d %d",a,b,c);
```

✂
 10 7 8 (2pt)
 ✂

► **Exercice 3. Les tableaux (8 Points)** On désire écrire un algorithme qui construit un tableau d'entiers et qui permet ensuite de faire certains traitements sur le tableau. L'algorithme, commenté par la suite, est le suivant :

Algorithme tableau

constantes

max :entier=100;

variables

tab [max] : entier;

nbVal : entier;

Début

init (tab, nbVal);

afficher ("La somme des elements de tab= ", **sommeBip**(tab, nbVal));

Fin

1. Définir une procédure en algorithmique **init()** qui demande à l'utilisateur de saisir la taille du tableau, entier n positif ou nul (cette valeur ne doit pas aussi dépasser la taille maximum du tableau, redemander si nécessaire), et qui ensuite saisit les n entiers (**2 points**).

✂
Procédure **init**(_{d/r} t[] : entier, _{d/r} n : entier) (0,5pt)
 i : entier; (0,25pt)
Début
Répéter
 Afficher("Entrer la taille du tableau : ");
 Lire(n); (0,25pt)
Jusqu'à (n>=0) et (n<=max); (0,5pt)
 Afficher("Donner les éléments du tableau : ");
Pour i ← 0 à n-1 (0,25pt)
 Lire(t[i]); (0,25pt)
FPour; Fin
 ✂

2. Définir une fonction en algorithmique de **somme bipolaire** (en partant des deux extrémités) nommée **sommeBip()** qui reçoit un tableau de n entiers et sa taille en paramètre, et qui calcule et renvoie la somme (**3 points**).

✂
Fonction **sommeBip**(t[] : entier, n : entier) : entier (0,5pt)
 i, j, s : entier; (0,25pt)
Début
 s ← 0; (0,25pt)
 i ← 0; j ← n-1; (0,25pt)
Tant que (i < j) **faire** (0,25pt)
 s ← t[i] + t[j]; (0,25pt)
 i ← i+1; (0,25pt)
 j ← j-1; (0,25pt)
FinTQ

```
si (n mod 2 <> 0) alors (0,25pt)
retourne s+t[j];(0,25pt)
sinon
retourne s;(0,25pt)
finSI
Fin
```

..... ✂

3. Traduire la fonction **sommeBip()** en langage C (**3 points**).

✂

```
int sommeBip( int t[] , int n) {(// (0,25pt)
int i=0,j=n-1,s=0;  // (0,25pt)
while(i<j){ // (0,25pt)
s=s+t[i]+t[j]; //(0,25pt)
i++; //(0,5pt)
j--; //(0,5pt)
}
if(n%2!=0) //(0,5pt)
    return (s+t[j]); //(0,25pt)
else
    return s; //(0,25pt)
}
```

..... ✂

Bon travail