**Exercise 4:**

#include<stdio.h>

void main(){

    int nbs[3], max, min, sum = 0, prod = 1, i=0;

    printf("Entrez trois nombres : ");

    scanf("%i%i%i", nbs, nbs + 1, nbs + 2);

    max = min = nbs[0];

    for (;i<3; i++){

        if(max < nbs[i])

            max = nbs[i];

        if(min > nbs[i])

            min = nbs[i];

        sum += nbs[i];

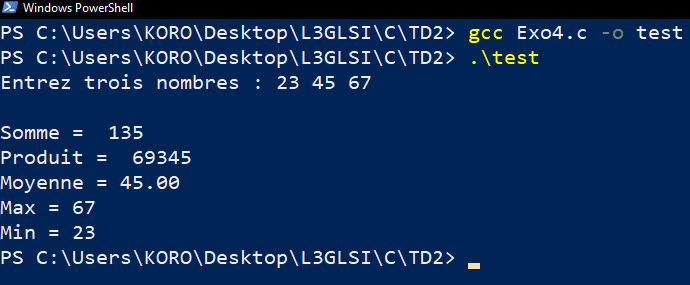
        prod \*= nbs[i];

    }

    float avg = sum / 3.0;

    printf("\nSomme =  %i\nProduit =  %i\nMoyenne = %1.2f\nMax = %i\nMin = %i\n", sum, prod, avg, max, min);

}

****

**Exercise 5:**

***Alternative 1 :***

#include<stdio.h>

void main(){

    int nbr = 0, i = 4, chiffre[5];

    do{

        printf("Entrez un entier de 5 chiffres : ");

        scanf("%i", &nbr);

    }while (nbr < 9999 ||  nbr > 99999);

    for(;nbr > 0;i--){

        int q = nbr%10;

        chiffre[i] = q;

        nbr /= 10;

    }

    for(i =0;i<5;i++)

        printf("%i , ", chiffre[i]);

}

***Alternative 2:***

#include<stdio.h>

#include<string.h>

void main(){

    char nombre[100];

    do{

        printf("Entrez un nombre de 5 chiffres : ");

        scanf("%s", nombre);

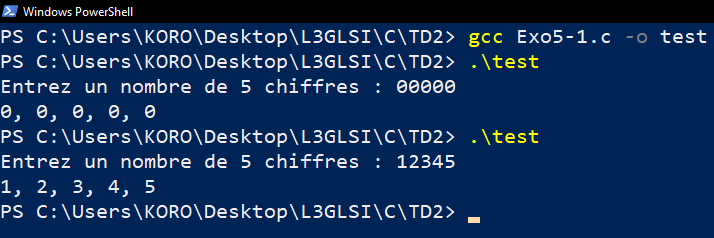
    }while (strlen(nombre) > 5 || strlen(nombre) < 5);

    for(int i = 0; i < strlen(nombre) - 1; i++)

        printf("%i, ", (int)nombre[i] - 48);

        printf("%i\n", (int)nombre[4] - 48);

}

****

**Exercise 6:**

void main(){

    printf("number   square   cube\n");

    for (int i=0; i < 11; i++)

        if(i < 3)

            printf("%i %8i %8i\n", i, i \* i, i \* i \* i);

        else if (i == 3)

            printf("%i %8i %9i\n", i, i \* i, i \* i \* i);

        else if (i == 4)

            printf("%i %9i %8i\n", i, i \* i, i \* i \* i);

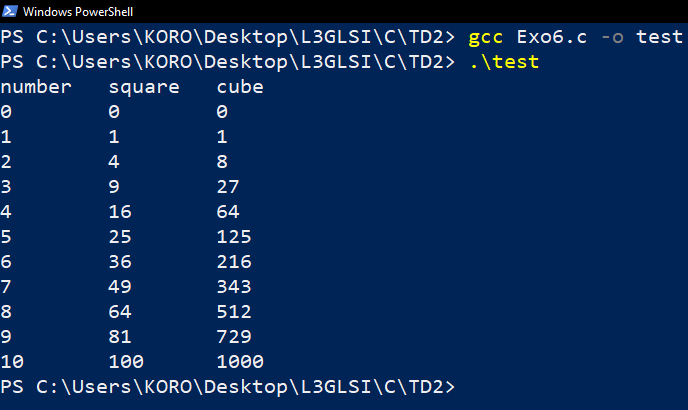
        else if (i > 4 && i < 10)

            printf("%i %9i %9i\n", i, i \* i, i \* i \* i);

        else if (i == 10)

            printf("%i %9i %9i\n", i, i \* i, i \* i \* i);

}

****

**Exercise 7:**

#include<stdio.h>

void main(){

    int nbs[10], max[2], min[2], i = 0;

    printf("Entrez 10 entiers : ");

    for(;i < 10;)

        scanf("%i", nbs + i++);

    max[0] = min[0] = nbs[0];

    for(i = 0;i < 10;i++)

        if(max[0] < nbs[i])

            max[0] = nbs[i];

        else if(min[0] > nbs[i])

            min[0] = nbs[i];

    max[1] = min[0];

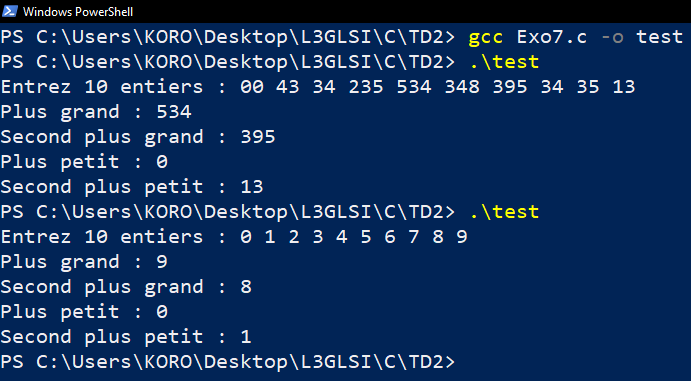
    min[1] = max[0];

    for(--i;i >= 0; i--)    if(max[1] < nbs[i] && nbs[i] != max[0]) max[1] = nbs[i];

    for(++i; i < 10; i++)   if(min[1] > nbs[i]) if(nbs[i] != min[0]) min[1] = nbs[i];

    printf("Plus grand : %i\nSecond plus grand : %i\nPlus petit : %i\nSecond plus petit : %i\n", max[0], max[1], min[0], min[1]);

}



**Exercise 13:**

#include<stdio.h>

void main(){

    int puissance = 1;

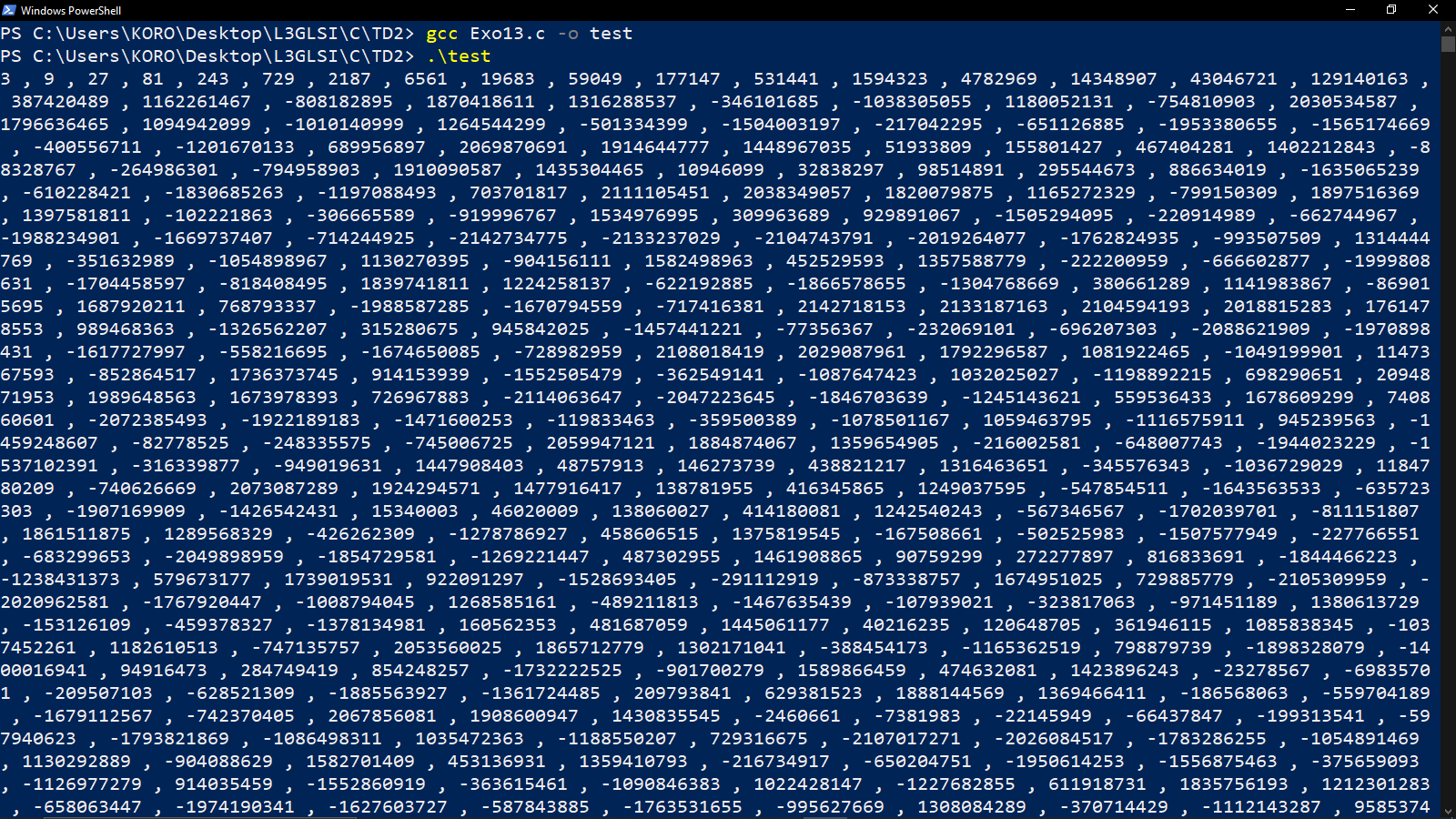
    while (puissance){

        puissance \*= 3;

        printf("%i , ", puissance);

    }

}

****

Le programme produit une boucle “infine” ne devant afficher que les puissances de 3, mais les types ayant des tailles limités, il un dépacement de capacité d’oules valeurs négatives.

**Exercise 14:**

***Alternative 1:***

#include<stdio.h>

int crypting(int n){

    int chiffres[4], i = 4;

    for(;n;i--){

        chiffres[i-1] = n % 10;

        n /= 10;

    }

    for(; i < 4;i++)    chiffres[i] = (chiffres[i] + 7) % 10;

    for(i = 0; i < 2;i++){

        int ech = chiffres[i];

        chiffres[i] = chiffres[i+2];

        chiffres[i+2] = ech;

    }

    for(i = 0; i < 4; i++)  n = n \* 10 + chiffres[i];

    return n;

}

int decrypting(int n){

    int chiffres[4], i=4;

    for(;n;i--){

        chiffres[i-1] = n % 10;

        n /= 10;

    }

    for(; i < 4;i++)    chiffres[i] += 10 - 7;

    for(i = 0; i < 2;){

        int ech = chiffres[i];

        chiffres[i] = chiffres[i+2];

        chiffres[i++ +2] = ech;

    }

    for(i = 0; i < 4; i++)  if(chiffres[i] >= 9)    chiffres[i] %= 10;

    for(i = 0; i < 4; i++)  n = n \* 10 + chiffres[i];

    return n;

}

void main(){

    int nombre, i;

    do{

        printf("Entrez un entier de 4 chiffres : ");

        scanf("%i", &nombre);

    }while (nombre < 999 || nombre > 9999);

    i = crypting(nombre);

    printf("%i\n", i);

    i = decrypting(i);

    printf("%i\n", i);

}

***Alternative 2:***

#include<stdio.h>

#include<string.h>

void crypting(int \* toEncrypt){

    for(int i = 3; i >= 0; i--)   toEncrypt[i] = (toEncrypt[i] + 7) % 10;

    for(int i = 3; i > 1; i--){

        int ech = toEncrypt[i];

        toEncrypt[i] = toEncrypt[i-2];

        toEncrypt[i-2] = ech;

    }

}

void decrypting(int \* crypted){

    for(int i = 3; i >= 0; i--) crypted[i] += 10 - 7;

    for(int i = 0; i < 2; i++){

        int ech = crypted[i];

        crypted[i] = crypted[i+2];

        crypted[i+2] = ech;

    }

    for(int i = 0; i < 4; i++)  if(crypted[i] >= 10) crypted[i] -= 10;

}

void main(){

    char nombre[100];

    int chiffres[4], i = 0;

    do{

        printf("Entrez un nombre de 4 chiffres : ");

        scanf("%s", nombre);

    }while (strlen(nombre) > 4 || strlen(nombre) < 4);

    for(;i < strlen(nombre);i++) chiffres[i] = (int)nombre[i] - 48;

    crypting(chiffres);

    for(i = 0; i < 4 ; i++) printf("%i", chiffres[i]);

    printf("\n");

    decrypting(chiffres);

    for(i = 0; i < 4 ; i++) printf("%i", chiffres[i]);

}

