Name : Abdelouahab

Last name: BELLA

IDSD G1

**Exercice 0 : 1)** Ecrivez un programme permettant la saisie d’un tableau de réels, et un tableau de caractère. Le nombre des éléments du tableau doit être fixer par l’utilisateur.

**2)** Ecrivez un programme permettant la saisie d’un tableau à deux dimensions de réels, et un tableau à deux dimensions de caractère. Le nombre des éléments du tableau doit être fixer par

l’utilisateur.

Réponse🡺

#include<stdio.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[x];

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();

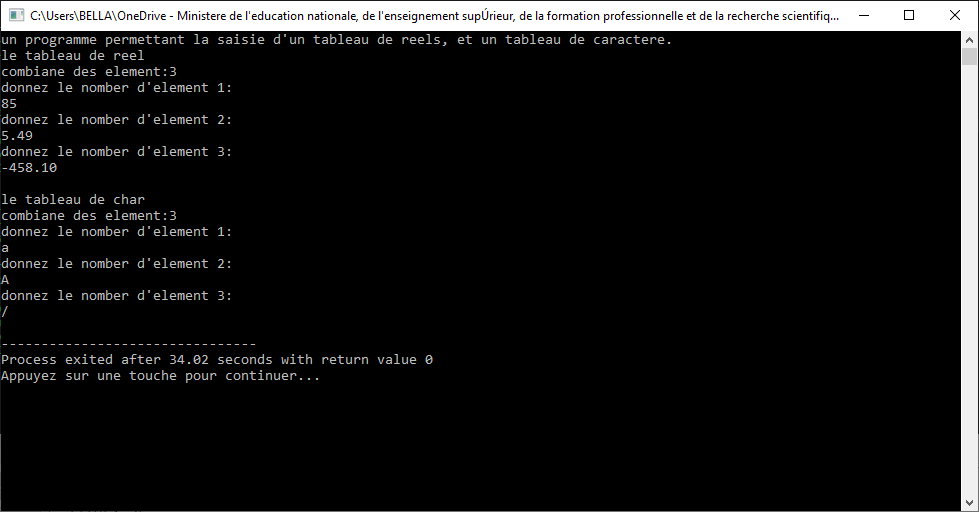
printf("donnez le nomber d'element %d:\n",i+1);

scanf("%c",&y);

tlab[i]=y;

}

}



#include<stdio.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:");

scanf("%d",&x);

float tlab[2][x];

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

{

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

scanf("%f",&y);

tlab[2][i]=y;

}

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

printf("combiane des element:");

getchar();

scanf("%d",&x);

char tab[2][x];

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

{

getchar();

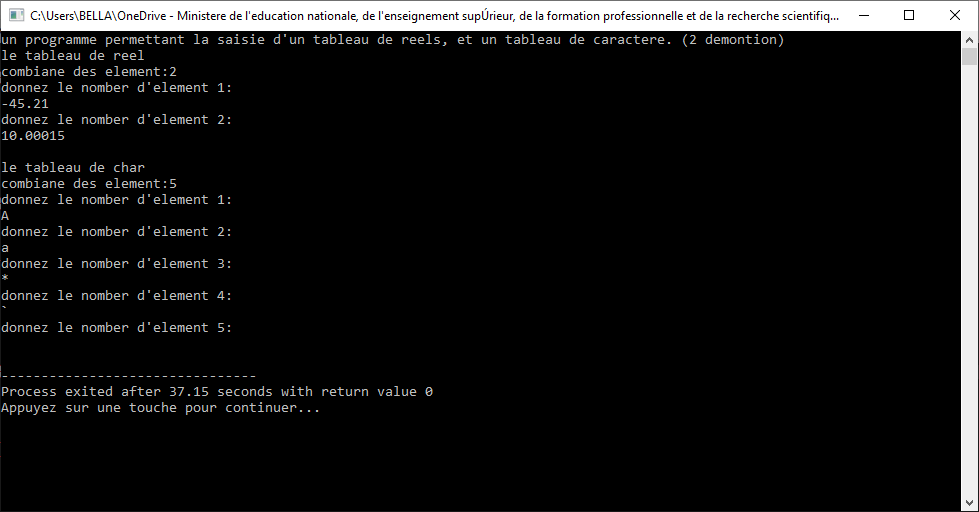
printf("donnez le nomber d'element %d:\n",i+1);

scanf("%c",&y);

tlab[2][i]=y;

}

}



**Exercice 1 :1)** Ecrire un programme permettant de lire et de calculer la somme, la

soustraction et la multiplication de deux Tableaux lus au clavier. (Afficher les résultats au

écran).

**2)** Ecrire un programme permettant de lire et de calculer la somme et la soustraction de deux

Matrices lus au clavier. (Afficher les résultats au écran).

**3)** Ecrire un programme qui, étant donnés deux tableaux A et B d’entiers de même

longueur N, détermine le nombre de positions où A[i] = B[i]..

Réponse🡺

#include<stdio.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[y];

float tab1[y];

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[y];

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);

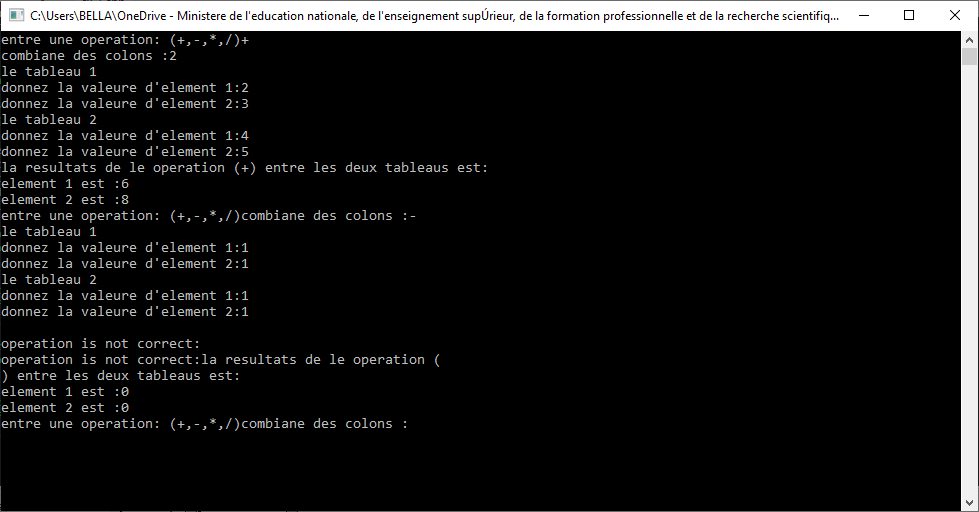
for(i=0;i<y;i++)

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

}

}

}



#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:{

if (Operator=='\*')

{

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);

}

else

printf("\nHow many lines and colons on both matrixes :");

int A[x][y];

int B[z][t];

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

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

{

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;j<z;j++)

{

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;j<x;j++)

{

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

{ l=0;

switch (Operator)

{

case '+':C[j][i]=A[j][i]+B[j][i];;break;

case '-':if (y!=t && x!=y) {printf("As I said the number of colons and lines should be the same ");goto obito;C[j][i]=A[j][i]-B[j][i];};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<z;q++,g++)

{

l=l+A[j][q]\*B[g][i];

}

C[j][i]=l;

}

;break;

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

}

printf(" %d",C[j][i]);

}

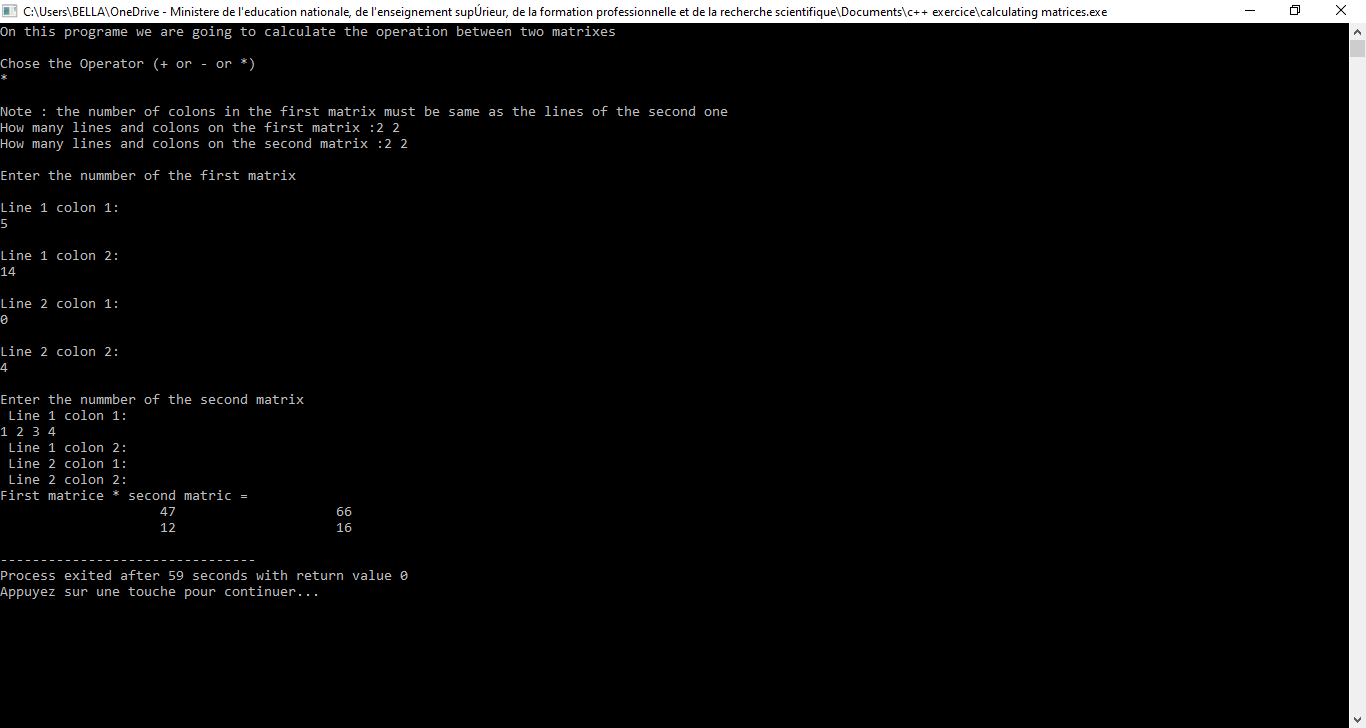
printf("\n");

}

}

}

}



#include<stdio.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 tlab[x],tab[x];

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;

}

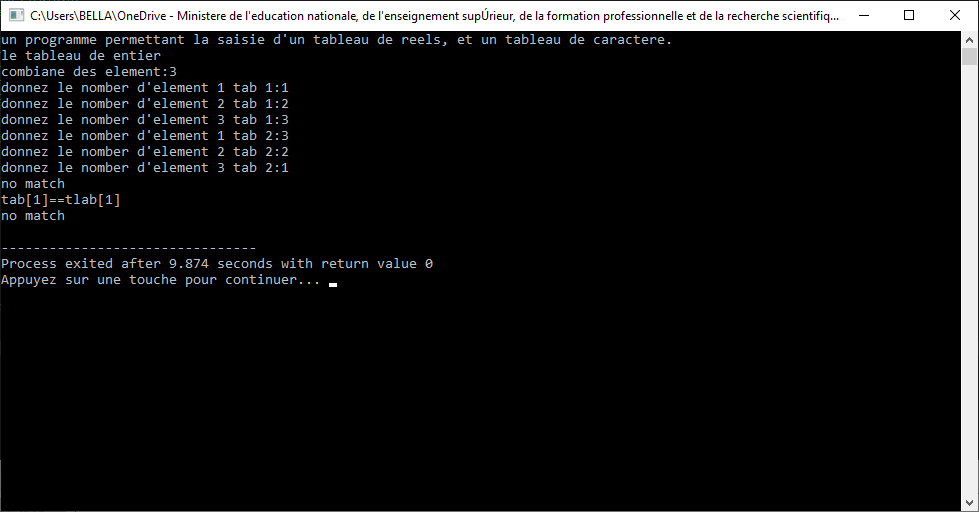
for(i=0;i<x;i++)

{

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

}

}



**Exercice 2 :1)** Ecrire un programme qui calcule le schtroumpf des deux tableaux. Pour

calculer le schtroumpf, il faut multiplier chaque élément du tableau 1 par chaque élément du

tableau 2, et additionner le tout.

**2**) Ecrire un programme qui permet de lire et de calculer le produit du Matrice (3 lignes ,3

colonnes) par un vecteur de trois elements.

**3)** Ecrire un programme qui calcule le produit du Matrice par un vecteur.

On rappelle que le produit Anla,nca Bnlb,ncb est une matrice Cnla,ncb définie par :

**4)** Saisir une matrice d’entiers 2x2, calculer et afficher son déterminant.

Réponse🡺

#include<stdio.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 tab1[y];

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[y];

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++)

{

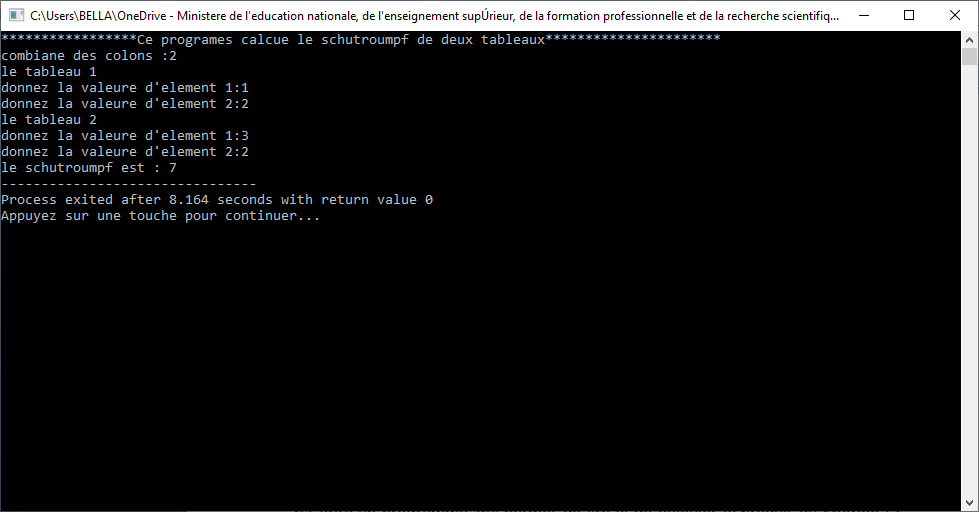
y=tab1[i]\*tab2[i];

x=x+y;

}

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

}



#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:{

if (Operator=='\*')

{

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);

}

else

printf("\nHow many lines and colons on both matrixes :");

int A[x][y];

int B[z][t];

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

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

{

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;j<z;j++)

{

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;j<x;j++)

{

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

{ l=0;

switch (Operator)

{

case '+':C[j][i]=A[j][i]+B[j][i];;break;

case '-':if (y!=t && x!=y) {printf("As I said the number of colons and lines should be the same ");goto obito;C[j][i]=A[j][i]-B[j][i];};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<z;q++,g++)

{

l=l+A[j][q]\*B[g][i];

}

C[j][i]=l;

}

;break;

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

}

printf(" %d",C[j][i]);

}

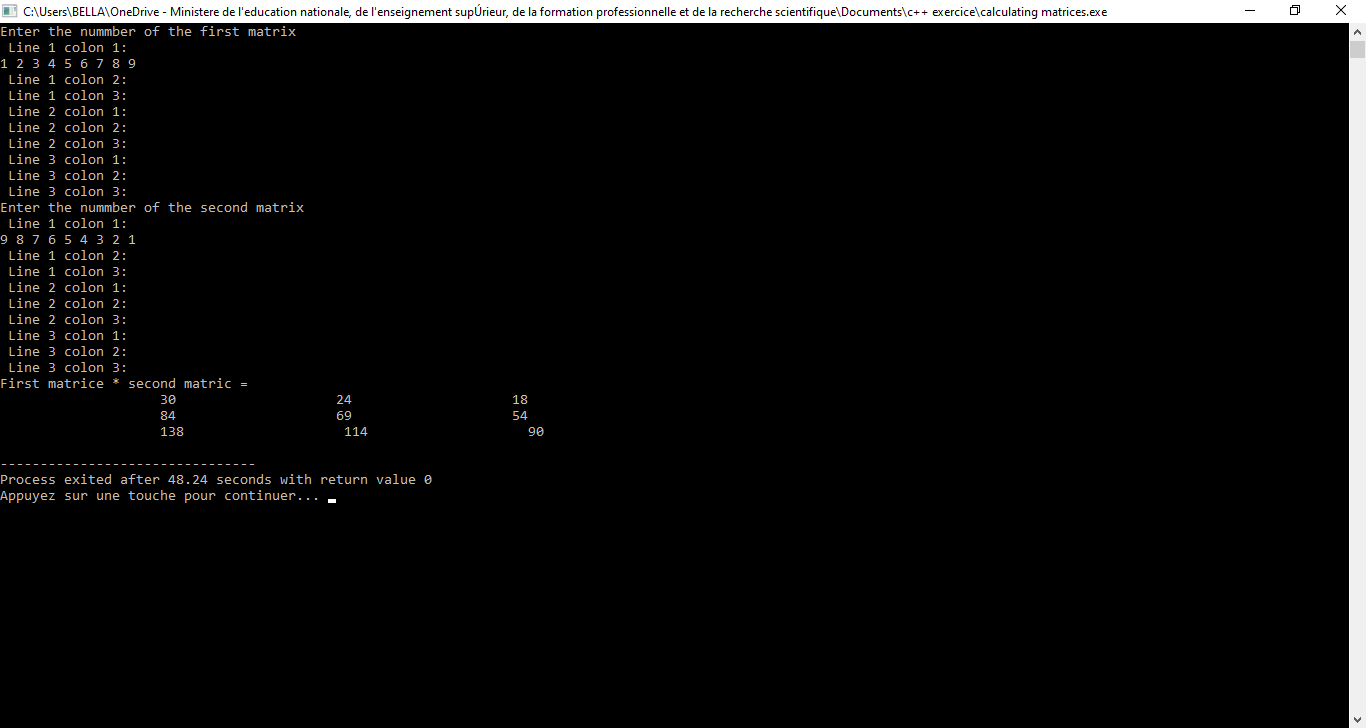
printf("\n");

}

}

}

}



int ex\_determinent\_matrice (void){

int det,x[4];

for(i=0;i<4;i++)

{

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

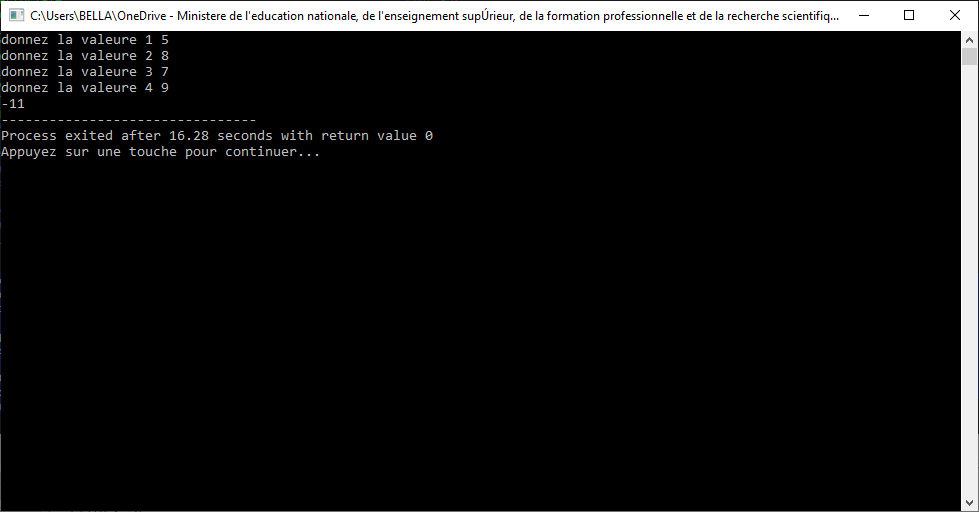
scanf("%d",&x[i]);

}

det=0;

det=x[0]\*x[3]-x[2]\*x[1];

return (det);



**Exercice 3: 1)** Réaliser une structure s\_point permettant de représenter un point dans l’espace. Chaque point sera caractérisé par un nom (de type char) et des coordonnées x, y (de type float).

**2)** Ecrire une fonction qui reçoit en argument une structure s\_point et qui affiche le contenu sous la forme

point B de coordonnées 10 12

1. a) En transmettant en arguments la valeur de la structure concernée.

Dans les deux cas, on écrira un petit programme d’essai de la fonction ainsi réalisée.

**3)** Ecrire une fonction qui met a zéros les différents champs d’une structure du type s\_point qui lui est transmise en argument. La fonction ne comportera pas de valeur de retour

**4)** Ecrire la déclaration d’un tableau (nommé Courbe) de NP points (NP supposé défini par une instruction #define)

**5)** Ecrire une fonction ( nommée affiche) qui affiche les valeurs « points » du tableau courbe transmis en argument, sous la forme

point D de cordonnées 10 2

réponse==>

#include<stdio.h>

main()

{

struct s\_point

{

char x;

float y;

float z;

};

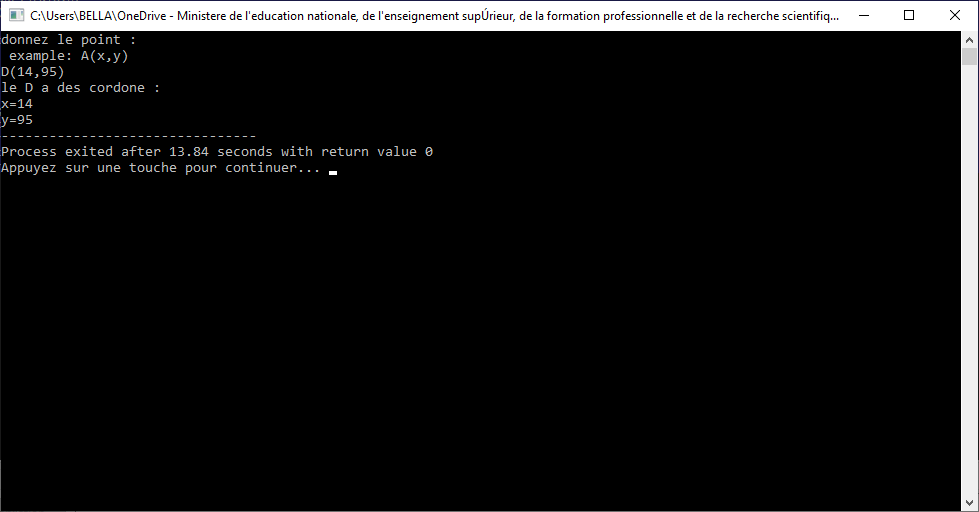
struct s\_point hi;

printf("donnez le point : \n example: A(x,y)\n");

scanf("%c(%f,%f)",&hi.x,&hi.y,&hi.z);

printf("le %c a des cordone : \nx=%f\ny=%f",hi.x,hi.y,hi.z);

}



2-

#include<stdio.h>

struct s\_point

{

char x;

float y;

float z;

};

main()

{

struct s\_point hei();

struct s\_point hi;

hi= hei();

printf("le %c a des cordone : \nx=%f\ny=%f",hi.x,hi.y,hi.z);

}

struct s\_point hei()

{

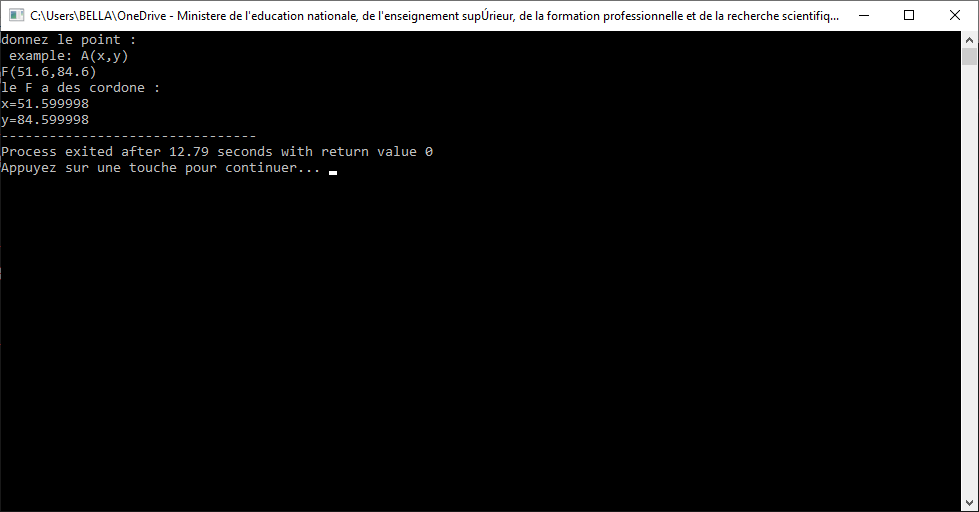
struct s\_point hi;

printf("donnez le point : \n example: A(x,y)\n");

scanf("%c(%f,%f)",&hi.x,&hi.y,&hi.z);

return(hi);

}



#include<stdio.h>

#define NP 3

int i;

struct s\_point

{

char x;

float y;

float z;

};

struct s\_point Courbe[NP];

main()

{

struct s\_point hei(int i);

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

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);

}

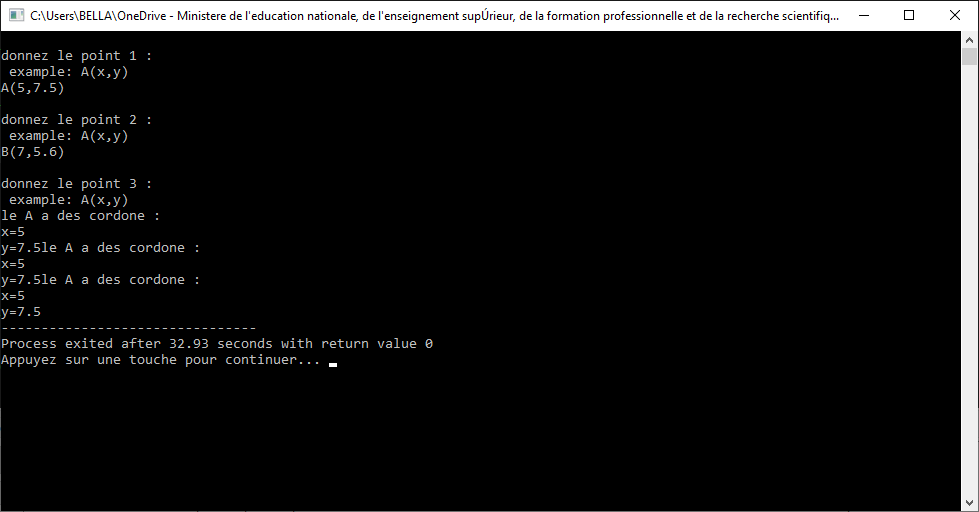
heli(struct s\_point x)

{

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

return 0;

}



**Exercice 4: 1)** Soit la structure suivante : typedef strut {

char titre ; unsigned long duree;

}chanson;

A- Définir le type "disque permettant de regrouper les informations suivantes : 1.l'auteur ;nom de variable "auteur" 2.l'année nom de variable :"annee" 3.Le titre du disque nom de variable :"titre" 4.la liste des chansons sur le disque nom de variable : liste 5.le nombre de chansons sur me disque ;nom de variable :nbchansons B- Déclarer un tableau de type disque (max=10) C- Remplir le tableau et afficher les éléments de ce tableaux sur l'écran D- Rechercher une chanson dans le tableau (la recherche s'effectue par le titre de la chanson )

et d'afficher ses informations si elle présente dans le tableau sinon message d'erreur

réponse🡺

#include<stdio.h>

#define NP 2

struct song

{

char auteur[64];

char titre[64];

int anne;

char list[64];

int nbchansons;

}hex[NP];

void focafich(struct song );

struct song foctake();

int main ()

{

int i;

char name;

int x;

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

{

hex[i]=foctake();

}

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

{

focafich(hex[i]);

}

printf("donnez le titre: ");

scanf("%s",&name);

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

{

if(\*hex[i].auteur==name)

{

focafich(hex[i]);

x=1;

break;

}

}

if(x=!1)

printf("le titre n\'existe pas dans la mimoire");

}

struct song foctake()

{

struct song atr;

printf("donnez l\'auteur:\n");

scanf("%s",&atr.auteur);

printf("donnez le titre:\n");

scanf("%s",&atr.titre);

printf("donnez le list:\n");

scanf("%s",&atr.list);

printf("donnez l\'anne:\n");

scanf("%d",&atr.anne);

printf("donnez le nomber de chasone:\n");

scanf("%d",&atr.nbchansons);

return(atr);

}

void focafich(struct song atr )

{

printf("l\'auteur:%s\n",atr.auteur);

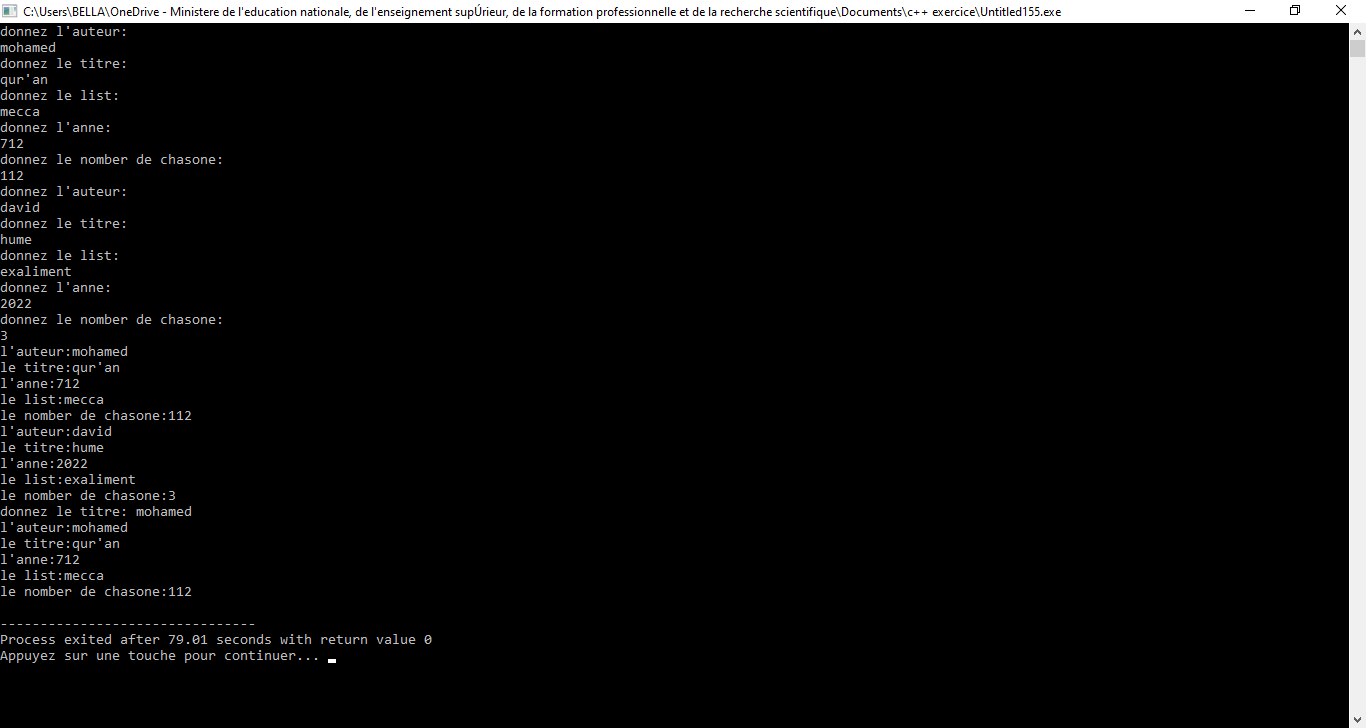
printf("le titre:%s\n",atr.titre);

printf("l\'anne:%d\n",atr.anne);

printf("le list:%s\n",atr.list);

printf("le nomber de chasone:%d\n",atr.nbchansons);

}



**Exercice 5:**On propose la structure de donnée suivante qui permet de représenter le réservoir d’une voiture par sa capacité maximale et la quantité de carburant y contenue.

struct reservoir\_voiture {

int capacite ; // capacité maximale

int quantité ; } // niveau de carburant

1. Créer un nouveau type nommé basé Reserve\_voiture sur la structure de donnée reservoir\_voiture.

2. Ecrire une méthode Réservoir(Reserve\_voiture voiture, entier Max) qui permet d’initialiser la capacité maximale Max du réservoir d’une voiture.

3. Ecrire une méthode entier verse(Reserve\_voiture voiture, entier Q) qui ajoute la quantité Q au réservoir si possible, sinon, on ne verse que ce qui est possible. La méthode fournit en résultat la quantité réellement ajoutée.

4. Ecrire une méthode entier puise(Reserve\_voiture voiture, entier Q) qui puise la quantité Q si possible, sinon, on puise le reste. La méthode fournit en résultat la quantité réellement puisée.

5. Ecrire une méthode entier jauge(Reserve\_voiture voiture) qui permet de fournir le niveau du réservoir.

6. Ecrire une fonction entier plein(Reserve\_voiture voiture) qui permet d’ajouter la quantité nécessaire au réservoir pour atteindre le plein. La méthode fournit en résultat la quantité nécessaire.

7. Les conducteurs s’intéressent au nombre de kilomètres parcourus par leurs voitures. Un conducteur a effectué plusieurs pleins de carburant en relevant chaque fois le nombre de kilomètres parcourus et le nombre de litres de carburant nécessaires pour les pleins.

Ecrire la fonction main() qui demande le kilométrage parcourus et le nombre de litres nécessaires lors de chaque plein sous la forme de valeurs entières, puis calcule et affiche le nombre de kilomètres parcourus par litre de carburant. La fonction doit afficher aussi les nombres de kilomètres par litre combinés jusqu’à ce point. Initialiser la capacité maximale du réservoir et la quantité initiale du carburant à la valeur 60.

#include<stdio.h>

#define max 60

struct reservoir\_voiture

{

int capacite ; // capacité maximale

int quantite ; // niveau de carburant

struct Reserve\_voiture

{

int Max;

int Q;

}doop;

int Q;

int jauge;

}loop;

int main()

{

int y,i,z,t;

int plein(int);

printf("combiane des pleines ?\n");

scanf("%d",&y);

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

{

printf("donnez le kilométrage parcourus avant le plein %d: ",i+1);

scanf("%d",&z);

printf("donnez le nombre de litres nécessaires pour le plein : ");

scanf("%d",&t);

}

}

int plein(int x)

{

if (x<loop.doop.Max)

loop.quantite=loop.quantite+loop.doop.Max-x;

loop.Q=loop.doop.Max-x;

}

**Exercice 1:** En utilisant la fonction, écrire un programme qui demande un nombre entier à

l'utilisateur, puis qui calcule et affiche le double de ce nombre?

#include<stdio.h>

int power(int number){

int db;

db=number \* number;

return db;

}

int main(){

int n,S;

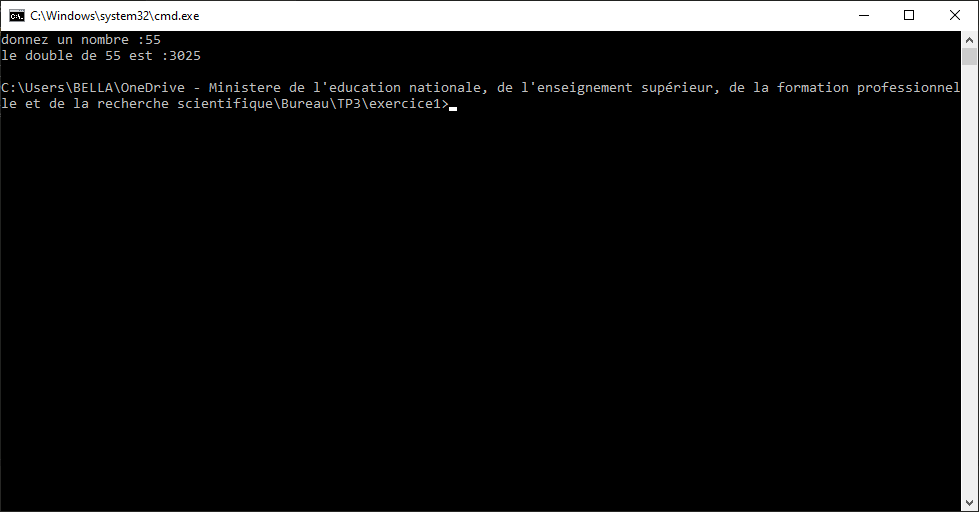
printf("donnez un nombre :");

scanf("%d",&n);

S=power(n);

printf("le double de %d est :%d\n",n,S);

}



**Exercice 2: a)** Écrire une fonction, nommée f1, se contentant d’afficher "bonjour" (elle ne

possèdera aucun argument ni valeur de retour),

**b)** Écrire une fonction, nommée f2, qui affiche "bonjour" un nombre de fois égal à la valeur

reçue en argument (int) et qui ne renvoie aucune valeur,

**c)** Écrire une fonction, nommée f3, qui fait la même chose que f2, mais qui, de plus, renvoie

la valeur (int) 0.

Écrire un petit programme appelant successivement chacune de ces trois fonctions, après les

avoir convenablement déclarées sous forme d’un prototype.

#include<stdio.h>

int f1(){

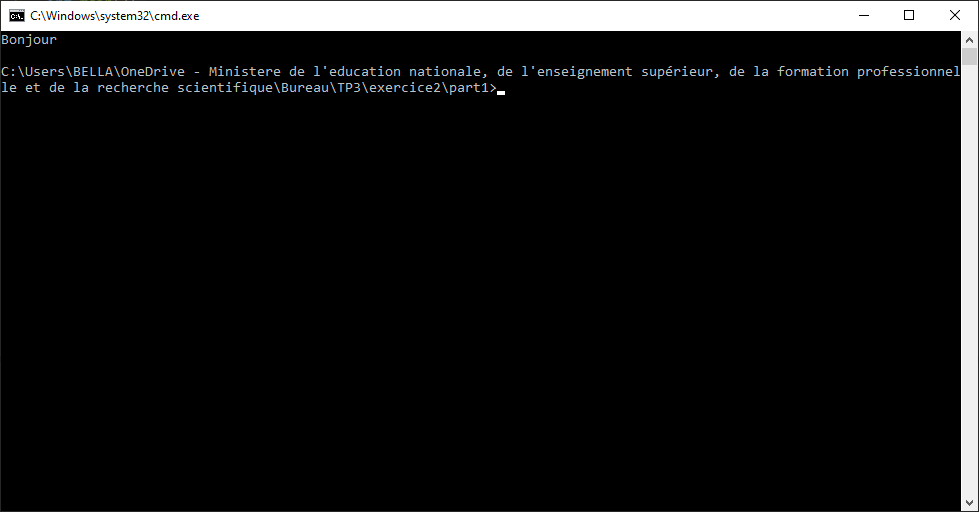
printf("Bonjour\n");

}

int main(){

f1();

}



2-

#include<stdio.h>

int f2(){

int i;

for(i = 0 ; i<4 ;i++){

printf("Bonjour\n");

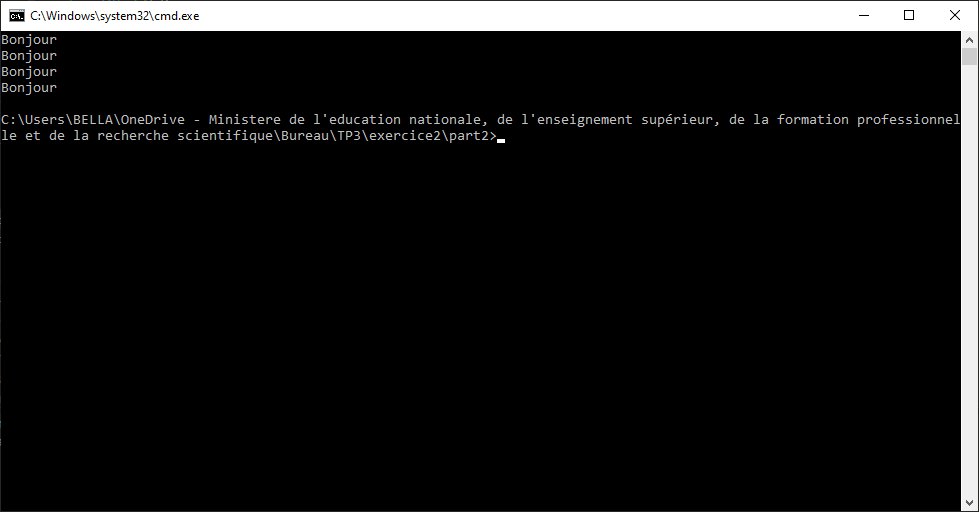
}

}

int main(){

f2();

}



3-

#include<stdio.h>

int f3(int fin){

int i;

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

printf("Bonjour\n");

}

}

int main(){

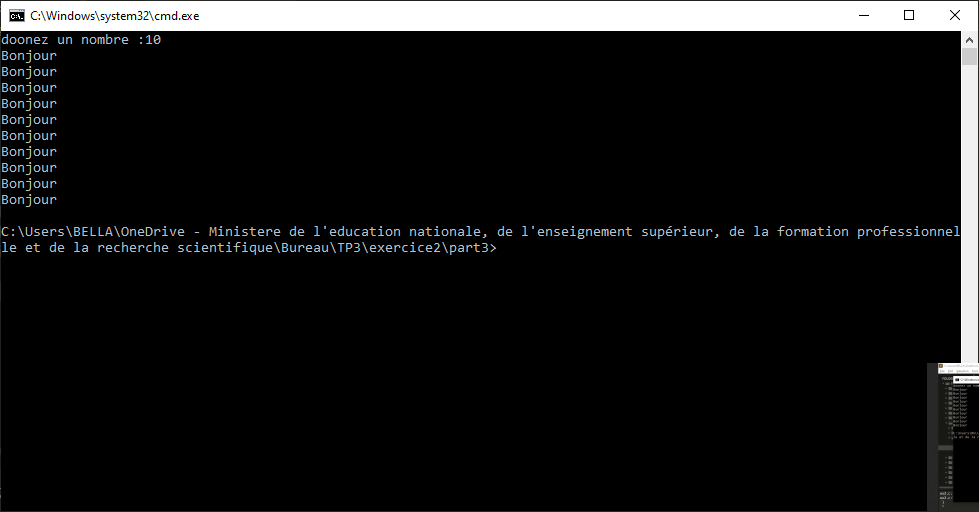
int n;

printf("doonez un nombre :");

scanf("%d",&n);

f3(n);

}

****

**Exercice 3:** Qu’affiche le programme suivant ?

int n=5 ;

main(){

void fct (int p) ;

int n=3 ;

fct(n) ;

}

void fct(int p){

printf("%d %d", n, p) ;

}

#include<stdio.h>

int n=5 ;

int main(){

void fct (int p) ;

int n=3 ;

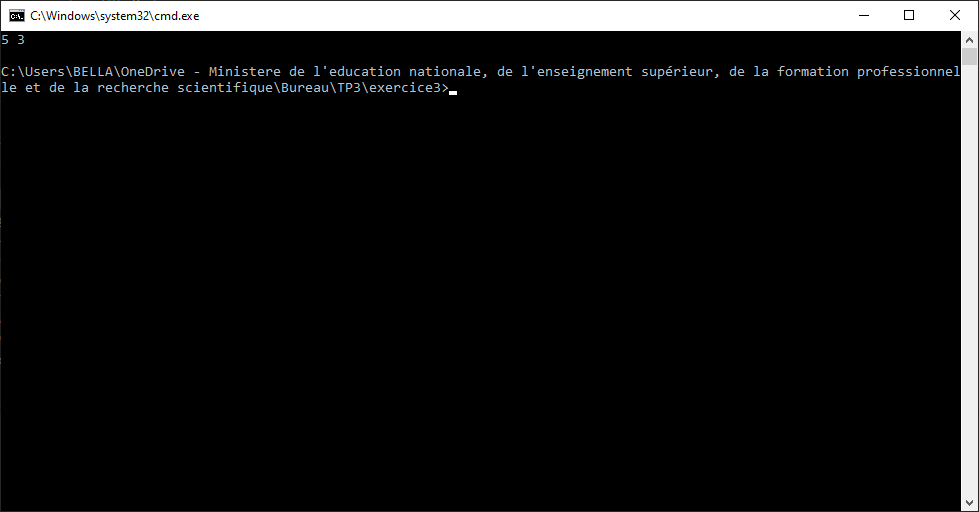
fct(n) ;

}

void fct(int p){

printf("%d %d\n", n, p) ;

}

****

**Exercice 4 :** Qu'affiche le programme suivant ? Expliquer.

void ma\_fonction( int x ) {

int y;

x = 0;

y = -1;

}

main() {

int x = 22;

int y = 23;

ma\_fonction(x);

printf( "x=%d y=%d\n", x, y ); }

#include<stdio.h>

void ma\_fonction( int x ) {

int y;

x = 0;

y = -1;

}

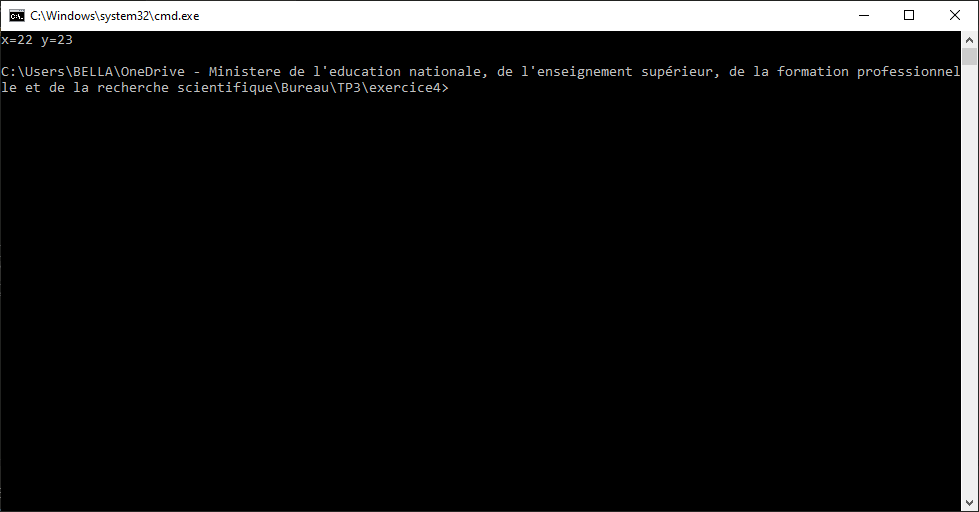
int main() {

int x = 22;

int y = 23;

ma\_fonction(x);

printf( "x=%d y=%d\n", x, y ); }

****

**Exercice 5:** Ecrire le programme principale utilisant la fonction suivante qui permet de

calculer et de retourner la plus grande valeur parmi deux réels.

float plusGrand(float x, float y)

{

if (x > y)

return x;

else

return y;

}

#include<stdio.h>

float plusGrand(float x, float y)

{

if (x > y)

return x;

else

return y;

}

int main(){

int a,b,res;

printf("donnez le nombre 1:");

scanf("%d",&a);

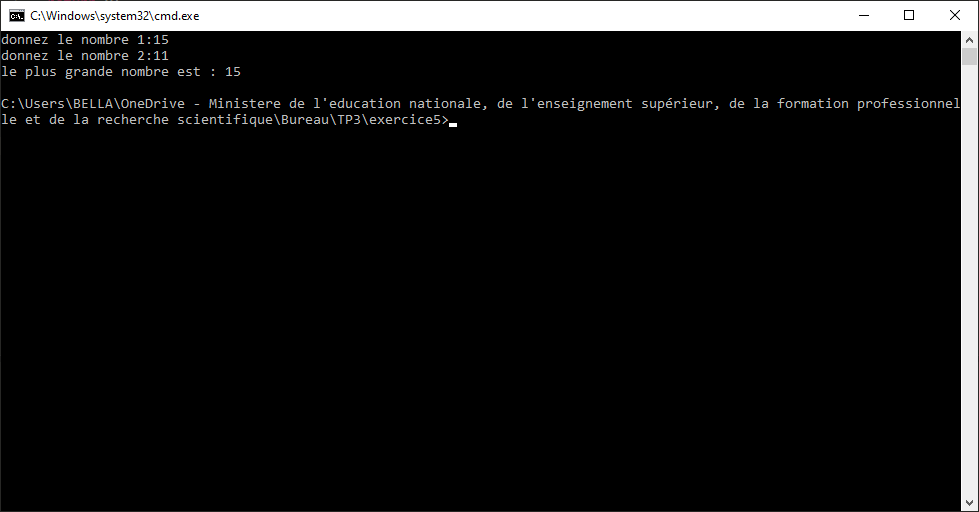
printf("donnez le nombre 2:");

scanf("%d",&b);

res=plusGrand(a,b);

printf("le plus grande nombre est : %d\n",res);

}

****

**Exercice 6: 1)** Ecrire une fonction somme\_div qui retourne la somme des diviseurs d’un

nombre passé en paramètre

**2)** Ecrire une fonction parfait qui teste si un nombre passé en paramètre est parfait et qui

retourne 1 s’ il est et 0 sinon.

**3)** Ecrire un programme principal qui affiche tous les nombres parfaits inférieurs à une

certaine limite

1-

#include<stdio.h>

int somme\_div(int number){

int i;

for(i=1; i <=number;i++){

if(number%i==0){

printf("%d\n",i);

}

}

}

int main(){

int n;

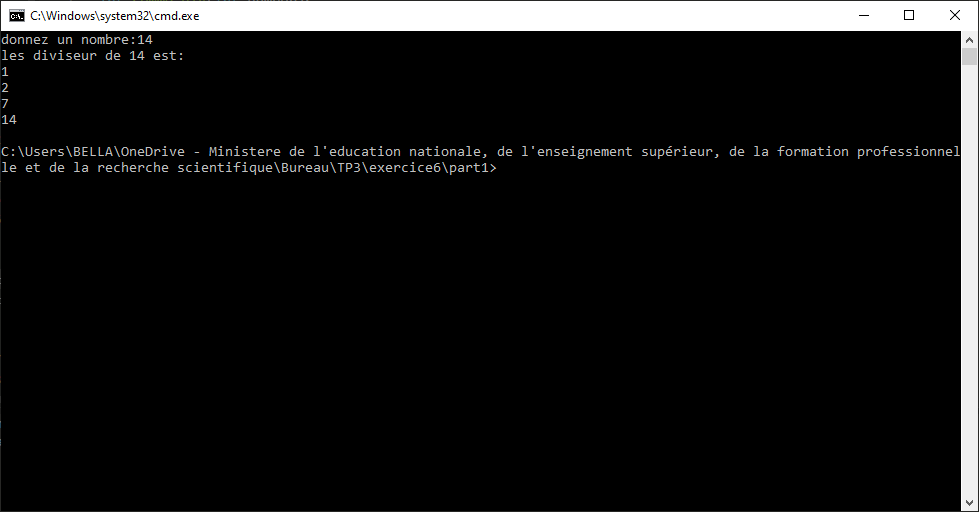
printf("donnez un nombre:");

scanf("%d",&n);

printf("les diviseur de %d est:\n",n);

somme\_div(n);

}



2-

#include<stdio.h>

int somme\_div(int number){

int i,S=0;

for(i=1; i <number;i++){

if(number%i==0){

S+=i; }

}

if (S==number){ return 1; }

else{return 0;}

}

int main(){

int n,S;

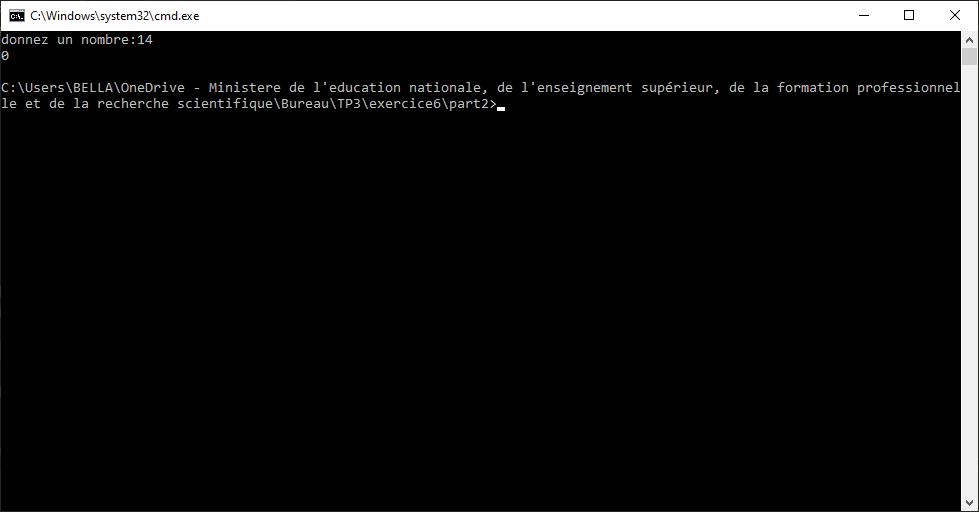
printf("donnez un nombre:");

scanf("%d",&n);

S=somme\_div(n);

printf("%d\n",S);

}



3-

#include<stdio.h>

int somme\_div(int number){

int i,S=0;

for(i=1; i <number;i++){

if(number%i==0){

S+=i; }

} return S;

}

int main(){

int n,S;

printf("les nombre parfait sont:");

for (n = 0; n < 10000; n++){

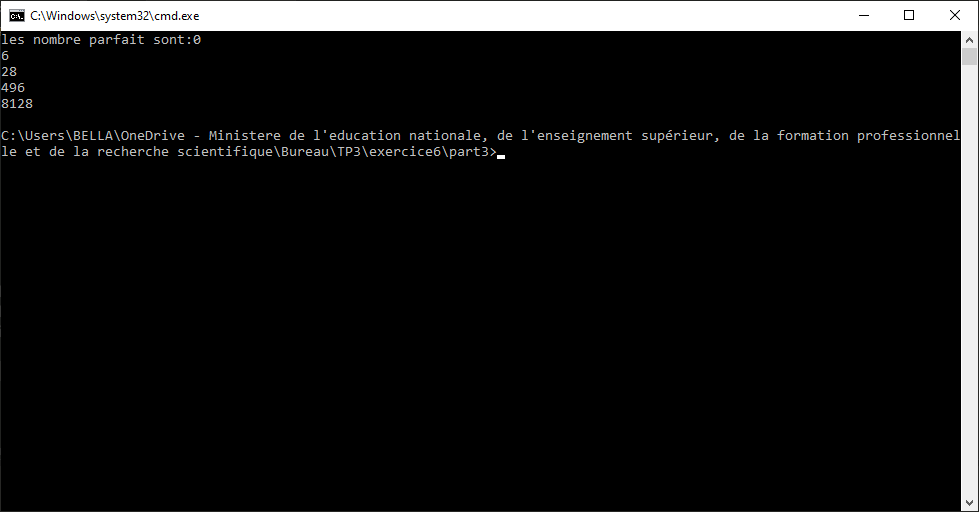
if (somme\_div(n)==n){

S=somme\_div(n);

printf("%d\n",S);}

}

}

****

**Exercice 7 :** (Fonctions qui renvoient ou affichent un résultat.)

Ecrire un programme qui définit et utilise :

- une fonction **fact(n)** qui renvoie la factorielle du nombre n.

- une fonction **affiche\_fact(n)** qui ne renvoie rien et affiche la factorielle du

nombre n.

- une fonction **comb (int n , int p)** qui renvoie le nombre de combinaisons de p elements

parmi n

- une fonction **estDivisible( a, b)** qui renvoie 1 si a est divisible par b

- une fonction **estPremier(n)** qui renvoie 1 si a est premier

**Rappels :**

-factorielle(n) = n ! = n (n-1) (n-2) ...1.

-un nombre entier n est dit "premier" s'il n'existe aucun entier d dans

l'intervalle [2,n] tel que n soit divisible par d.

#include<stdio.h>

//calculer la factorielle d'un nombre//

int fact(int n){

int i,S=1;

for (i = n; i >1; i--){

S\*=i;

}

return S;

}

//calculer la factorielle d'un nombre (no retourne)//

void affiche\_fact( int n){

int i,S=1;

for (i = n; i >1; i--){

S\*=i;

}

printf("le factorielle de %d est : %d\n",n,S);

//renvoie le nombre de combinaisons//

}

int comb (int n , int p){

int S;

S= fact(n) / (fact (p) \* fact (n-p));

return S;

}

// a divisible par b ou no //

int estDivisible (int a, int b){

if (a%b == 0){

return 1;}

else{

return 0;}

//qui renvoie 1 si a est premier//

}

int estPremier(int n){

int i ;

for ( i =2; i < n-1; i = i + 1)

if ( estDivisible(n,i )){

return 0;

}

return 1;

}

//partie de utilisateur//

int main (){

int n,p,r;

printf("donnez un nombre 1:");

scanf("%d", &n);

printf("entrez le nombre 2:");

scanf("%d", &p);

affiche\_fact (n);

r=fact(n);

printf ("la factorielle de %d est :%d\n",n,r);

printf ("le nombre de faccons de prendre %d valeurs parmi %d est: %d \n", p,n,comb(n,p));

printf ("%d\n",estDivisible(n,p));

if (estPremier( comb(n,p))){

printf ("ce nombre est premier\n");

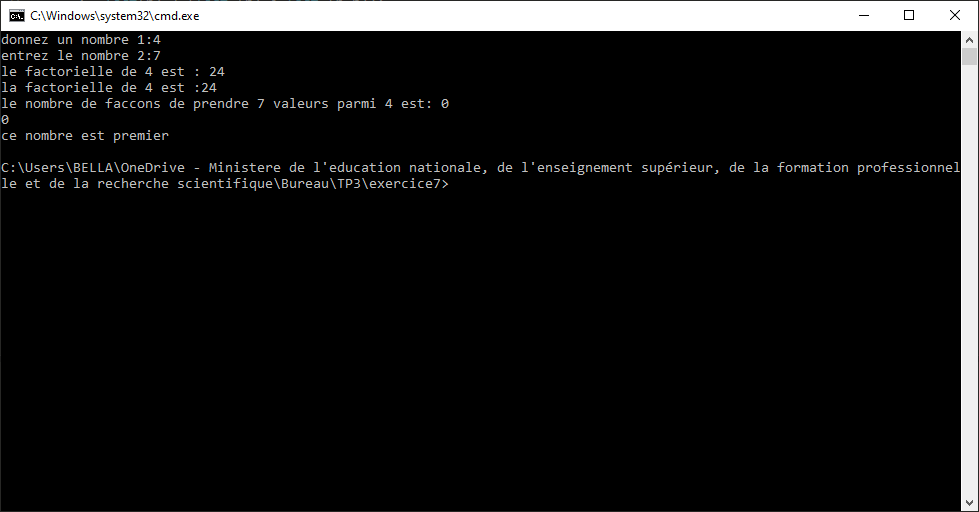
}

else{

printf ("ce nombre n’est pas premier\n");

}

}

****

**Exercice 8 : -** Écrire une fonction DivisionEuclidienne qui donne le quotient et le reste de la

division de 2 nombres entiers. On utilisera les opérateurs / et % du langage C.

#include<stdio.h>

int DivisionEuclidienne(int a,int b){

int quo,res;

quo=a/b;

res=a%b;

printf("le quotient = %d\n",quo);

printf("le reste = %d\n",res);

}

int main(){

int a,b,res,quo;

printf("donnez le nombre a:");

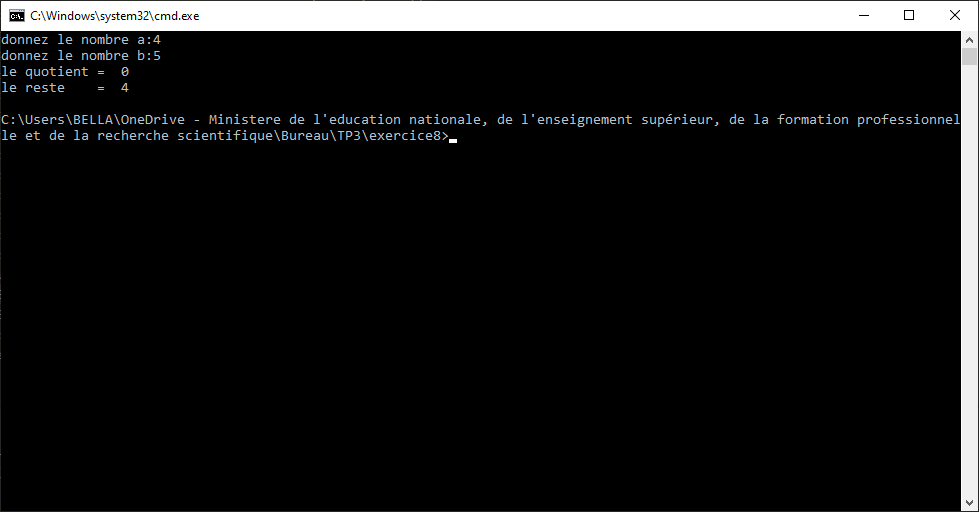
scanf("%d",&a);

printf("donnez le nombre b:");

scanf("%d",&b);

DivisionEuclidienne(a,b);

}

****

**Exercice 9 :** La fonction rand retourne un entier compris entre 0 et RAND\_MAX de manière

aléatoire. La constante RAND\_MAX ainsi que le prototype de la fonction rand sont définis

dans le fichier stdio.h.

Ecrire un programme qui initialise de manière aléatoire un entier entre 0 et 100 et propose de

le faire deviner à l’utilisateur. Pour ce faire, a chaque tentative de l’utilisateur, le programme

lui indique si sa proposition est plus petite ou plus grande que la valeur de l’entier recherché.

#include <stdio.h>

#include <stdlib.h>

int RandomValue(int n){

int randomValue = rand()%100;

while (n!=randomValue){

if (n>randomValue){

printf("plus petit:");

scanf("%d",&n);

}

else if (n<randomValue){

printf("plus grande:");

scanf("%d",&n);}

}

printf("merci \n");

}

int main(){

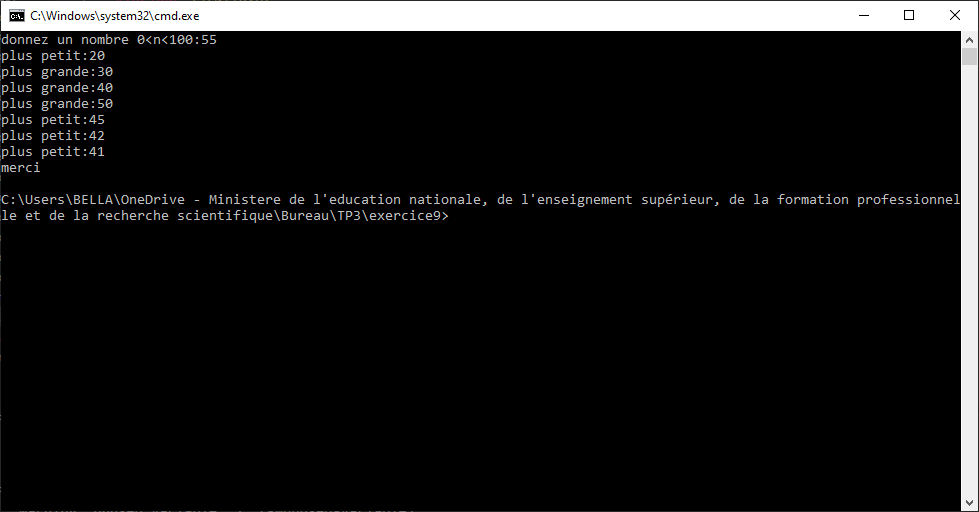
int n,S;

printf("donnez un nombre 0<n<100:");

scanf("%d",&n);

RandomValue(n);

}

****

**Exercice 10 :** Ecrire une fonction calculant la moyenne des éléments d’un tableau de réel :

float moyenne (float tab[], int dim){…}

#include<stdio.h>

float moyenne (float tab[], int dim){

int i;

float moy,S=0;

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

S+=tab[i];

}

moy=S/dim;

return moy;}

int main(){

int elem,i;

float moy;

printf("doonez les nombre des elements du tableau:");

scanf("%d",&elem);

float tab[elem];

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

printf("doonez le nombre %d:",i+1);

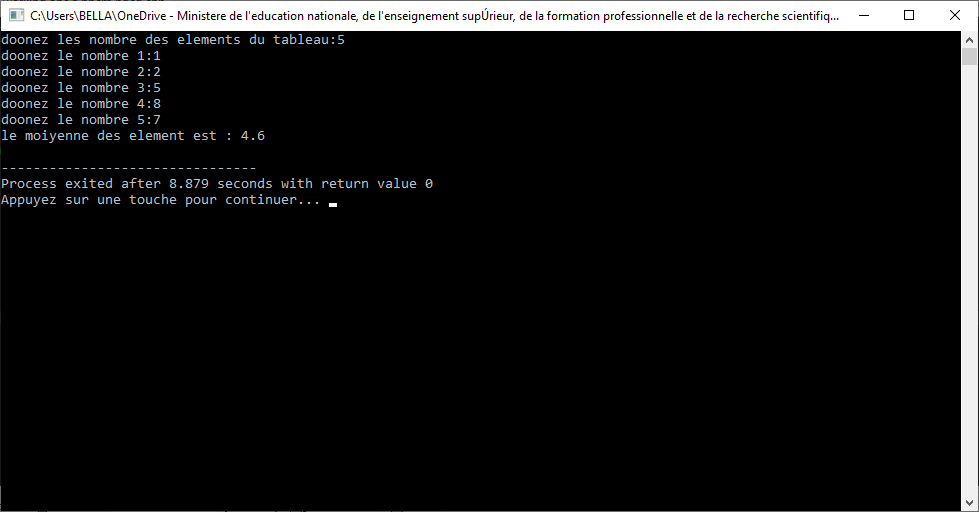
scanf("%f",&tab[i]);

}

moy=moyenne(tab,elem);

printf("le moiyenne des element est : %g\n",moy);

}

****

**Exercice 11 :** (Premier nombre premier plus grand que n) .

1. Ecrire une fonction premier d'un paramètre entier m et retournant TRUE

si le nombre est premier et FALSE dans le cas contraire.

2. Ecrire une fonction prochain premier prenant un paramètre entier n et

retournant le plus petit nombre premier plus grand ou égal à n.

3. Ecrire un programme qui demande un entier n à l'utilisateur et affiche le

premier nombre premier plus grand ou égal à n.

#include <stdlib.h>

#include <stdio.h>

#define TRUE 1

#define FALSE 0

int premier(int m){

int i ;

for(i = 2; i < m; i = i + 1){

if ((m % i) == 0)

return FALSE;}

return TRUE;}

int prochain\_premier(int n){

while(!premier(n)){

n = n + 1;

}

return n;}

void main(){

int n;

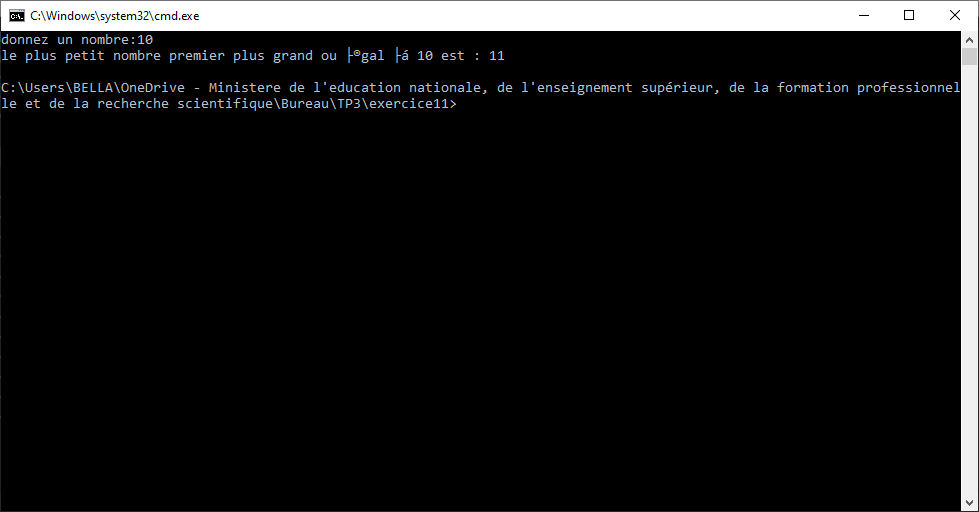
printf ("donnez un nombre:");

scanf("%d", &n);

premier(n);

printf("le plus petit nombre premier plus grand ou égal à %d est : %d\n",n, prochain\_premier(n));

}

****

**Exercice 12: 1)** Ecrire un programme qui calcule l’intégrale d’une fonction de type ax2 + bx

+ c entre deux points x1 et x2. Le programme devra demander à l’utilisateur les coefficients

a,b et c ainsi que les bornes x1 et x2 .

**2)** Ecrire une fonction qui calcule et renvoie la valeur du polynôme P(X), représenté par le

tableau de ses coefficients, pour une valeur donnée de la variable X.

#include<stdio.h>

#include<math.h>

float intergrale(float a,float b,float c,float x){

float d;

d=(a/3)\*pow(x,3)+(b/2)\*pow(x,2)+c\*x;

return (d);

}

int main(){

float a,b,c,x1,x2;

float s=0;

printf("donnez le debute de borne ");

scanf("%f",&x1);

printf("donnez le fin de borne ");

scanf("%f",&x2);

printf("donnez le nombre a:");

scanf("%f",&a);

printf("donnez le nombre b:");

scanf("%f",&b);

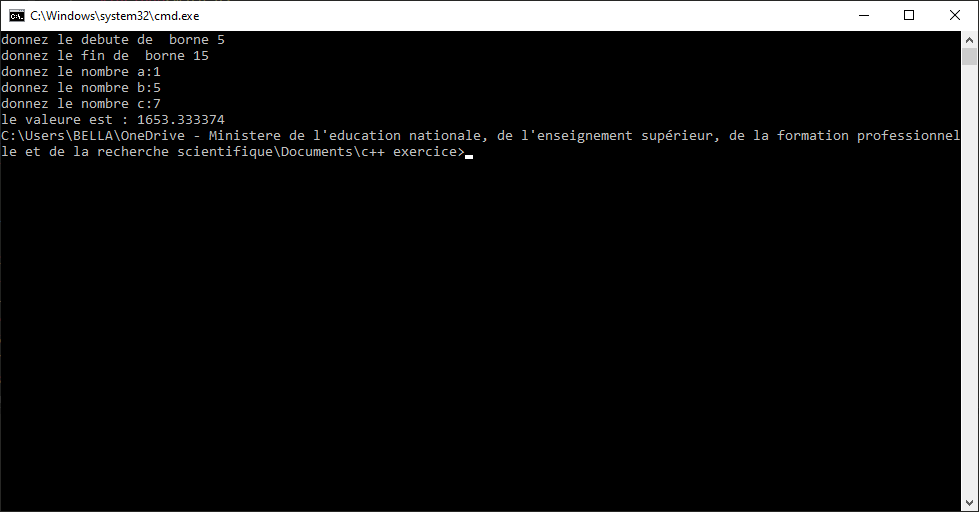
printf("donnez le nombre c:");

scanf("%f",&c);

s=intergrale(a,b,c,x2)-intergrale(a,b,c,x1);

printf("le valeure est : %f",s);

}



2-

#include<stdio.h>

int polynome(float A[20],int N,int X){

int I;

float P;

for (P=0.0, I=0 ; I<=N ; I++){

P = P\*X + A[N-I];

}

return P;

}

int main(){

int N;

float X,A[20];

float P;

printf("Entrer le degré N du polynôme (max.20) : ");

scanf("%d", &N);

printf("Entrer la valeur X de l'argument : ");

scanf("%f", &X);

for (int I=0 ; I<=N ; I++){

printf("Entrer le coefficient A%d : ", I);

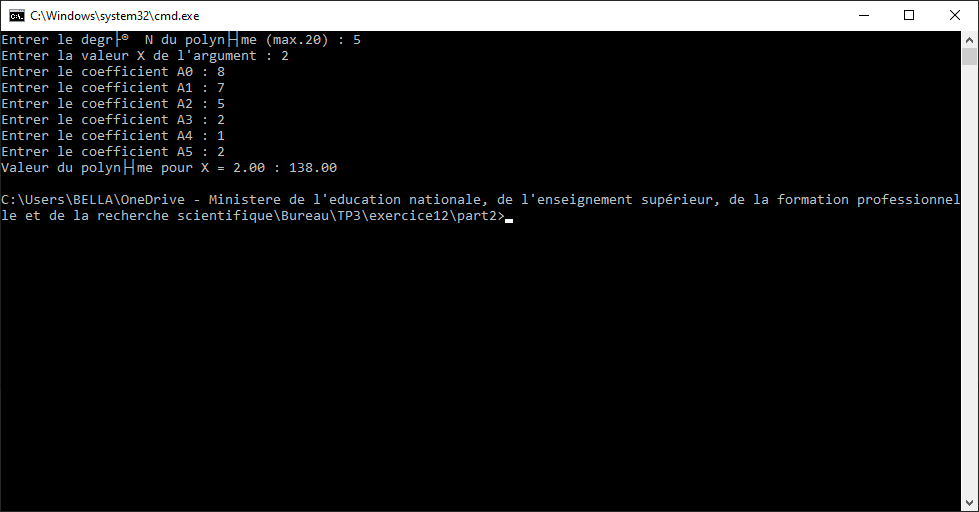
scanf("%f", &A[I]);

}

P=polynome(A,N,X);

printf("Valeur du polynôme pour X = %.2f : %.2f\n", X, P);

}

****

**Exercice 13 :** (Calcul d'une intégrale par la méthode des trapèzes)

1. Ecrire une fonction f qui calcule l'inverse 1=x de son argument.

2. Ecrire une fonction I de trois paramètres a, b et n qui calcule une valeur approchée de

l'intégrale de la fonction f entre a et b en utilisant la méthode des trapèzes :

I = ((b –a)/n)\*( f(x0)/2 +f(x1)+…+f(xn)/2) avec xi = a+(b-a)\*(i/n)

3. Ecrire un programme qui calcule les valeurs approchées de I pour a = 1

et b = 2 et n = 5; 10; 20; 50; 100 et compare ces valeurs avec la valeur

théorique (ln 2).

Note : Pour obtenir ln x il faut appeler log(x) dont le prototype se trouve dans math.h.

#include <stdlib.h>

#include<stdio.h>

#include <math.h>

float f(float x){

return 1/x;

}

float I(float a, float b, int n){

float s;

int i ;

s = (f(a) + f(b))/2;

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

s = s + f(a + (b-a)\*i/n);

return s\*(b-a)/n;

}

void approcheLn2(int n){

float x, y, res;

x = I(1, 2, n);

y = log(2);

res = (x-y) / y \* 100;

res = (res >= 0) ? res : res;

printf ("N = %d : I = %g L’erreur avec ln2 est %g%%\n", n, x, res);

}

void main(){

approcheLn2(5);

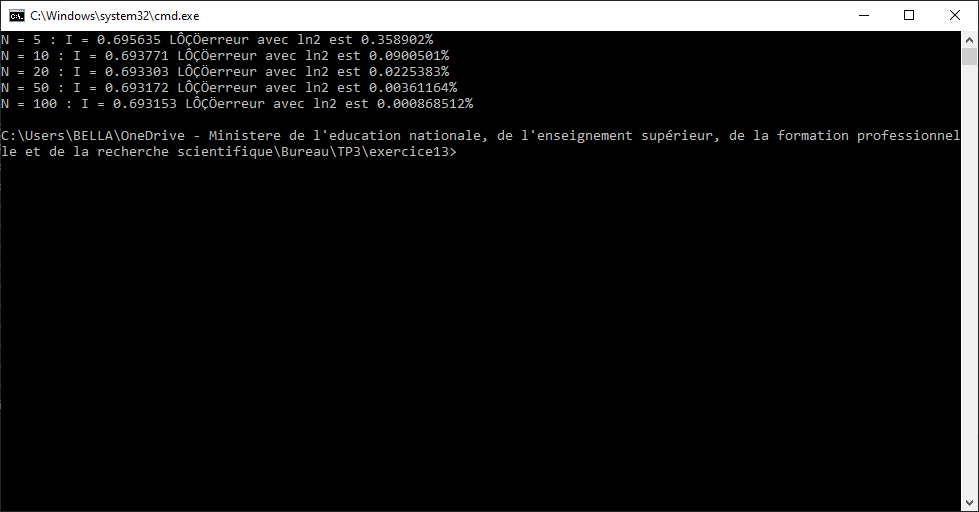
approcheLn2(10);

approcheLn2(20);

approcheLn2(50);

approcheLn2(100);

}

****

**Exercice 14:** Écrire une fonction récursive calculant la valeur de la « fonction d’Ackermann

» A définie

pour m>0 et n>0 par :

A(m,n) = A(m-1,A(m,n-1)) pour m>0 et n>0

A(0,n) = n+1 pour n>0

A(m,0) = A(m-1,1) pour m>0.

#include <stdio.h>

#include <stdlib.h>

int Ackermann(int X,int Y) {

if(X == 0) return Y+1;

else {

if(Y == 0) return Ackermann(X-1,1);

else return Ackermann(X-1,(Ackermann(X,Y-1)));

}

}

int main()

{

int i,n,j=0;

for(i=1;i<=2;i++) for(n=1;n<=10;n++) {

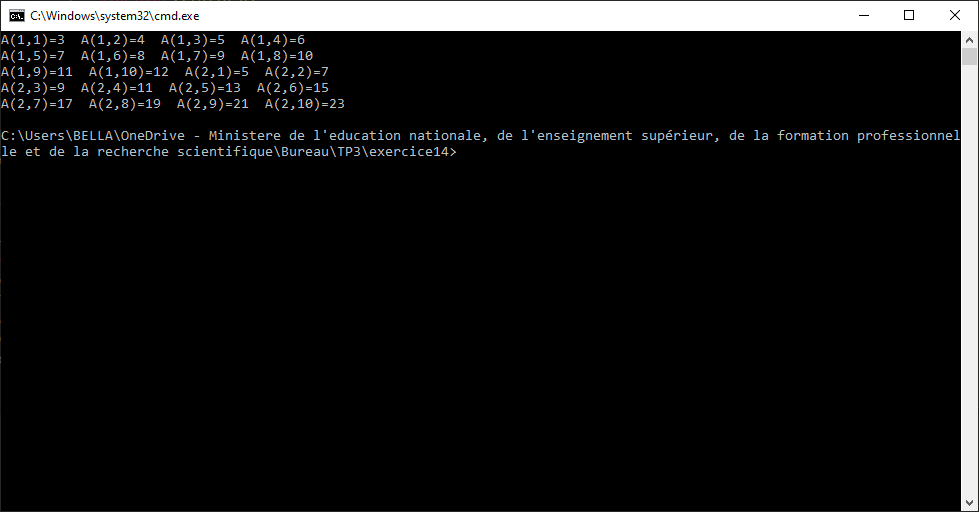
printf("A(%i,%i)=%i ",i,n,Ackermann(i,n));

j++;

if(j==4){printf("\n");j=0;}

}

}



**Exercice 15 :** 1- Écrire une fonction qui échange le contenu de deux variables entières.

Exemple d'utilisation :

… int a = 10, b = 22;

Echanger( &a, &b );

/\* ici a == 22 et b == 10 \*/

2- En utilisant la fonction Echanger, écrire une fonction Permute3 qui effectue une

permutation circulaire de trois variables, comme dans l'exemple ci-dessous :

... int a = 10, b = 22, c = 33;

Permute3( &a, &b, &c ); /\* ici a == 33, b == 10, c == 22 \*/

#include <stdio.h>

int Echange(int a, int b){

int temp=a;

a = b;

b = temp;}

int main() {

int x,y,z,m;

printf("Entrez le premier entier: ");

scanf("%d", &x);

printf("Entrez le deuxieme entier: ");

scanf("%d", &y);

printf("Entrez le troisieme entier: ");

scanf("%d", &z);

Echange(x,z);

Echange(z,y);

printf(" fonction Echange:\n\tx = %d\n\ty = %d\n\tz = %d\n", z, x,y);}

