



Examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en C
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

1 Affichage

6 pts. ☹️25'

Qu'affiche les deux programmes suivants :

```
1 #include<stdio.h>
2 void main()
3 {
4     int n=10, s=0, p=1, i ;
5     if (s+p)
6         printf("s=%d, p=%d\n",s||p, s&&p);
7     for (i=1; i<=n; i++)
8     {
9         if (i%2 == 0)
10            s = s + i ;
11        else
12            p = p - i;
13    }
14    printf("s = %d, p=%d\n",s, p) ;
15    for (i=1, s=0, p=1; i<=n; i++)
16        s = s + 1;
17        p = p + 1;
18    printf("s = %d, p=%d\n",s, p) ;
19 }
```

```
1 #include<stdio.h>
2 int i=2,j=3;
3 void toto(int a){
4     a++;
5     printf("a = %d\n", a);
6 }
7 int loulou(int a){
8     j = a*5;
9     printf("j = %d\n", j);
10    a++;
11    return j-i;
12 }
13 void main(){
14     int x = 1;
15     toto(x);
16     printf("x = %d, j = %d\n", x, j) ;
17     j=loulou(i);
18     printf("i = %d, j = %d\n", i, j);
19 }
```

2 Tour de magie

8pts. ☹️35'

On souhaite écrire un programme qui permet de jouer un tour de magie basé sur des mathématiques. Ce tour de magie demande à une personne de penser à un nombre entier puis lui demande d'exécuter les calculs suivants :

1. Multiplier le nombre choisi par 4
2. Ajouter 6 au résultat
3. Diviser ce résultat par 2
4. Et soustraire ce résultat par le nombre choisi au départ (*càd*, retrancher le nombre de départ de ce résultat).

Enfin, la personne annonce le résultat obtenu. Le magicien devine ensuite, à partir du résultat annoncé, le nombre de départ choisi .

- a. Écrire une fonction Calcul qui prend en entrée un entier (le nombre choisi) puis exécute les calculs de ce tour de magie et renvoie à la fin le résultat obtenu.
- b. Écrire une fonction Magique qui prend en entrée un entier (qui représente le résultat obtenu) et devine puis renvoie le nombre qui avait été choisi au départ.
- c. Écrire une fonction main qui :

- demande à l'utilisateur de saisir un nombre entier
- appelle la fonction Calcul,
- affiche le résultat obtenu,
- appelle la fonction Magique,
- affiche le nombre de départ deviné.

d. Montrer que si la première instruction du calcul multipliait le nombre choisi au départ par 2 au lieu de 4, le résultat obtenu serait toujours égale à 3 quelque soit le nombre choisi au départ.

3 Multiplication égyptienne

6pts. ☹️30'

La multiplication égyptienne est une technique de multiplication de deux nombres basée sur le principe d'une multiplication par 2. Une variante de cette technique consiste à :

- diviser par 2 le multiplicateur (le premier nombre) ainsi que les quotients obtenus à chaque fois, jusqu'à un quotient nul,
- et à multiplier parallèlement le multiplicande (le deuxième nombre) par 2.
- on additionne alors les multiples obtenus du multiplicande correspondant aux quotients impairs.

Exemples :

85*257 =?		
x	y	
85	257	+ 257
42	514	
21	1028	+ 1028
10	2056	
5	4112	+ 4112
2	8224	
1	16448	+ 16448
0		
		= 21845

50*65 =?		
x	y	
50	65	
25	130	+130
12	260	
6	520	
3	1040	+1040
1	2080	+2080
0		
		= 3250

- Écrire une fonction Multiplication_Egyp qui prend en entrée deux entiers et qui calcule et renvoie leur produit en utilisant la technique de multiplication égyptienne.
- Écrire la fonction main qui permet de tester la fonction Multiplication_Egyp sur deux entiers saisis par l'utilisateur.



Correction de l'examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en C
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

1 Affichage

6 pts. ☹25'

Qu'affiche les deux programmes suivants :

```
1 #include<stdio.h>
2 void main()
3 {
4     int n=10, s=0, p=1, i ;
5     if (s+p)
6         printf("s=%d, p=%d\n",s||p, s&& p);
7     for (i=1; i<=n; i++)
8     {
9         if (i%2 == 0)
10            s = s + i ;
11        else
12            p = p - i;
13    }
14    printf("s = %d, p=%d\n",s, p) ;
15    for (i=1, s=0, p=1; i<=n; i++)
16        s = s + 1;
17        p = p + 1;
18    printf("s = %d, p=%d\n",s, p) ;
19 }
```

```
1 #include<stdio.h>
2 int i=2,j=3;
3 void toto(int a){
4     a++;
5     printf("a = %d\n", a);
6 }
7 int loulou(int a){
8     j = a*5;
9     printf("j = %d\n", j);
10    a++;
11    return j-i;
12 }
13 void main(){
14     int x = 1;
15     toto(x);
16     printf("x = %d, j = %d\n", x, j) ;
17     j=loulou(i);
18     printf("i = %d, j = %d\n", i, j);
19 }
```

Solution

Affichage

s=1, p=0
s = 30, p=-24
s = 10, p=2

Affichage

a = 2
x = 1, j = 3
j = 10
i = 2, j = 8

2 Tour de magie

8pts. ☹35'

On souhaite écrire un programme qui permet de jouer un tour de magie basé sur des mathématiques. Ce tour de magie demande à une personne de penser à un nombre entier puis lui demande d'exécuter les calculs suivants :

1. Multiplier le nombre choisi par 4
2. Ajouter 6 au résultat
3. Diviser ce résultat par 2
4. Et soustraire ce résultat par le nombre choisi au départ (càd, retrancher le nombre de départ de ce résultat).

Enfin, la personne annonce le résultat obtenu. Le magicien devine ensuite, à partir du résultat annoncé, le nombre de départ choisi.

- Écrire une fonction `Calcul` qui prend en entrée un entier (le nombre choisi) puis exécute les calculs de ce tour de magie et renvoie à la fin le résultat obtenu.
- Écrire une fonction `Magique` qui prend en entrée un entier (qui représente le résultat obtenu) et devine puis renvoie le nombre qui avait été choisi au départ.
- Écrire une fonction `main` qui :
 - demande à l'utilisateur de saisir un nombre entier
 - appelle la fonction `Calcul`,
 - affiche le résultat obtenu,
 - appelle la fonction `Magique`,
 - affiche le nombre de départ deviné.
- Montrer que si la première instruction du calcul multipliait le nombre choisi au départ par 2 au lieu de 4, le résultat obtenu serait toujours égale à 3 quelque soit le nombre choisi au départ.

Solution

```
1 #include<stdio.h>
2 int calcul(int x)
3 {
4     return (x*4+6)/2-x;
5 }
6 int magique (int r)
7 {
8     return r-3 ;
9 }
10 void main()
11 {
12     int x=0, r=0;
13     printf("Donnez un nombre entier : "); scanf("%d", &x);
14     r = calcul(x);
15     printf("Le resultat est : %d\n", r);
16     x = magique(r);
17     printf("Le nombre de depart est donc : %d\n", x);
18 }
```

La réponse à la dernière question peut être prouvée mathématiquement comme suit :

$$\begin{aligned}\text{Calcul}(x) &= r \\ &= \frac{x \times 2 + 6}{2} - x \\ &= \frac{x \times 2}{2} + \frac{6}{2} - x \\ &= x + 3 - x \\ &= 3\end{aligned}$$

3 Multiplication égyptienne

6pts. ☹️30'

La multiplication égyptienne est une technique de multiplication de deux nombres basée sur le principe d'une multiplication par 2. Une variante de cette technique consiste à :

- diviser par 2 le multiplicateur (le premier nombre) ainsi que les quotients obtenus à chaque fois, jusqu'à un quotient nul,
- et à multiplier parallèlement le multiplicande (le deuxième nombre) par 2.
- on additionne alors les multiples obtenus du multiplicande correspondant aux quotients impairs.

Exemples :

85*257 =?		
x	y	
85	257	+ 257
42	514	
21	1028	+ 1028
10	2056	
5	4112	+ 4112
2	8224	
1	16448	+ 16448
0		
		= 21845

50*65 =?		
x	y	
50	65	
25	130	+130
12	260	
6	520	
3	1040	+1040
1	2080	+2080
0		
		= 3250

- Écrire une fonction `Multiplication_Egyp` qui prend en entrée deux entiers et qui calcule et renvoie leur produit en utilisant la technique de multiplication égyptienne.
- Écrire la fonction `main` qui permet de tester la fonction `Multiplication_Egyp` sur deux entiers saisis par l'utilisateur.

Solution

```

1 #include<stdio.h>
2 int Multiplication(int x, int y)
3 {
4     int r = 0;
5     while (x != 0)
6     {
7         if (x % 2)
8             r = r + y;
9         x = x / 2;
10        y = y * 2;
11    }
12    return r;
13 }
14 void main()
15 {
16     int x,y;
17     printf("Donner x :"); scanf("%d", &x);
18     printf("Donner y :"); scanf("%d", &y);
19     printf("x*y = %d", Multiplication(x,y));
20 }

```