B+


```c
#include <stdio.h>

/*Question 1*/
void swap_int(int *a, int *b) {
 int x = *a;
 *a = *b;
 *b = x;
}

/*Question 2*/
int main(void){
    //int var1 = 1;
    //int var2 = 2;

    int var1,var2;
    printf("Entrez les valeurs");
    scanf("%d%d", &var1, &var2);
    swap_int(&var1, &var2);
    printf("var 1 = %d\n",var1);
    printf("var 2 = %d\n",var2);
    return 0;
}
```

✓

✓

Il est prudent de vérifier la valeur retournée par scanf (ici, ça devrait être 2).

```c
#include<stdio.h>
#include "mcu_putint.h"

/*Question 1*/
void division(int diviseur, int dividende,int * quotient, int * reste){

    *quotient = diviseur/dividende;
    *reste = diviseur%dividende;
}

/*Question 2*/
int main(void){
    int x,y,var1,var2;
    printf("Entrez les valeurs");
    scanf("%d%d", &var1, &var2);
    division(var1, var2, &x, &y);
    putdec(x);
    printf("\n");
    putdec(y);
    printf("\n");
    return 0;
}
```

Attention à la division par 0.

✓

```c
#include <stdio.h>
#include <stdlib.h>

/*Question 1*/
int main (int argc, char *argv[]) {
    int i;

    for(i = 1; i < argc ; i++) {
        printf("%s ", argv[i]);
    }
    putchar('\n');

    exit(EXIT_SUCCESS);
}


/*Question 2
- Pour % mecho "Hello world !" :
un tableau d'une case contenant "Hello world !"
[Hello world !]

- Pour % mecho Hello world !
un tableau de trois case contenant chacun un mot de la phrase :
[Hello]
[world]
[!]

- Pour chaque paramÃ¨tres (ou mot) passÃ© Ã  cette fonction, elle l'imprime sur
la sortie standard
*/
```

de pointeurs

argv[0] contient le nom du programme (ici, "mecho").

✓

Eviter les accents / caractères spéciaux dans les fichiers de code.

```c
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <ctype.h>

/*Question 1*/
char * skip_spaces(char t[]){
    int i = 0;
    while(t[i] != '\0'){
        if (t[i] != ' '){
            return &t[i];
        }
        i++;
    }
    return &t[i];
}

/*Question 2*/
int main(int argc, char *argv[]){
    char * strip;
    int i;
    assert(argc == 2);

    printf("argv : %s\n", argv[1]);
    strip = skip_spaces(argv[1]);
    printf("strip : %s\n", strip);

    for (i=0 ; strip[i]; i++)
        strip[i] = toupper(strip[i]);

    printf("strip : %s\n", strip);
    printf("argv : %s\n", argv[1]);

    exit(EXIT_SUCCESS);
}
```

✓ → répondre à la question !

```c
#include <stdio.h>
#include <stdlib.h>

/*Question 1
 * Le prototype d'une fonction permettant l'Ã©change de valeurs de deux pointeur
s
 * de type int* est int ** (pointeur pointant Ã  un pointeur)
 */

/*Question 2*/
void swap_ptr(int **x, int **y){
    int i,*l=&i,**u=&l;
    *u = *x;
    *x = *y;
    *y = *u;
}

/*Question 3*/
int main(){
    int a, b;
    int *x = &a;
    int *y = &b;

    swap_ptr(&x, &y);

    if ((x == &b) && (y == &a)) {
        printf("OK ;)\n");
        exit(EXIT_SUCCESS);
    }

    else {
        printf("KO ;(\n");
        exit(EXIT_FAILURE);
    }
}
```

C'est un peu compliqué. Ici, on veut stocker la valeur pointée par x dans une variable tampon. Cette valeur est de type int* (puisque x est un int**), donc il suffit d'une variable l de type int*:

```
int *l;
l = *x;
:
```

Note: il n'est pas indispensable d'initialiser un pointeur à sa déclaration (ou, mieux, on peut l'initialiser avec NULL). Donc ;, par exemple, est inutile ici.

```c
#define SIZE 10

#include <stdio.h>

float * search_dicho(float v, float *tab, int size){

    float * res = NULL;
    int middle = (size/2);
    if(size < 1){
        res = NULL;
    }
    else if(tab[middle] == v){
        res = &tab[middle];
    }
    else if(tab[middle] > v){
        res = search_dicho(v, &tab[0], middle);
    }
    else if(tab[middle] < v){
        res = search_dicho(v, &tab[middle + 1], size - (middle + 1));
    }
    return res;
}

int main(void){

    float tab[SIZE] = {1,15.53,20.89,27,38,42.2,63,64.9,78,80};
    float * p = &tab[0];
    float * res = NULL;
    float x;
    scanf("%f", &x);
    res = search_dicho(x, p, SIZE);
    if(res == NULL){
        printf("Po lo\n");
    }
    else {
        printf("l'Ã©lement %f est lÃ , merci Ã  lui\n", *res);
    }
    return 0;
}
```

```c
#include <stdio.h>

extern int putchar(int);

static int put_digit(int digit){

  int res = 0;

  if ( digit < 0 || digit > 9){
    res = -1;
  }

  else {

    int b = putchar(digit + 48);

    if (b == EOF){
      res = -1;
    }

  }

  return res;
}

int putdec(int d){

    int res = 0;
    int index = 1000000000;
    int b;

  if (d == 0){
    putchar('0');
  }

  if (d < 0){
    putchar('-');
  }


  while (index > 0 && ((d / index) % 10 == 0)){
      index = index / 10;
        }


    while (index > 0){
      int chiffre = (d/ index) % 10;
      chiffre = (chiffre < 0) ? -chiffre : chiffre;
      b = put_digit(chiffre);
      index = index / 10;

      if(b == -1){
        res = -1;
      }

    }

    return res;

}
```

```
/*
   mcu - affichage d'entiers
*/

#ifndef _MCU_PUTINT_H_
#define _MCU_PUTINT_H_

/* Affiche (sous forme dÃ©cimale) un entier donnÃ© en paramÃ¨tre */
extern void putdec(int n) ;

#endif
```

inutile