Level: 1st year of computer science
Course: ADS1

Typical solution
TD/TP N°: 05

Academic year:2023/2024
Chapter 4 : Loops

Exercise 1: TD/TP

Write an algorithm with its C program that calculates the factorial of a number.

```
Algorithm factorial
Var n, f, i: integer
begin
 write("enter a nbr: ")
 read(n)
 f←1
 for i←2 to n do
  f←f*i
 end for
 write(n,"!=",f)
end.
```

```
#include <stdio.h>
int main() {
 int n, f, i;
 printf("enter a nbr: ");
 scanf("%d", &n);
 f=1;
 for(i=2; i<=n; i++)
  f*=i ;
 printf("%d!= %d", n, f) ;
}
```

Exercise 2: TP

Write a program to display all the divisors of a number.

```
Algorithm dividers
Var n, i: integer
begin
 write("enter a nbr: ")
 read(n)
 for i←1 to n do
  if n mod i = 0 then
   write(i)
  end if
 end for
end.
```

```
#include <stdio.h>
int main() {
 int n, i;
 printf("enter a nbr: ");
 scanf("%d", &n);
 for(i=2; i<=n; i++)
  if (n%i==0)
    printf("%d\t", i) ;
}
```

Exercise 3: TD

Write an algorithm to display all the common divisors of two numbers.

```
Algorithm common
Var n,m, i: integer
begin
 write("enter two nbrs: ")
 read(n,m)
 i←1
 while i<=n and i<=m do
  if (n mod i = 0) and (m mod i=0) then
   write(i)
  end if
  i=i+1
 end while
end.
```

```
#include <stdio.h>
int main() {
 int n, i;
 printf("enter two nbrs: ");
 scanf("%d%d", &n, &m);
 i=1;
 while(i<=n && i<=m){
  if ((n%i==0)&&(m%i==0))
    printf("%d\t", i) ;
  i++ ;
 }
}
```

Exercise 4: TP

Write a program that displays the mirror image of an integer (displays it in reverse).

```
Algorithm mirror
Var n: integer
begin
 write("enter a nbr: ")
 read(n)
 do
  write(n mod 10)
  n←n div 10
 while n>0
end.
```

```
#include <stdio.h>
int main() {
 int n;
 printf("enter a nbr: ");
 scanf("%d", &n);
 do{
  printf("%d", n % 10) ;
  n=n / 10;
 }while(n>0) ;
}
```

Exercise 5: TD/TP

Write an algorithm with its C program that determines if a number is prime or not.

```
Algorithm Prime
Var x, i: integer
    isPrime : Boolean
begin
 write("enter a nbr: ")
 read(x)
 isPrime ←true
 for i←2 to x div 2 do
  if x mod i = 0 then
   isPrime ←false
  end if
 end for
 if isPrime then
  write(x," is Prime")
 else
  write(x," is not Prime ")
 end if
end.
```

```
#include <stdio.h>
int main() {
 int x, i, isPrime;
 printf("enter a nbr: ");
 scanf("%d", &x);
 isPrime =1;
 for (i=2 ;i<=(x / 2);i++)
  if (x%i==0)
    isPrime=0;
 if (isPrime==1)
  printf("%d est Prime" , x) ;
 else
  printf("%d n'est pas Prime", x) ;
}
```

```
Algorithm Prime2
Var x, i: integer
    isPrime: Boolean
begin
 write("enter a nbr: ")
 read(x)
 i←2
 isPrime ←true
 WHILE i<=(x div 2) et isPrime do
  if x mod i = 0 then
   isPrime ←false
  end if
  i←i+1
 end while
 if isPrime then
  write(x," is Prime ")
 else
  write(x," is not Prime ")
 end if
end.
```

```
#include <stdio.h>
int main() {
 int x, i, isPrime;
 printf("enter a nbr: ");
 scanf("%d", &x);
 i=2 ;
 isPrime =1;
 while ((i<=(x / 2) && (isPrime==1)) {
  if (x%i==0)
    isPrime=0;
  i++ ;
 }
 if (isPrime==1)
  printf("%d est Prime" , x) ;
 else
  printf("%d n'est pas Prime", x) ;
}
```

```
Algorithm Prime3
Var x, i, N: integer
    isPrime :booléen
begin
 write("entrer un nbr: ")
 read(N)
 for x←2 to N do
  i←2
  isPrime←true
  WHILE i<=(x div 2) et isPrime do
   if x mod i = 0 then
    isPrime←false
   end if
   i←i+1
  end while
  if isPrime then
   write(x)
  end if
 end for
end.
```

```
#include <stdio.h>
int main() {
 int x, i, isPrime, N;
 printf("entrer un nbr: ");
 scanf("%d", &N);
 for (x=2 ;x<=N;x++){
  i=2 ;
  isPrime =1;
  while ((i<=x / 2) && (isPrime==1)) {
   if (x%i==0)
     isPrime=0;
   i++ ;
  }
  if (isPrime==1)
   printf("%d\n" , x) ;
 }
}
```

Exercise 6: TD/TP

Write an algorithm with its C program that calculates the GCD (Greatest Common Divisor). Given that:

$$PGCD(a,b) = \begin{cases} PGCD\big(b,(a\ \%\ b)\big), & b \neq 0 \\ a, & b = 0 \end{cases}$$

```
Algorithm PGCD
Var x, y, t, a, b: integer
```

```
#include <stdio.h>
int main() {
```

```
begin
 write("enter two nbrs: ")
 read(x,y)
 a←x // for keeping x and y values
 b←y
 WHILE b>0 do
  t←b
  b←a mod b
  a← t
 end while
 write("PGCD(",x,"،",y,")=",a)
end.
```

```
 int x, y, t, a, b;
 printf("enter two nbrs: ");
 scanf("%d%d", &x, &y);
 a=x ;
 b=y ;
 while(b>0){
  t=b ;
  b=a % b;
  a= t;
 }
 printf("PGCD(%d،%d)= %d" ,x ,y ,a) ;
}
```

Exercise 7: TD

Write an algorithm to calculate the nth term of the Fibonacci sequence deended by:

$$u(n) = \begin{cases} 0 & si\ n = 0 \\ 1 & si\ n = 1 \\ u(n-2) + u(n-1), si\ n > 1 \end{cases}$$

```
Algorithm Fibonacci
Var n, i, Un_2, Un_1, Un: integer
begin
 write ("enter n ")
 read(n)
 Un_1 ← 1
 Un ← 0
 for i ← 1 to n do
  Un_2 ← Un_1
  Un_1 ← Un
  Un ← Un_2 + Un_1
 End for
 write("U(",n, ")=", Un)
end.
```

```
#include <stdio.h>
int main() {
 int n, i, Un_2, Un_1, Un;
 printf("enter n ");
 scanf("%d", &n);
 Un_1= 1;
 Un = 0;
 for(i = 1 ; i<=n; i++) {
   Un_2 = Un_1;
   Un_1 = Un;
   Un = Un_2 + Un_1;
 }
 printf("U(%d)= %d" , n , Un) ;
}
```

Exercise 8: TP

If you knew that

$$\pi = 4\sum_{k=0}^{n}\frac{(-1)^k}{2k+1} = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11}$$

Write a program that calculates the approximate value of π.

```
Algorithm alg_pi
Var n, i, p, d: integer
p: real
begin
 do
  write ("enter a positive integer ")
  read(n)
 WHILE n<0
 p ← 0
 d ← 4
 for i ← 1 to n do
  p ← p + d / (i*2+1)
  d ← -d
 End for
 write("pi=", p)
end.
```

```
#include <stdio.h>
int main() {
 int n, i, d;
 float p ;
 do{
  printf("enter a positive integer ");
  scanf("%d", &n);
 }while (n<0) ;
 p = 0;
 d = 4;
 for(i = 0 ; i<n; i++) {
  p = p + d *1./ (i*2+1);
  d = -d;
 }
 printf("pi=%f", p) ;
}
```

Exercise 9: TD

If you knew that

$$\exp(x) := \sum_{k=0}^{n}\frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} +$$

Write a program that calculates exp(x) (x is a real number and n is an integer).

```
Algorithm alg_exp
Var n, i, exp, makam: integer
    exp, bast: real
begin
 write ("enter x ")
 read(x)
 do
  write ("enter a positive integer ")
  read(n)
 while n<0
 exp ← 0
 bast ← 1
 makam ← 1
 for i ← 1 to n do
  exp ← exp + bast / makam
  bast ← bast*x
  makam ← makam * i
 End for
 write("exp(",x, ")=", exp)
end.
```

```
#include <stdio.h>
int main() {
 int n, i, makam;
 float exp, bast;
 printf("enter x ");
 scanf("%f", &x);
 do{
  printf("enter a positive integer ");
  scanf("%d", &n);
 |while (n<0) ;
 exp = 0;
 bast = 1;
 makam = 1;
 for(i = 1 ; i<=n; i++) {
  exp = exp + bast / makam;
  bast = bast*x;
  makam = makam * i;
 }
 printf("exp(%f)= %f" , x , exp) ;
}
```

Exercise 12: (at home)

If you know that the square root of a number "a" is calculated by the following recursive relationship :

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

$$x_0 = 1$$

Write an algorithm with its C program that calculates the square root of a number « a » with approximation error $\varepsilon = 10^{-6}$. In other words $(x_n)^2 - \text{a} \leq \varepsilon$

```
Algorithm alg_sqrt
Const eps=0.000001
Var a, x: real
begin
 write ("enter a nbr: ")
 read(a)
 x ← 1
 do
  x ← (x+a/x)/2
 while x*x-a>eps
 write("√",a ,"=", x)
end.
```

```
#include <stdio.h>
int main() {
  const float eps=0.000001;
  float a, x;
  printf("enter a nbr: ");
  scanf("%f", &a);
  x = 1;
  do
    x=(x+a/x)/2 ;
  while(x*x-a>eps);
  printf("v%f = %f" , a, x) ;
}
```