**Ecole Supérieure de Technologie d'Essaouira D.U.T Info & IDSD**

**Année 2020-2021**

**Enseignants : N. Chouhad ; F. Karami & S.Gounane**

**Matière : Algorithmique & Programmation (Langage C)**

**Exemples de cours:**

0-Testez les exemples de cours.

1- Donner le prototype d'une fonction nommée EstPremier testant si un nombre n est ou non

premier.

2- Donner le prototype d'une fonction DemandePoint qui demande à l'utilisateur les

coordonnées (x,y) d'un point (on ne demande pas d'écrire le code de la fonction).

3- Soit le programme

main(){

int a;

int \*pa;

int \*\*ppa;

a = 1;

pa = &a;

ppa = &pa;

}

Que valent les expressions suivantes &a, \*pa, &pa, \*ppa, \*\*ppa , &ppa?

Parmi les expressions précédentes, lesquels désignent la même la valeur que celle contenue

dans a, pa et ppa?

4- Que fait la fonction C suivante ?

void Mystere( int \*dp, int \*sp ) {

if ( \*sp > 0 )

\*dp = \*sp;

else

\*dp = 0;

}

5- Dans le programme suivant, rayez les lignes illégales (erreurs). Qu'est ce qui s'affiche ?

main( ) {

int i = 0;

int \*p;

float x = 3.14;

float \*f;

p = &i;

\*f = 666;

f = &x;

\*f = \*p;

\*p = 34;

p = f ;

\*p = \*p + 1;

printf( "%d %f\n", i, \*f ); }

REPONSE🡺

Example1 :

main(){

int v=12;

int u=10;

int \*vP; /\*pointeur sur int\*/

vP = &v; /\*affectation du pointeur \*/

u = \*vP;

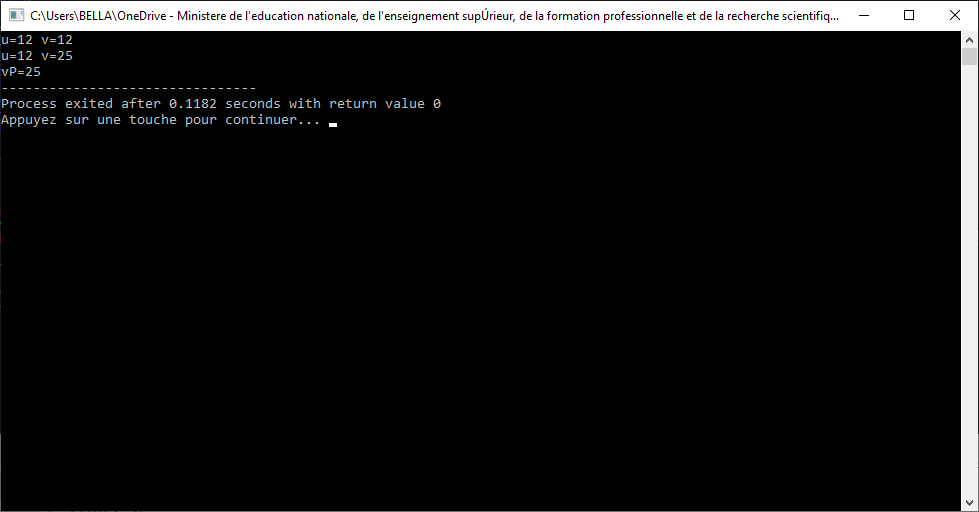
printf("u=%d v=%d\n",u,v);

\*vP = 25;

printf("u=%d v=%d\n",u,v);

printf("vP=%d",\*vP);

}



Example 2:

#include<stdio.h>

#include<conio.h>

int main( ){

float a , \*ptr; /\*supposons que ces variables sont reprÃ©sentÃ©es en

mÃ©moire Ã  partir de lâ€™adresse 1000Â   
\*/

//clrscr(); // pour effacer lâ€™Ã©cran<conio.h>

ptr = &a;

printf("Entrer une valeur : ");

scanf("%f" , ptr); // on saisie la valeur 12.4Â

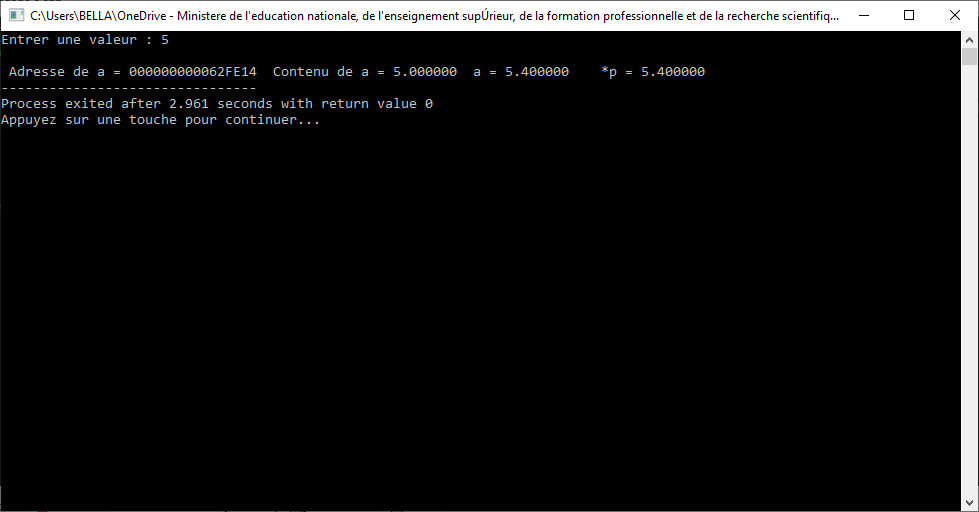
printf("\n Adresse de a = %p Contenu de a = %f" ,ptr,\*ptr);

\*ptr += 0.4;

printf(" a = %f \*p = %f ", a,\*ptr);

return(0);

}



Example 3:

#include <stdio.h>

#define N 10

main(){

float T[N] , \*pt ;

printf("Entrez %d entiers\n", N) ;

for (pt = T ; pt<T+N; pt++)

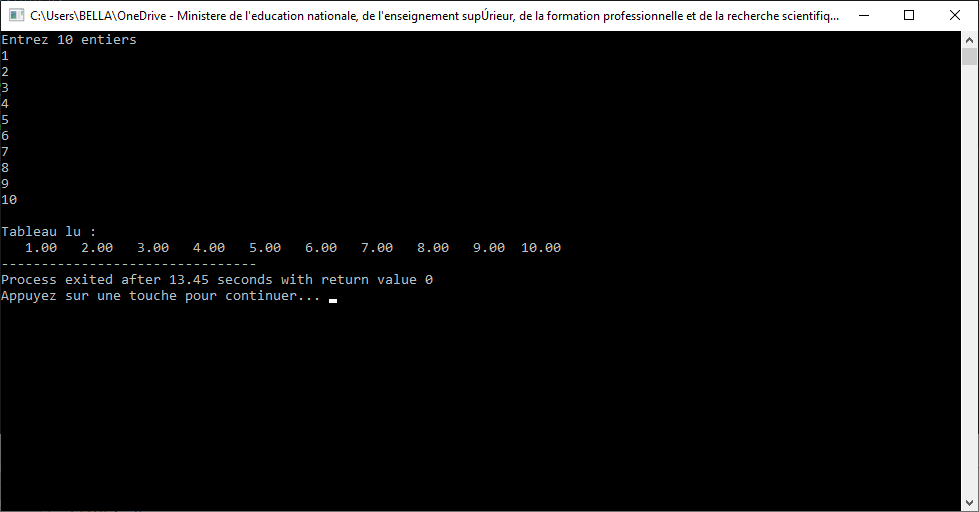
scanf("%f", pt) ;

printf("\nTableau lu : \n") ;

for (pt = T ; pt<T+N; pt++)

printf("%7.2f", \*pt) ;

}



Example4:

main(){

short \*pt;

int N , i;

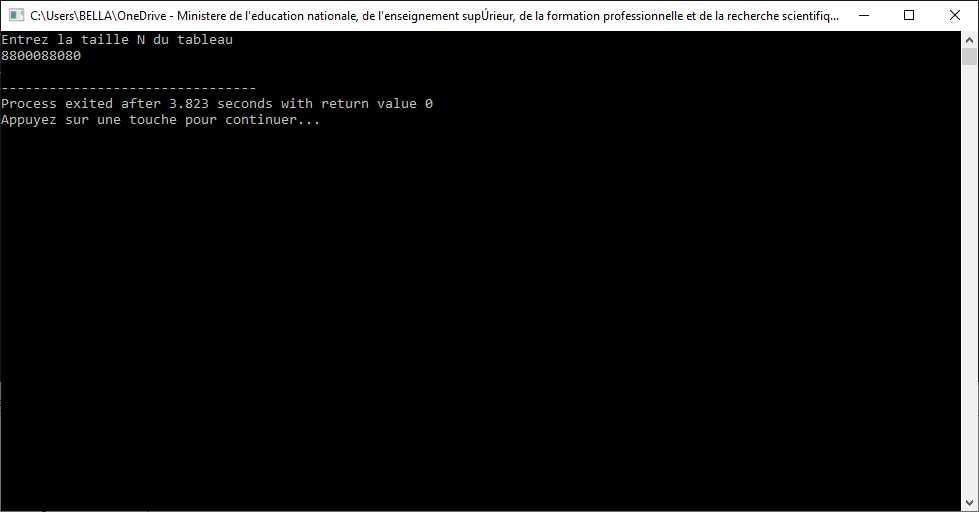
printf("Entrez la taille N du tableau \n") ;

scanf("%d", &N) ;

pt = ( short \* ) **malloc**( N \* **sizeof**( short ) );

if (**pt == NULL)**

printf("Mémoire non disponible") ;



1&2-le prototype

I do not know what a prototype looks like

3-

&a : adress de a

\*pa : valeur de a

&pa :valeur de a

\*ppa :adress de a

\*\*ppa :valr de a

&ppa :vlr de a

4-

void Mystere( int \*dp, int \*sp ) {

if ( \*sp > 0 )

\*dp = \*sp;

else

\*dp = 0;

}

Si \*sp a une adresse donc on mette l’adresse de sp dans \*dp

5-

main( ) {

int i = 0;

int \*p;

float x = 3.14;

float \*f;

p = &i;

\*f = 666;

f = &x;

\*f = \*p;

\*p = 34;

p = f ;

\*p = \*p + 1;

printf( "%d %f\n", i, \*f ); }

**Exercice 0 :**

Ecrire un programme dans lequel vous :

♦Déclarez un entier **i** et un pointeur **p** sur un entier,

♦Initialiser l’entier **i** par une valeur arbitraire et pointez le pointeur **p** sur **i,**

♦Modifiez l’entier pointer par **p** (en utilisant **p** et non **i**)

♦Imprimez la valeur de **i** et de **p**

main( ) {

int i;

int \*p;

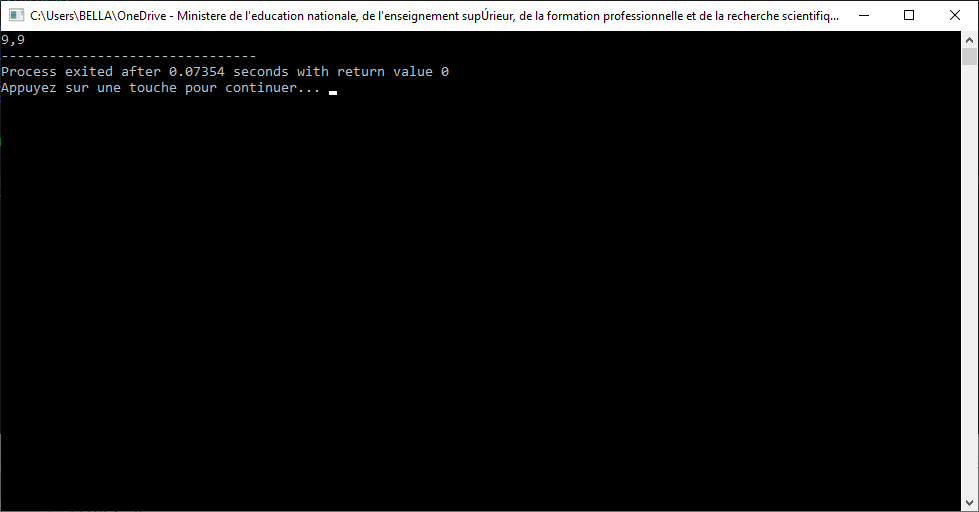
i=55;

p=&i;

\*p=9;

printf("%d,%d",i,\*p);

}



**Exercice 1 :** On considère que les déclarations suivantes ont été faites :

**int a;**

**char tab[10];**

Une expression avec pointeurs (resp. sans pointeurs) vous est donnée, vous devez la réécrire

sans (respectivement avec) l’usage explicite des pointeurs.

1. \*(&a)

2. \*tab

3. \*(tab + 0)

4. (\*tab) + 1

5. &(tab[0])

6. &(tab[i])

7. ++tab[i]

Reponse🡺

1.a

2.tab[0]

3.tab[0]

4.tab[1]

5.tab[0]

6.tab[i]

7.++tab[i]

**Exercice 2:** Il est demandé dans cet exercice de représenter en mémoire les données

déclarées dans un programme, ainsi que leurs différentes valeurs, à un moment donné de

l’exécution. Pour cela, vous représenterez l’occupation des données en mémoire dans un

tableau à 3 colonnes comme montré ci-dessous :

**identificateur adresse valeur**

a ? ?

**... ... ...**

Pour déterminer les adresses, on fera les approximations suivantes :

– les données sont réservées en mémoire dans l’ordre de leur déclaration

– la première adresse démarre à 10001

– l’architecture est 32 bits

Le programme est donné ci-dessous. Notez que le programme fait une utilisation abusive des

pointeurs, l’objectif étant simplement de vous familiariser avec la syntaxe et la sémantique

des instructions manipulant des pointeurs.

**#include <stdlib.h>**

**#include <stdio.h>**

**main(){**

**int a = 10;**

**int b = 5;**

**int tab[3] = {1,2,3};**

**int \*p\_tab;**

**}**

1) Représentez l’occupation mémoire ?

*2)* On ajoute au programme les instructions suivantes :

**tab[0] = a;**

**\*(tab + 1) = b;**

**p\_tab = tab + 2;**

Représentez l’occupation mémoire ?

3) Maintenant on ajoute au programme les instructions suivantes :

**\*p\_tab = \*(p\_tab - 1);**

**- -p\_tab;**

**\*p\_tab = \*(p\_tab- 1);**

**- - p\_tab;**

**\*p\_tab= \*(p\_tab + 2);**

Représentez l’occupation mémoire ?

REPONSE 🡺

1.

|  |  |  |
| --- | --- | --- |
| A | 10001 | 10 |
| B | 10002 | 5 |
| Tab[0] | 10003 | 1 |
| Tab[1] | 10004 | 2 |
| Tab[2] | 10005 | 3 |
| \*p\_tab | 10006 | Non initialise |

2.

|  |  |  |
| --- | --- | --- |
| A | 10001 | 10 |
| B | 10002 | 5 |
| Tab[0] | 10003 | 10 |
| Tab[1] | 10004 | 5 |
| Tab[2] | 10005 | 3 |
| \*p\_tab | 10005 | 3 |

3-

|  |  |  |
| --- | --- | --- |
| A | 10001 | 10 |
| B | 10002 | 5 |
| Tab[0] | 10003 | 10 |
| Tab[1] | 10004 | 5 |
| Tab[2] | 10005 | 3 |
| \*p\_tab | 10003 | 3 |

**Exercice 3 :** Soit le programme :

**main{**

**int A = 1, B=2, C=3;**

**int \*P1, \*P2;**

**P1=&A;**

**P2=&C;**

**\*P1=(\*P2)++;**

**P1=P2;**

**P2=&B;**

**\*P1-=\*P2;**

**++\*P2;**

**\*P1\*=\*P2;**

**A=++\*P2\*\*P1;**

**P1=&A;**

**\*P2=\*P1/=\*P2;**

**}**

Quels résultats fournit par ce programme? Donnez la valeur de chaque variable?

A=6 B=6 C=6 \*P1=6 \*P2=6

**Exercice 4 :** Soit le programme :

**#include <stdio.h>**

**main()**

**{**

**void echange (int \* ad1, int \* ad2) ;**

**int a=10, b=20 ;**

**printf ("avant appel %d %d\n", a, b) ;**

**echange (&a, &b) ;**

**printf ("après appel %d %d", a, b) ;**

**}**

**void echange (int \* ad1, int \* ad2){**

**int x ;**

**x = \* ad1 ;**

**\* ad1 = \* ad2 ;**

**\* ad2 = x ;**

**}**

Que fait ce programme? Quelle est la difference avec le passage par valeur?

Le programme changer les valeur de a et b mais en utilisent leur adresse

Pour le différence : dans le passage par adresse on ne pas besoin de ajoute un return a la fonction .

**Exercice 5** : Ecrire de deux façons différentes, un programme qui lit 10 nombres entiers

dans un tableau avant d’en rechercher le plus grand et le plus petit :

a) En utlisant le formalisme tableau

b) En utilisant le formalisme pointeur

reponse 🡺

#include <stdio.h>

main()

{

int max,tab[10],i;

for(i=0;i<10;i++)

{

printf("donne la valuer %d dans le tableau:",i+1);

scanf("%d",&tab[i]);

}

//1er facon:

max=0;

for(i=0;i<9;i++)

{

if(tab[i]>tab[i+1])

max=tab[i];

else

max=tab[i+1];

}

//2eme facon

for(i=0;i<9;i++)

{

if(\*(tab+i)>\*(tab+i+1))

max=\*(tab+i);

else

max=\*(tab+i+1);

}

}

**Exercice 6 :** Soit le programme :

**#include<stdio.h>**

**#define dim 10**

**main (){**

**int tab[dim] = { 6,7,8,9,1,0,3,2,5,4 };**

**int i,j,tmp;**

**for(i=0;i<dim-1;i++)**

**for (j=i;j<dim;j++)**

**if (tab[i]>tab[j]) {**

**tmp=tab[i];**

**tab[i]=tab[j];**

**tab[j]=tmp;**

**} }**

1) Que fait ce programme?

2) Quell résultat fournit ce code ?

3) Réécrire ce programme en utilisant, dans le corps de la boucle de tri, un pointeur p

pointant sur le premier élément du tableau tab, au lieu de tab[x]. Le programme devra

contenir la déclaration ainsi que l’instruction suivantes:

**int \*p;**

**p=tab;**

**reponse🡺**

1.ce programme permet de arrange les valeurs d’un tableau 9 4 8🡺 4 8 9

2.le resultat est : 0 1 2 3 4 5 6 7 8 9

3.

#include<stdio.h>

#define dim 10

main (){

int tab[dim] = { 6,7,8,9,1,0,3,2,5,4 };

int i,j,tmp;

int \*p;

p=tab;

for(i=0;i<dim-1;i++)

for (j=i;j<dim;j++)

if (\*(p+i)>\*(p+j)) {

tmp=\*(p+i);

\*(p+i)=\*(p+j);

\*(p+j)=tmp;

} }

**Exercice 7 :**

**1)** Ecrire une fonction qui fournit en retour la somme des valeurs d’un tableau de

flottants dont la dimension est fournie en argument.

2) Ecrire une fonction qui fournit en retour la somme des valeurs d’un tableau de

flottants deux indices dont les dimensions sont fournies en argument.

Reponse🡺

#include<stdio.h>

#define N 5

float tab[N],tab1[N][N];

int i,j;

float some\_fonc(int x);

float some\_fonc1(int x);

int main()

{

float x;

x=N;

float som,som1;

for(i=0;i<N;i++)

{

tab[i]=i\*5;

}

for(j=0;j<N;j++)

{

for(i=0;i<N;i++)

{

tab1[j][i]=i\*5;

}

}

som=some\_fonc(x);

som1=some\_fonc1(x);

printf("%f %f",som,som1);

}

float some\_fonc(int x)

{

float some=0;

for(i=0;i<x;i++)

{

some+=tab[i];

}

return(some);

}

float some\_fonc1(int x)

{

float some=0;

for(j=0;j<x;j++)

{

for(i=0;i<x;i++)

{

some+=tab1[j][i];

}

}

return(some);

}

**Exercice 8** : **1)** En utilisant l’allocation dynamique, écrire un programme permettant de

calculer la somme, la soustraction et la multiplication de deux Tableaux lus au clavier.

(Afficher les résultats au écran).

Reponse🡺

#include<stdio.h>

#include<stdlib.h>

#define N 4

main ()

{

int x,y,i,j,h;

float z;

char Operator;

for(j=0;j<N;j++)

{

printf("entre une operation: (+,-,\*,/)");

scanf("%c",&Operator);

printf("combiane des colons :");

scanf("%d",&y);

printf("le tableau 1\n");

float \*tab3;

tab3=(float\*)malloc(y\*sizeof(float));

float \*tab1;

tab1=(float\*)malloc(y\*sizeof(float));

for (i=0;i<y;i++)

{

printf("donnez la valeure d'element %d:",i+1);

scanf("%f",&z);

tab1[i]=z;

}

printf("le tableau 2\n");

float \*tab2;

tab2=(float\*)malloc(y\*sizeof(float));

for (i=0;i<y;i++)

{

printf("donnez la valeure d'element %d:",i+1);

scanf("%f",&z);

tab2[i]=z;

}

erruer:

{

for(i=0;i<y;i++)

switch (Operator)

{

case '+' : \*(tab3+i)=\*(tab1+i)+\*(tab2+i);break;

case '-' : \*(tab3+i)=\*(tab1+i)-\*(tab2+i);break;

case '\*' : \*(tab3+i)=\*(tab1+i)\*\*(tab2+i);break;

case '/' : \*(tab3+i)=\*(tab1+i)/ \*(tab2+i);break;

default : printf("\noperation is not correct:");break;

}

}

printf("la resultats de le operation (%c) entre les deux tableaus est:\n",Operator);

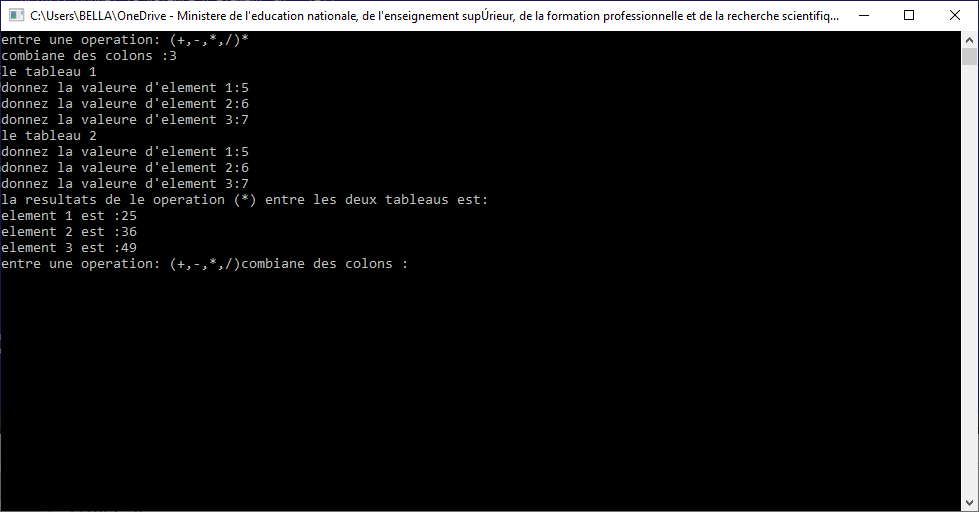
for(i=0;i<y;i++)

{ printf("element %d est :%g\n",(i+1),\*(tab3+i));

}

}

}



**2)** Reprenez les exercices 0, 1 2 et 3 de la feuille TP N°4 (juste Tableau une

dimension) et remplacez les tableaux statiques par une allocation dynamique.

0.1

#include<stdio.h>

#include<stdlib.h>

int main ()

{

int x,i,y;

printf("un programme permettant la saisie d'un tableau de reels, et un tableau de caractere. ");

printf("\nle tableau de reel\n");

printf("combiane des element:");

scanf("%d",&x);

float \*tlab;

tlab=(float\*)malloc(x\*sizeof(float));

for (i=0;i<x;i++)

{

printf("donnez le nomber d'element %d:\n",i+1);

scanf("%f",&y);

\*(tlab+i)=y;

}

printf("\nle tableau de char\n");

printf("combiane des element:");

getchar();

scanf("%d",&x);

char tab[x];

for (i=0;i<x;i++)

{

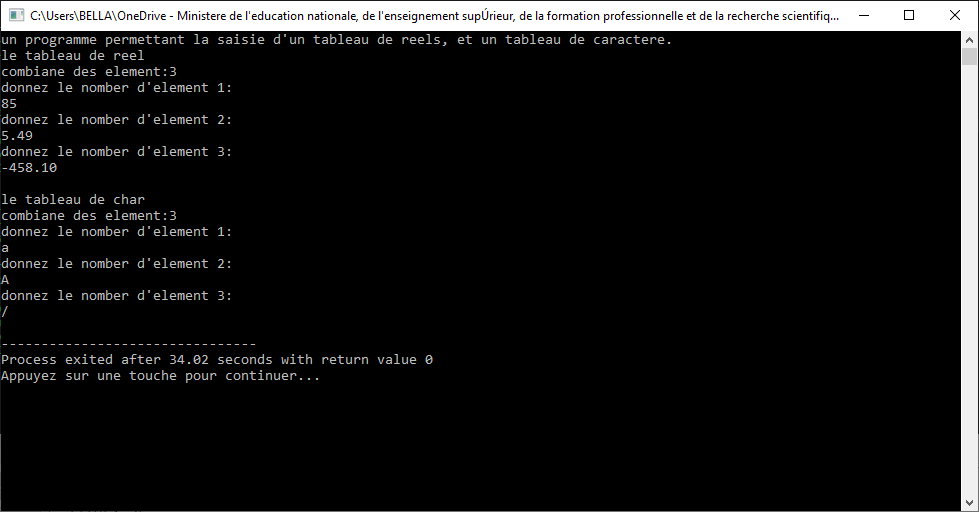
getchar();

printf("donnez le nomber d'element %d:\n",i+1);

scanf("%c",&y);

\*(tlab+i)=y;

}

}

0.2

#include<stdio.h>

#include<stdlib.h>

int main ()

{

int x,i,y;

printf("un programme permettant la saisie d'un tableau de reels, et un tableau de caractere. (2 demontion)");

printf("\nle tableau de reel\n");

printf("combiane des element:2\*");

scanf("%d",&x);float \*\*tlab;

tlab=(float\*\*)malloc(2\*sizeof(float\*));

for(i=0;i<2;i++)

\*(tlab+i)=(float\*)malloc(x\*sizeof(float));

for (i=0;i<2\*x;i++)

{

printf("donnez le nomber d'element %d:\n",i+1);

scanf("%f",&y);

\*\*(tlab+i)=y;

}

printf("\nle tableau de char\n");

printf("combiane des element:");

getchar();

scanf("%d",&x);

char tab[2][x];

for (i=0;i<2\*x;i++)

{

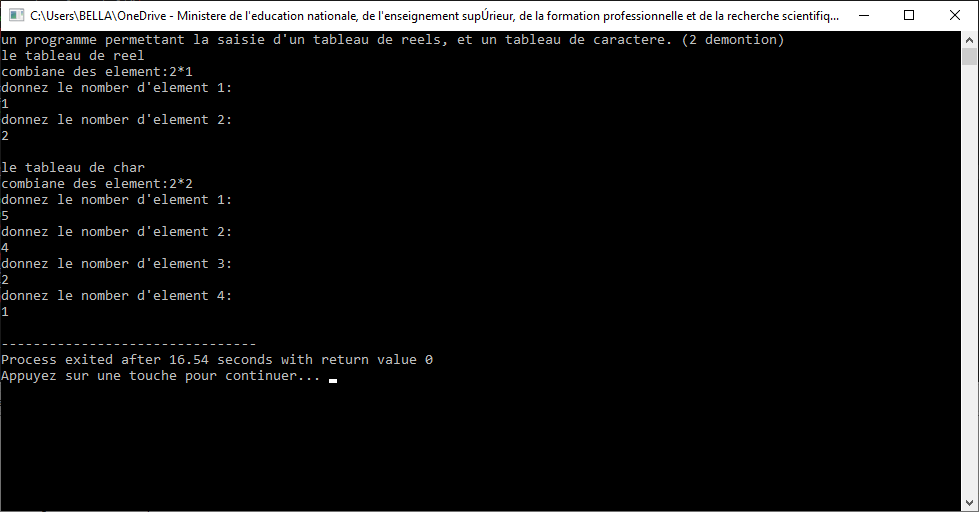
getchar();

printf("donnez le nomber d'element %d:\n",i+1);

scanf("%c",&y);

\*\*(tab+i)=y;

}

}

1.1

Déjà dans le 1er question

1.2

#include<stdio.h>

#include<stdlib.h>

int main ()

{

int a,b,c,x,y,i,d,z,j,t,u,l,q,g;

char Operator;

printf("On this programe we are going to calculate the operation between two matrixes\n");

obit:{

printf("\nChose the Operator (+ or - or \*)\n");

scanf("%c",&Operator);

if (Operator=='\*')

printf("\nNote : the number of colons in the first matrix must be same as the lines of the second one\n");

else if (Operator=='+' || Operator=='-')

printf("\nNote : the number of colons and lines should be the same\n");

else

{

printf("\nInvalid operator!!!!!");

goto obit;

}

obito:{

printf("How many lines and colons on the first matrix :");

scanf("%d %d",&x,&y);

printf("How many lines and colons on the second matrix :");

scanf("%d %d",&z,&t);

int k=x,n=y;

int \*\*A;

A=(int\*\*)malloc(k\*sizeof(int\*));

for(i=0;i<k;i++)

\*(A+i)=(int\*)malloc(n\*sizeof(int));

k=z;

n=t;

int \*\*B;

B=(int\*\*)malloc(k\*sizeof(int\*));

for(i=0;i<k;i++)

\*(B+i)=(int\*)malloc(n\*sizeof(int));

printf("\nEnter the nummber of the first matrix\n");

for(j=0,d=0;d<x;j+=y,d++)

{

for(i=0;i<y;i++)

{

printf(" \nLine %d colon %d:\n",j+1,i+1);

scanf("%d",&u);

\*(\*A+j+i)=u;

}

}

printf("\nEnter the nummber of the second matrix\n");

for(j=0,d=0;d<z;j+=t,d++)

{

for(i=0;i<t;i++)

{

printf(" Line %d colon %d:\n",j+1,i+1);

scanf("%d",&u);

\*(\*B+j+i)=u;

}

}

int C[x][t];

printf("First matrice %c second matric = \n",Operator);

for(j=0,d=0;d<x;j+=t,d++)

{

for(i=0;i<t;i++)

{ l=0;

switch (Operator)

{

case '+':(\*(\*C+j+i))=(\*(\*A+j+i))+(\*(\*B+j+i));break;

case '-':\*(\*C+j+i)=\*(\*A+j+i)-\*(\*B+j+i);if (y!=t && x!=y) {printf("As I said the number of colons and lines should be the same ");goto obito;};break;

case '\*':if(y!=z){printf("As I said the number of colons in the first matrix must be same as the lines of the second one ");goto obito;};if (y==z)

{

for(q=0,g=0;q<y,g<y\*z;q++,g+=z)

{

l=l+(\*(\*A+j+q))\*(\*(\*B+g+i));

}

\*(\*C+j+i)=l;

}

;break;

default: printf("somthing is not correct!!!");

}

printf(" %d",\*(\*C+j+i));

}

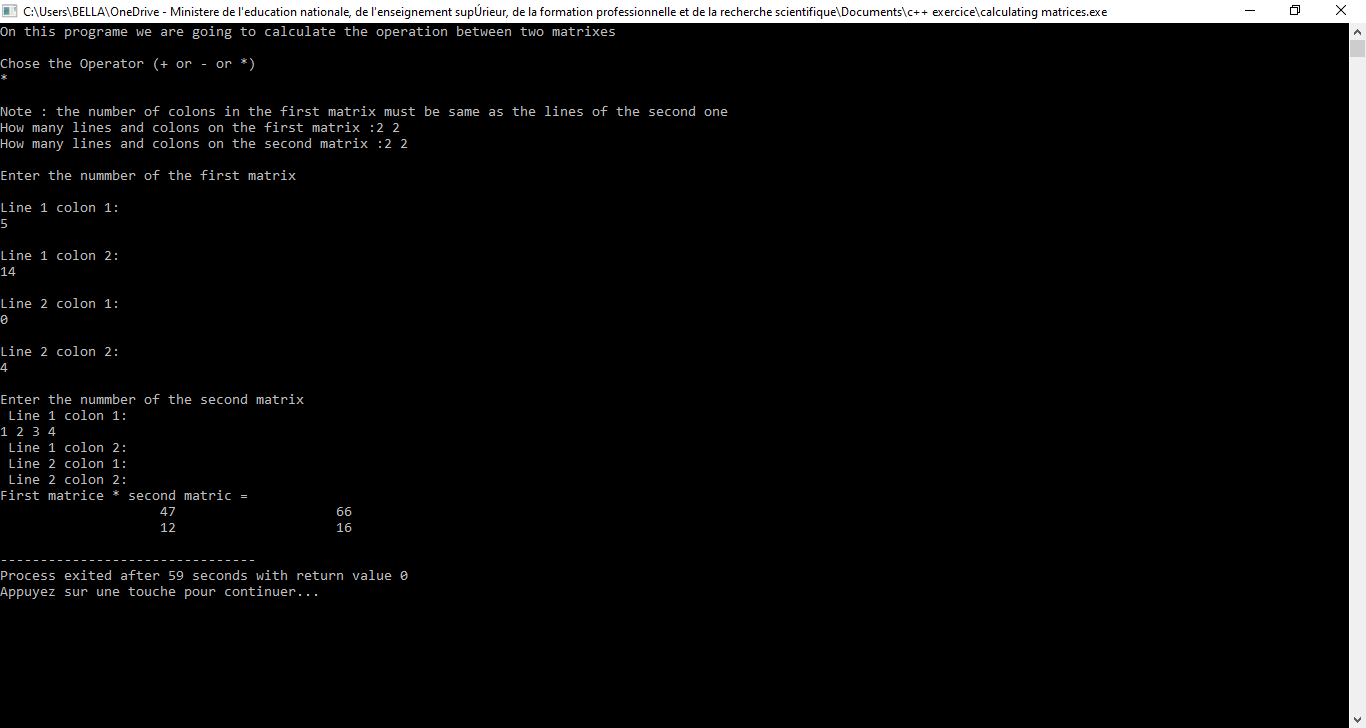
printf("\n");

}

}

}

}



1.3

#include<stdio.h>

#include<stdlib.h>

int main ()

{

int x,i,y;

printf("un programme permettant la saisie d'un tableau de reels, et un tableau de caractere. ");

printf("\nle tableau de entier\n");

printf("combiane des element:");

scanf("%d",&x);

int n=x;

int \*tab;

tab = (int\*)malloc(n \* sizeof(int));

int \*tlab;

tlab = (int\*)malloc(n \* sizeof(int));

for (i=0;i<x;i++)

{

printf("donnez le nomber d'element %d tab 1:",i+1);

scanf("%d",&y);

\*(tlab+i)=y;

}

for (i=0;i<x;i++)

{

printf("donnez le nomber d'element %d tab 2:",i+1);

scanf("%d",&y);

\*(tab+i)=y;

}

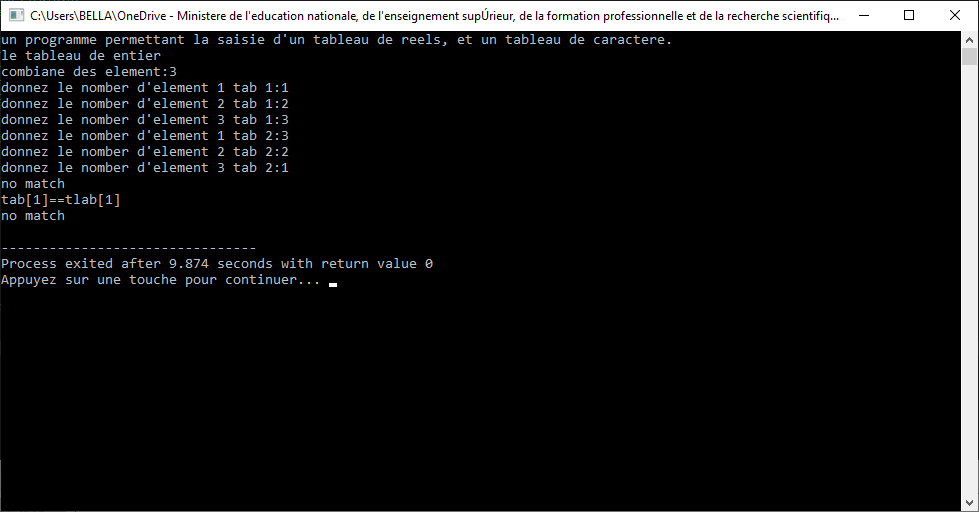
for(i=0;i<x;i++)

{

\*(tab+i)==\*(tlab+i)? printf("tab[%d]==tlab[%d]\n",i,i):printf("no match\n");

}

}



Ex2

2.1 :

#include<stdio.h>

#include<stdlib.h>

int main()

{

int y,z,i,x;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ce programes calcue le schutroumpf de deux tableaux\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("combiane des colons :");

scanf("%d",&y);

printf("le tableau 1\n");

int n=y;

int \*tab1;

tab1 = (int\*)malloc(n \* sizeof(int));

for (i=0;i<y;i++)

{

printf("donnez la valeure d'element %d:",i+1);

scanf("%d",&z);

\*(tab1+i)=z;

}

printf("le tableau 2\n");

int \*tab2;

tab2 = (int\*)malloc(n \* sizeof(int));

for (i=0;i<y;i++)

{

printf("donnez la valeure d'element %d:",i+1);

scanf("%d",&z);

\*(tab2+i)=z;

}

x=0;

for (i=0;i<y;i++)

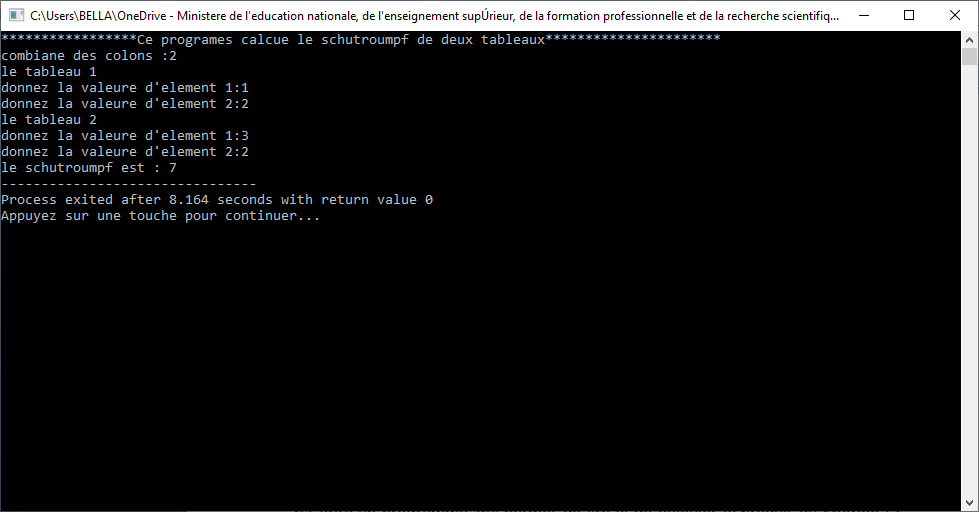
{

y=\*(tab1+i)\*\*(tab2+i);

x=x+y;

}

printf("le schutroumpf est : %d",x);

}

2.2&2.3 :déjà fait dans le dernier question

2.4:

I made a better one !!!!

Took me a week.

a program that calculate determinant of any matrix n\*n

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int\* take (int ,int \*);

void show (int , int \*);

int grandsome=0;

int calcule(int ,int \*,int );

int bestline( int ,int \* );

int main()

{

int p=0,MAX,i;

printf("This programe can calculate the determenent of any matrix n\*n\n");

printf("Give matrix dimension : n=");

scanf("%d",&MAX);

MAX=MAX\*MAX;

int /\*tab[MAX]\*/b=0;

int n=MAX;

int \*tab;

tab = (int\*)malloc(n \* sizeof(int));

take (MAX,tab);/\*take tableau input\*/

show(MAX,tab);/\*show the matrix\*/

b=calcule1(MAX,tab,p);

printf("Its determenent is :%d\n",b );

return 0;

}

int\* take (int MAX,int \*tab)

{

int i,j;

for (i=0;i<MAX;i++)

{

j=i/sqrt(MAX);

scanf("%d",&tab[i]);

}

return(tab);

}

void show(int MAX,int \*tab)

{

int j,i;

printf("Your matrix is :\n");

for ( i = 0; i < MAX; i++)

{

printf("%4d ",tab[i] );

for(j=sqrt(MAX)-1;j<MAX;j+=sqrt(MAX))

{

if(i==j)

printf("\n");

}

}

}

int calcule1(int i,int \*tabprim,int c)

{

int d=1;

for(c=0;c<sqrt(i);c++)

{

grandsome+=d\*tabprim[c]\*calcule(i,tabprim,c);

d=-d;

}

printf("\n");

return(grandsome);

}

int calcule(int i,int \*tabprim,int c)

{

int r=0,lastcalcule(int \*tab1);

int\* takefrom(int i,int \*tabprim,int \*tab1,int r,int c);

r=(sqrt(i)-1)\*(sqrt(i)-1);

int tab1[r];

takefrom(i,tabprim,tab1,r,c);

if(r>4)

{

int p;

int d=1;

for(p=0;p<sqrt(r);p++)

{

grandsome+=d\*tab1[c]\*calcule(r,tab1,p);

d=-d;

}

printf("\n");

return(grandsome);

}

else

{

return (lastcalcule(tab1));

}

}

int\* takefrom(int i,int \*tabprim,int \*tab1,int r,int c)

{

int t,j,d=1;

c=c+(int)sqrt(i);

for(j=(int)sqrt(i),t=0;j<i,t<r;j++,t++)

{

if(j!=c)

{

tab1[t]=tabprim[j];

}

else

{

t--;

c=c+(int)sqrt(i);

}

}

return(tab1);

}

int lastcalcule(int \*tab1)

{

int some,a,b,c,d;

a=tab1[1];

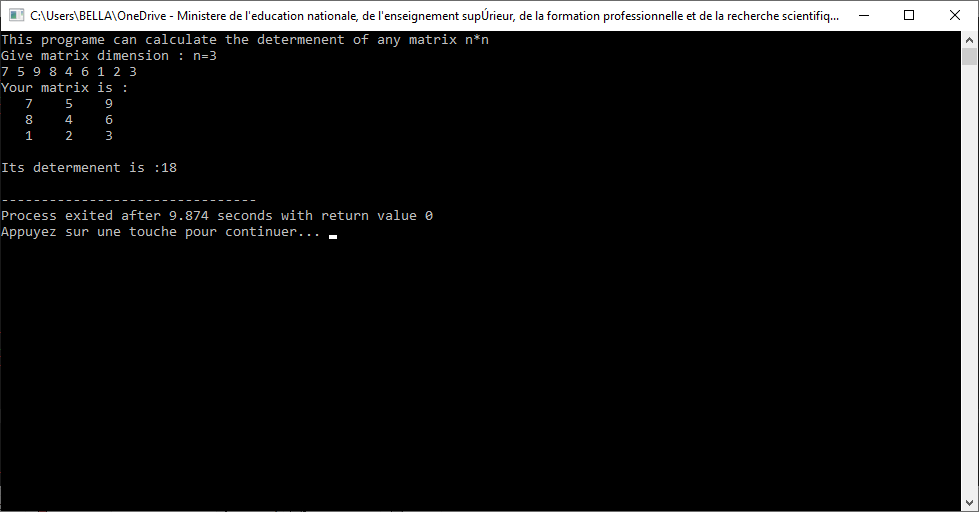
b=tab1[2];

c=tab1[3];

d=tab1[0];

some=d\*c-a\*b;

return(some);

}

EX3:

1-il n y a pas un tableau dans ce quistion

2- il n y a pas un tableau dans ce quistion

3-4-5 :

#include<stdio.h>

#include<stdlib.h>

#define NP 3

int i;

struct s\_point

{

char x;

float y;

float z;

};

main()

{

struct s\_point hei(int i);

struct s\_point heli(struct s\_point x);

struct s\_point \*Courbe;

Courbe=(struct s\_point \*)malloc(NP\*sizeof(struct s\_point));

free(Courbe);

for(int i=0;i<NP;i++)

{

Courbe[i]= hei(i);

}

for(int i=0;i<NP;i++)

{

heli(Courbe[i]);

}

}

struct s\_point hei(int i)

{

struct s\_point Courbe1;

printf("\ndonnez le point %d : \n example: A(x,y)\n",i+1);

scanf("%c(%g,%g)",&Courbe1.x,&Courbe1.y,&Courbe1.z);

return(Courbe1);

}

heli(struct s\_point x)

{

printf("le %c a des cordone : \nx=%g\ny=%g",Courbe[i].x,Courbe[i].y,Courbe[i].z);

return 0;

}

**Exercice 9: Pointeurs et tableaux à plusieurs dimensions** :

Un tableau à deux dimensions est, par définition, un tableau de tableaux. Il s'agit donc en fait

d'un pointeur vers un pointeur. Pour créer avec un pointeur de pointeur une matrice à k lignes

et n colonnes à coefficients entiers , on écrit :

**main(){**

**int k, n;**

**int \*\*tab;**

**tab = (int\*\*)malloc(k \* sizeof(int\*));**

**for (i = 0; i < k; i++)**

**tab[i] = (int\*)malloc(n \* sizeof(int));**

**....**

**for (i = 0; i < k; i++)**

**free(tab[i]);**

**free(tab);}**

Reprenez les exercices de la feuille TP N°4 (Tableau deux dimension) et remplacez les

matrices statiques par une allocation dynamique

déjà fait dans le dernier question

**Exercice 10 :** Soit le code

**main(){**

**char msg[]="Bonjour !";**

**char \*p;**

**for (p=msg; \*p!=’\0’;p++) printf("%c",\*p);**

**}**

Que fait ce programme ?

Reponse🡺

Ce programme écrire bonjour en utilisent de les pointuers.

Par exemple le B été enregistré dans l’adresse 1005x1

Donc automatiquement le « o » va enregistre dans l’dresse 1005x2 ;

Et ainsi de suit…

**Exercice 11:** Écrivez une fonction « **int\* concat\_tab(int n1,int t1[],int n2, int t2[])** » qui prend en

arguments deux tableaux et leurs tailles respectives, et qui renvoie leur concaténation. Il faudra allouer un

nouveau tableau pour contenir cette concaténation.

Reponse🡺

#include<stdio.h>

#include <stdlib.h>

int\* concat\_tab(int n1,int t1[],int n2, int t2[]){

int i;

int j;

int \*res = malloc((n1+n2)\*sizeof(int));

for (i = 0; i < n1; i++){

res[i]=t1[i];

printf("%d ",res[i]);

}

for (j=n1 ; j < n2+n1; j++){

res[j]=t2[j-n1];

printf("%d ",res[j]);

}

printf("\n");

}

int main(){

int L,N,i,j;

printf("donnez nombre des colone de premier tableux :");

scanf("%d",&L);

int tab1[L];

printf("donnez nombre des colone de deuxieme tableux :");

scanf("%d",&N);

int tab2[N];

printf("donnez les valeur de premier tableu:\n");

for (i = 0; i <L; i++) {

scanf("%d",&tab1[i]);

}

printf("donnez les valeur de deuxieme tableu:\n");

for (i = 0; i <L; i++){

scanf("%d",&tab2[i]) ;

}

printf("\*\*\*concaténation\*\*\*\n");

concat\_tab(L,tab1,N,tab2);

}

**Exercice 12 : 1)** Soit le programme :

**main(){**

**int i;**

**char \*chaine;**

**chaine = "chaine de caracteres";**

**for (i = 0; \*chaine != '\0'; i++)**

**chaine++;**

**printf("nombre de caracteres = %d\n",i);**

**}**

Quels résultats fournit par ce programme?

**2)** Ecrire un programme qui affiche le code ASCII de chaque lettre de votre prénom.

**3)** Ecrire un programme qui saisit successivement chaque caractère de votre prénom puis les

affiches.

Reponse🡺

#include<stdio.h>

#include<string.h>

void majuscule(char \*) ;

main()

{

int i=0;

char \*prenom;

prenom="Abdelouahab";

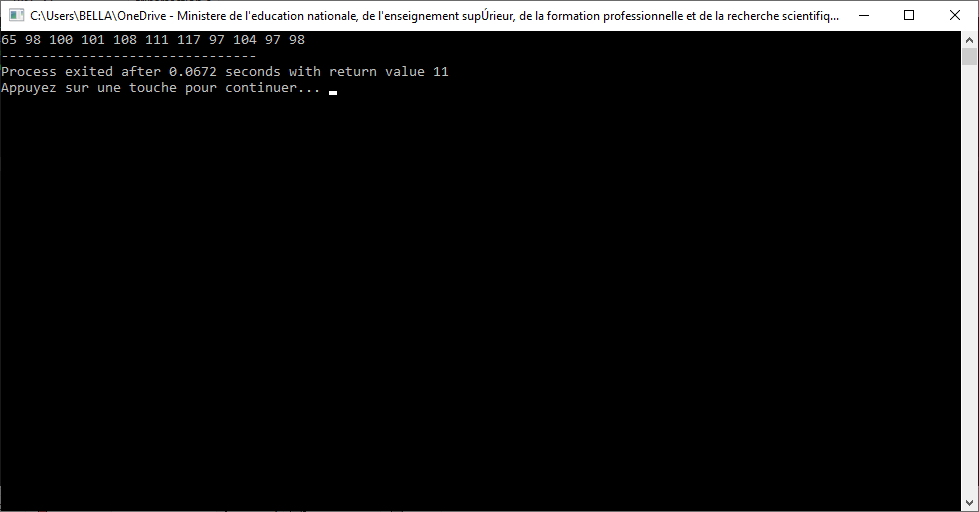
for(i=0;i<strlen(prenom);i++)

{

printf("%d ",\*(prenom+i));

}

}



#include<stdio.h>

#include<string.h>

#include<stdlib.h>

main()

{

int i=0,x;

char prenom[50];

for(i=0;;i++)

{

if(i==0)

printf("\nto end entre <.>\n");

printf("donnez le %d eme letter dans votre nom :",i+1);

scanf("%c",&\*(prenom+i));

getchar();

if(\*(prenom+i)==(int)'.')

break;

}

x=i;

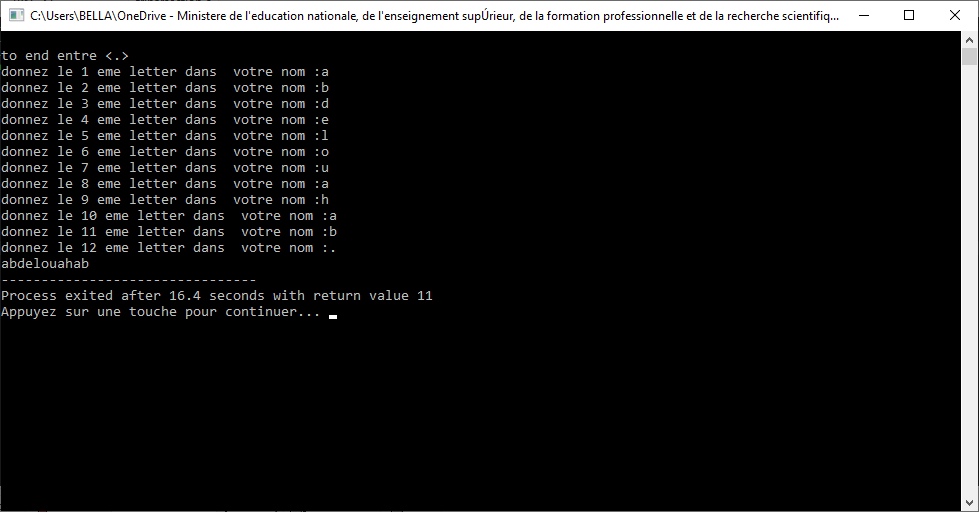
for(i=0;i<x;i++)

{

printf("%c",\*(prenom+i));

}

}



**Exercice 13 :** Soit le programme :

**#include <string.h>**

**main(){**

**int i;**

**char \*chaine1, \*chaine2, \*res, \*p;**

**chaine1 = "chaine ";**

**chaine2 = "de caracteres";**

**res = (char\*)malloc((strlen(chaine1) + strlen(chaine2)) \* sizeof(char));**

**p = res;**

**for (i = 0; i < strlen(chaine1); i++)**

**\*p++ = chaine1[i];**

**for (i = 0; i < strlen(chaine2); i++)**

**\*p++ = chaine2[i];**

**printf("%s\n",res);**

**}**

Que fait ce programme?

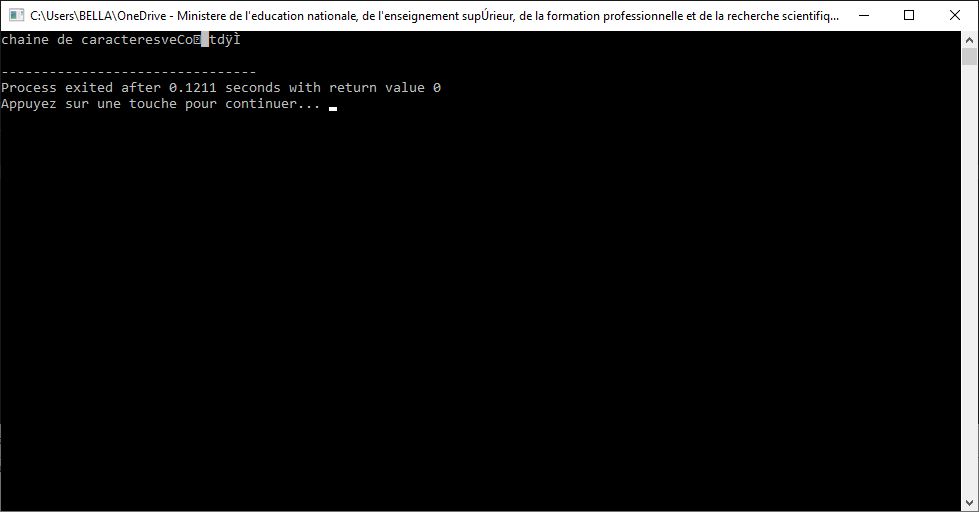
Ce programme

Déclare deux chaine de caractère

Et un tableau d’un longueur de les deux chaine

Et âpre ansière le tout dans le premier élément de tableau

Le résultat est :



**Exercice 14 :** Réécrire la fonction longueur (strln dans string.h) qui calcul la longueur d’une

chaîne de caractères.

Prototype : *int longueur(char \*)*

*#include<stdio.h>*

*#include<string.h>*

*main()*

*{*

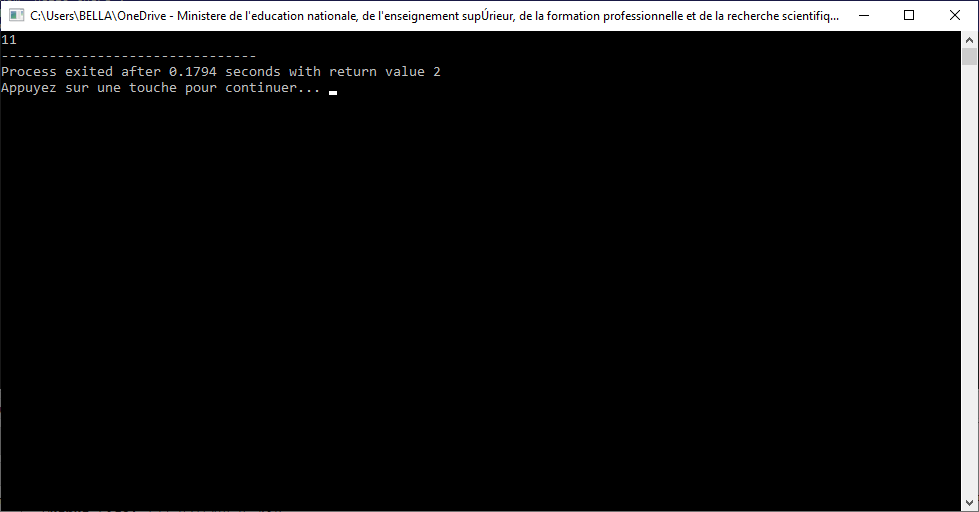
*int x;*

*char \*prenom="hello there";*

*x=strlen(prenom);*

*printf("%d",x);*

*}*

**

**Exercice 15 :** Soit le programme:

***#include<stdio.h>***

***void majuscule(char \*) ;***

***main(){***

***char chaine[] = "Ceci est une chaine !" ;***

***majuscule(chaine) ;***

***printf("%s\n",chaine) ;***

***}***

***void majuscule(char \*chaine){***

***int i=0;***

***while(chaine[i] != '\0'){***

***if ((chaine[i] >= 'a') && (chaine[i] <= 'z'))***

***chaine[i] += (int)'A' - (int)'a' ;***

***i++ ;***

***}***

Que fait ce programme?

Ce programme remplace les minuscule alphabète par les majuscule un , « en utilisent le tableau ASCII ».

**Exercice 16 :** Soit le programme:

**main(){**

**char ch1[50] = "bonjour" ;**

**char \* ch2 = " monsieur" ;**

**printf ("avant : %s\n", ch1) ;**

**strncat (ch1, ch2, 6) ;**

**printf ("après : %s", ch1) ;**

**}**

Que fait ce programme?

Cde programme déduire la rôle de la fonction strncat()

Sa fonction inséré la contenu de l’adresse ch2 dans les liber blancs dans le tableau ch1.

**Exercice 17 :** *Programmer les différentes méthodes de trie du cours*

*(dernier chapitre en algorithmique) et comparer leurs complexités.*

#include <stdio.h>

#define n 9

void tripar\_selection();

void tripar\_remplacment ();

void tripar\_insertion();

void tri\_a\_bulles();

void affiche();

void full\_table();

float table[n];

int i,j;

int main()

{

int h;

full\_table();

getchar();

printf("\nwhich method you want to use ?\n1-tri par remplacment\n2-tri par selection\n3-tri parinsertion\n4-tri a bulles\n enter ints number:");

scanf("%d",&h);

h==1?tripar\_remplacment():h==2?tripar\_selection():h==3?tripar\_insertion():h==4?tri\_a\_bulles():printf("invalide numbre");

affiche();

return 0;

}

void full\_table()

{

printf("fulle the table 3\*3\n");

for ( i = 0; i < n; i++)

{

scanf("%f",&table[i]);

}

}

void affiche()

{

for (i = 0; i < n; i++)

{

printf("%g ",table[i]);

}

}

void echangeV(float a,float b)

{

float c;

c=a;

a=b;

b=c;

}

void echangeA(float \*a,float \*b)

{

float c;

c=\*a;

\*a=\*b;

\*b=c;

}

void tripar\_selection()

{

for ( i = 0; i < n; i++)

{

for ( j = i+1; j < n; ++j)

{

if(table[j]<table[i])

{

echangeA(table+j,table+i);

}

}

}

}

void tripar\_remplacment ()

{

int a;

float maxt,max(),min(),tab1[n];

maxt=max();

for(i=0;i<n;i++)

{

a=min();

tab1[i]=table[a];

table[a]=maxt;printf("",a);

}

for(i=0;i<n;i++)

{

float c;

c=tab1[i];

tab1[i]=table[i];

table[i]=c;

}

}

float max()

{

float maxi;

for(i=0;i<n;i++)

{

if (maxi<table[i])

maxi=table[i];

}

return (maxi);

}

float min()

{

float min,mini;

for(i=0;i<n;i++)

{

if (min>table[i])

min=table[i];

}

for(i=0;i<n;i++)

{

if(table[i]==min)

mini=i;

}

return (mini);

}

void tripar\_insertion()

{

for(i=1;i<n;i++)

{

j=i-1;

while(j>=0&&table[j]>table[j+1])

{

echangeA(table+j,table+j+1);

j--;

}

}

}

void tri\_a\_bulles()

{

for(i=n;i>0;i--)

{

for(j=0;j<i-1;j++)

{

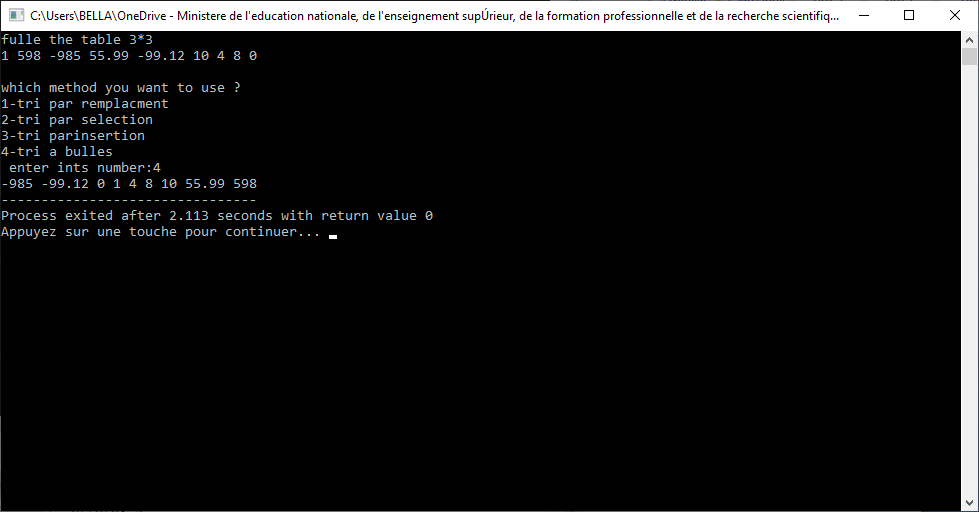
if(table[j]>table[j+1])

echangeA(table+j,table+j+1);

}

}

}



Le complexité :

Tri par remplacement<tri par sélection<tri par insertion < tri a bulles.