

CI

Gilles Maire <gilles@gillesmaire.com>

Corrigé des exercices

30/10/2018

Contents

Exercice 1 : Premier exercice	2
Exercice 2 : Portée des variables	2
Exercice 3 : Exercice opérateur	3
Exercice 4 : Taille mémoire	3
Exercice 5 : Codes ASCII	4
Exercice 6 : Caractères ASCII	4
Exercice 7 : Nombre d'or	5
Exercice 8 : Tableaux	6
Exercice 9 : Exercices fonctions	6
Exercice 10 : Message d'erreur	8

Exercice 1 : Premier exercice

Énoncé

- Reprendre le thème du premier exemple, c'est à dire écrire un programme qui affiche une chaîne de caractères à l'écran
- Le compiler et l'exécuter.

Solution

```
#include <stdio.h>

void main()
{
    printf ("premier exercice\n");
    return;
}
```

Exercice 2 : Portée des variables

Énoncé

- Sachant que le programme printf que nous n'avons pas encore vu permet d'afficher un entier i via la syntaxe : `printf ("La valeur de i est %d\n",i)`
- Que l'on déclare une valeur entière qui vaut la valeur 2 par la syntaxe `int i=2;`
- Déclarer dans votre programme main.c plusieurs blocs imbriqués, dans chacun des blocs déclarer une variable i avec une valeur différente, l'afficher.
- Quelles sont les conclusions ?## Solution

```
#include <stdio.h>

void main()
{
    int i =1 ;
    printf ("i = %d\n",i);
    {
        int i=2;
        printf ("i = %d\n",i);
    }
    printf ("i = %d\n",i);
}
```

```
return ;  
}
```

Exercice 3 : Exercice opérateur

Énoncé

- Peut-on faire ceci ?

```
int i = 3.7;
```

- Que se passe-t-il si on fait cela juste après ?

```
int i= 3.7+7.3 ;
```

- Que donne ?

```
i=3.7+7.3;
```

Solution

```
#include <stdio.h>  
  
void main()  
{  
  int i ;  
  i =3.7;  
  printf ("i = %d\n",i);  
  i = 3.7+7.3;  
  printf ("i = %d\n",i);  
  return ;  
}
```

Exercice 4 : Taille mémoire

Énoncé

- Afficher la taille mémoire prise par un entier, par une adresse sur un entier, par un long unsigned.## Solution

```
#include <stdio.h>

void main()
{
    int i ;
    long j;
    i =3.7;
    printf ("Taille entier %d\n",sizeof(i));
    printf ("Taille long %ld\n",sizeof(j));
    printf ("Taille adresse %ld\n",sizeof(&j));
    printf ("Taille adresse %d\n",sizeof(&i));

    return ;
}
```

Exercice 5 : Codes ASCII

Énoncé

- Afficher le code ASCII des caractères de a à z
- Afficher ensuite les caractères ASCII A à z ## Solution

```
#include <stdio.h>

void main()
{
    char i;
    for ( i='A'; i<='z'; i++)
    {
        printf ("%d\n",(int)i);
    }
    return ;
}
```

Exercice 6 : Caractères ASCII

Énoncé

- Afficher les caractères ASCII de A à z en affichant cette fois leur valeur ASCII et leur valeur caractère.

Solution

```
#include <stdio.h>

void main()
{
    char i;
    for ( i='A'; i<='z'; i++)
    {
        printf ("Caractère %c (%d)\n",i,(int)i);
    }
    return ;
}
```

Exercice 7 : Nombre d'or

Énoncé

- Définir un entier entre 0 et 100 appelé nombre d'or
- Demander à un joueur d'entrer un nombre et à chaque réponse indiquer si le nombre d'or est plus petit ou plus grand
- Une fois que le nombre a été trouvé afficher en combien de passes le jeu a été gagné
- Ceux qui auront du temps pourront se documenter sur la fonction rand pour produire le nombre d'or de façon aléatoire. ## Solution

```
#include <stdio.h>

void main()
{
    const int NombredOr=78;
    int point=0;
    int i;
    do
    {
        printf( "Entrez un nombre:");
        scanf("%d",&i);
        if ( i > NombredOr )
            printf ("Le nombre d'or est inferieur\n");
        else if ( i < NombredOr )
            printf ("Le nombre d'or est superieur\n");
        else
            printf("Bravo vous avez gagne avec %d points!\n",point);
    }
```

```

    point ++;
}
while (i!=NombredOr);
return;
}

```

Exercice 8 : Tableaux

Énoncé

- Afficher un tableau qui comprend 1 au premier rang et le double au rang suivant. 1, 2, 4, 8 etc..
- On limitera l'exercice au 30 premiers entiers. Que se passe-t-il si on va jusqu'aux 40 premiers entiers ?## Solution

```

#include <stdio.h>

void main()
{
    int i;
    int tab[30];
    for (i=0 ;i<30;i++)
    {
        int res;
        if ( i == 0 ) res=1;
        else res=2*tab[i-1] ;
        tab[i]=res;
        printf ("tab[%d]=%d\n",i,res);
    }
}

```

Exercice 9 : Exercices fonctions

Énoncé

- Écrire la fonction max qui renvoie le maximum de deux nombres,
- Écrire la fonction min qui renvoie le minimum de deux nombres
- Écrire la fonction ascii qui renvoie le code ASCII d'un caractère
- Écrire la fonction Char qui renvoie le caractère correspondant à un entier entre 0 et 255
- Écrire la fonction premmajuscule qui met le premier caractère d'une chaîne de caractères en majuscule.
- Écrire la fonction majuscule qui met une chaîne de caractères en majuscule,
- Écrire la fonction minuscule qui met une chaîne de caractères en minuscule
- Les tester dans le corps de la fonction main()## Solution

```
#include <stdio.h>
#include <string.h>

int min(int x, int y)
{
    if ( x > y) return y ;
    else return x;
}

int max (int x, int y )
{
    if ( x>y) return x;
    else return y;
}

int ascii ( char c)
{
    return ((int)c);
}

char Char ( int i)
{
    return ((char)i);
}

char *PremMajuscule( char c[])
{
    if ( (int)c[0] >=97 && (int)c[0] <= 97+26-1 )
        c[0]=(char)(c[0]-97+65);
    return ( c );
}

char * Majuscule ( char c[] )
{
    int i;
    for ( i=0;i<strlen(c);i++)
    {
        if ( (int)c[i] >=97 && (int)c[i] <= 97+26-1 )
            c[i]=(char)(c[i]-97+65);
    }
    return (c);
}

char * Minuscule ( char c[] )
{
    int i;
    for ( i=0;i<strlen(c);i++)
    {
        if ( (int)c[i] >=65 && (int)c[i] <= 65+26-1 )
            c[i]=(char)(c[i]-65+97);
    }
}
```



```

    }
    return (c);
}

void main()
{
    printf ("Max 2 4 : %d\n",max(2,4));
    printf ("Min 2 4 : %d\n",min(2,4));
    printf ("Code ASCII de %c : %d\n",'0',ascii('0'));
    printf ("Caractère ASCII de %d : %c\n",65,Char(65));
    char c[100]="zora";
    printf ("PremMajuscule => %s\n",PremMajuscule(c));
    printf ("Majuscule : %s\n",Majuscule(c));
    printf ("Minuscule : %s\n",Minuscule(c));
}

```

Exercice 10 : Message d'erreur

Énoncé

- Afficher un message d'erreur via `#error` si la variable `TARGET` n'est pas définie
- Si la variable `TARGET` est définie on l'affichera à l'écran

Solution

```

#include <stdio.h>
#include <string.h>

#define TARGET "arm9t"

void main ()
{

#ifdef TARGET
#error "Pas de TARGET"
#else
#define getchaine(x) #x
#define okchaine getchaine(TARGET)
char str[30];
printf ("%s\n",TARGET);
#endif

}

```