REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de M'sila
Faculté des Mathématiques et de l'Informatique
Département d'informatique

جامعة المسيلة
كلية الرياضيات والإعلام الآلي
قسم الإعلام الآلي

Niveau: 1e année informatique
Matière: ASD2

Corrige type TD/TP N° : 03

Année universitaire : 2022/2023
Chapitre 1 : Les pointeurs

Exercice 1 : (TD)

Complétez le tableau suivant qui indique la valeur de chaque variable après chaque instruction.

instruction	**a**	**b**	**c**	**p1**	**p2**
int a, b, c,*p1,*p2;	/	/	/	/	/
a=1;b=2;c=3;	1	2	3	/	/
p1=&a;p2=&c;	1	2	3	&a	&c
*p1=(*p2)++;	3	2	4	&a	&c
p1=p2; p2=&b;	3	2	4	&c	&b
*p1-=*p2;	3	2	2	&c	&b
instruction	**a**	**b**	**c**	**p1**	**p2**
++*p2;	3	3	2	&c	&b
p1=*p2;	3	3	6	&c	&b
a=++*p2**p1;	24	4	6	&c	&b
p1=&a;	24	4	6	&a	&b
*p2=*p1/=*p2;	6	6	6	&a	&b

Exercice 2 : (TP)

Écrivez un programme qui remplit un tableau T de nombres réels, puis crée deux tableaux TP et TN, et place tous les nombres positifs dans TP et tous les nombres négatifs dans TN, et laisse les nombres nuls tels quels.

```
#include <stdio.h>
#include <stdlib.h>
int main() {
  int *T,*TP,*TN, n, i, j, k;
  printf("entrez nbr des elements\n");
  scanf("%d",&n);
  T=(int*)malloc(n*sizeof(int));
  for( i=0; i<n; i++) {
    printf("T[%d]=",i);
    scanf("%d",T+i);
  }
  TP=(int*)malloc(n*sizeof(int));
  TN=(int*)malloc(n*sizeof(int));
  j=k=0;
  for( i=0; i<n; i++)
    if(T[i]>0)
      TP[j++]=T[i];
    else if(T[i]<0)
      TN[k++]=T[i];
  TP=(int*)realloc(TP, j*sizeof(int));
  TN=(int*)realloc(TN, k*sizeof(int));
  printf("\nla table des positifs\n");
  for( i=0; i<j; i++)
    printf("%d\t",TP[i]);
  printf("\nla table des négatifs\n");
  for( i=0; i<k; i++)
    printf("%d\t",TN[i]);
  return 0;
}
```

Exercice 3 : (TD)

Soit ***p*** un pointeur pointant sur le tableau T :

```
int T[] = {8, 17, 7, 9, 48, 76, 22, 27};   int *p =T;
```

Quelle est la valeur ou l'adresse renvoyée par chacune des expressions suivantes :

1. *p+3	T[0]+3 ⇒11
2. p+(*p-7)	p+(T[0]-7) ⇒p+1⇒&T[1]
3. &T[5]-p	p+5-p⇒5
4. &p+1	&p+1
5. *(p+3)	*(&T[3]) ⇒9
6. *(P+*(P+7)-T[6])	*(p+T[7]-T[6]) ⇒T[5] ⇒76
7. T+2	p+2⇒&T[2]
8. &T[5]-2	p+5-2⇒p+3⇒&T[3]

Exercice 4 : (TD/TP)

Écrivez la fonction ***strcat*** qui concatène deux chaînes de caractères dans une nouvelle chaîne. (utiliser l'opérateur de déréférencement * au lieu de []).

```
#include <stdio.h>
#include <stdlib.h>
char * strcat(char *s1,char *s2) {
  char *s;
  int i=0, n=strlen(s1)+strlen(s2)+1;
  s=(char*)malloc(n*sizeof(char));
  while(*s1){
    *(s+i)=*s1;
    i++; s1++;
  }
  while(*s2){
    *(s+i)=*s2;
    i++; s2++;
  }
  *(s+i)='\0';
  return s;
}
int main() {
 char * s;
 char str1[100] = "Hello, ";
 char str2[100] = "world!";
 s = strcat(str1, str2);
 printf("After concatenation: %s", s);
 return 0;
}
```

Exercice 5 : (TP)

Écrivez la fonction ***copy*** qui copie une partie d'un tableau. La fonction prend le tableau et sa taille, le

début et la longueur a copié et renvoie un pointeur vers le nouveau tableau.

```
#include <stdio.h>
#include <stdlib.h>
int * copy(int *t,int n,int pos,int size)
{ int i, *c;
  If (pos+size>n) return NULL;
  c=(int*)malloc(size*sizeof(int));
  for(i=0; i<size; i++)
    c[i]=t[i+pos];
  return c;
}
int main() {
 int i, pos=2, size=3;
 int t[] = {1, 2, 3, 4, 5, 6};
 int *c = copy(t, 6, pos, size);
 if(c !=NULL){
  for (i = 0; i < size; i++)
    printf("%d ", c[i]);
  free(c);
 }
 return 0;
}
```

Exercice 6 : (TD/TP)

Supposons que nous avons une image en noir et blanc de dimensions n×m, stockée sous la forme d'une matrice d'entiers. Écrivez une fonction nommée " ***flip*** " qui retourne l'image verticalement

```
#include <stdio.h>
#include <stdlib.h>
int ** flip(int **A,int n,int m) {
  int i, j, **T;
  T=(int**) malloc( n*sizeof(int*));
  for(i=0; i<n; i++)
    T[i]=(int*) malloc(m*sizeof(int));
  for(i=0; i<n; i++)
    for(j=0; j<m; j++)
      T[i][j]=A[n-i-1][j];
  return T;
}
```

```
void flip2(int **B, int n) {
  int i, *lin;
  for(i=0; i<=n%2; i++) {
    lin=B[i];
    B[i]=B[n-i-1];
    B[n-i-1]=lin;
  }
}
```

```
int main() {
  int i, j, n = 2, m = 3,**A,**B;
  //creer A
  A = (int**)malloc(n*sizeof(int*));
  for (i = 0; i < n; i++)
   A[i] = (int*)malloc(m*sizeof(int));
  //remplir A
  for ( i = 0; i < n; i++)
    for ( j = 0; j < m; j++)
      A[i][j] = i*m + j + 1;
  //afficher A
  printf("Before flip:\n");
  for ( i= 0; i < n; i++) {
    for (j = 0; j < m; j++)
      printf("%d ", A[i][j]);
    printf("\n");
  }
  B= flip(A, n, m);
  printf("After flip:\n");
  for ( i= 0; i < n; i++) {
    for (j = 0; j < m; j++)
      printf("%d ", B[i][j]);
    printf("\n");
  }
  //supprimer A
  for (i = 0; i < n; i++)
    free(A[i]);
  free(A);
//supprimer B
  for (i = 0; i < n; i++)
    free(B[i]);
  free(B);
  return 0;
}
```

Exercice 7 : (TD)

Écrivez la fonction ***Mat2Tab*** qui convertit une matrice en un tableau.

Université de M'sila
Faculté des Mathématiques et de l'Informatique
Département d'informatique

جامعة المسيلة
كلية الرياضيات والإعلام الآلي
قسم الإعلام الآلي

```c
#include <stdio.h>
#include <stdlib.h>
int* Mat2Tab(int**M,int n, int m) {
  int i, j, k=0, *T;
  T=(int*)malloc(n*m*sizeof(int));
  for( i=0; i<n; i++)
    for( j=0; j<m; j++) {
      T[k]=M[i][j];
      k++;
    }
  return T;
}
int main() {
  int i, j, n = 2, m = 3,**mat,*T;
  //creer mat
  mat = (int**)malloc(n*sizeof(int*));
  for (i = 0; i < n; i++)
   mat[i] = (int*)malloc(m*sizeof(int));
  //remplir mat
  for ( i = 0; i < n; i++)
    for ( j = 0; j < m; j++)
      mat[i][j] = i*m + j + 1;
  //afficher mat
  printf("2D array:\n");
  for ( i= 0; i < n; i++) {
    for (j = 0; j < m; j++)
      printf("%d ", mat[i][j]);
    printf("\n");
  }
  //transformer mat a 1d
  T = Mat2Tab(mat, n, m);
  //afficher T
  printf("1D array:\n");
  for ( i = 0; i < n*m; i++)
    printf("%d ", T[i]);
  //supprimer T
  free(T);
  //supprimer mat
  for (i = 0; i < n; i++)
    free(mat[i]);
  free(mat);
  return 0;
}
```