Matière : Algorithmique et structures de données I
Date : 08 Décembre 2022
Durée : 1h30
Semestre 1

Université Abou Bakr Belkaïd - Tlemcen
Faculté des Sciences
Département de Mathématiques
1ère Année LMD-Maths

Contrôle Continu

Aucun document n'est autorisé
Les appareils portables doivent être éteints et posés sur le bureau du surveillant
Les solutions doivent être rédigées en **C**

1 Affichage (10 pts)

Montrer l'historique d'exécution des deux programmes suivants :

```
#include <stdio.h>
#include <stdlib.h>

void main ( ) {
 int i, toto=5;
 for (i=5 ; i>=0 ; i--)
 {
   switch ( toto )
   {  case 2 :  toto = i-3;
      case 4 :  toto = toto+1;  break ;
      case 8 :  toto = toto/10;  break ;
      case 10 :  toto = toto-8; break;
      case 6 || 0 :  toto = toto+7;  break ;
      default :  toto =toto+5;   break ;
   }
  printf (" toto = %d \n", toto );
}
}
```

```
#include <stdio.h>
#include <stdlib.h>
void main( ) {
 int  x=1 , y=2 , z=4 , A;
 if  (!(x && !z))   A = x+y/z+2;
    else   A = x+y*z+2;
 printf("A=%d\n",A);
 for (int i=1 ; i<=z ; i++) {
     y=2;
     while(y<=2*z)  {
       if (x%2)
         {z--;
          x--;}
       else
         x++;
       y+=2;
     printf("x=%d, y=%d, z=%d\n",x,y,z);
                  }       }  }
```

2 Somme (4 pts)

Écrire un programme qui demande à l'utilisateur de saisir un nombre entier positif *n* puis calcule et affiche la somme des factorielles de tous les nombres inférieurs à *n* : $S=\sum_{k=1}^{n} k!$

Exemple : Pour $n=1$: $S=1!$
Pour $n=2$: $S=1!+2!$
Pour $n=3$: $S=1!+2!+3!$

3 Racine carrée (6 pts)

La racine carrée d'un réel positif ***A*** peut être obtenue par une méthode itérative en utilisant la suite récurrente

$$\begin{cases} U_0 = A/2 \\ U_n = \left(U_{n-1} + \dfrac{A}{U_{n-1}} \right) /2 \end{cases}$$

qui converge vers la racine carrée de ***A.*** Le calcul s'arrête lorsque la condition suivante sera vérifiée :

$$\left|\frac{U_{n+1}-U_n}{U_{n+1}}\right| \leq \varepsilon \quad (\varepsilon \text{ donnée})$$

Écrire un programme qui calcule l'approximation de $\sqrt{A}$ avec une certaine précision ε et qui affiche le nombre de termes de la suite (U_n) nécessaires pour arriver au résultat.

Remarque :

- N'oubliez pas de contrôler la saisie de ***A.***
- En C, il existe une fonction **fabs** dans l'entête **math.h** permettant de calculer la valeur absolue d'un nombre réel.

Exemple d'éxecution:

Donnez un nombre positif ***A*** = -9
Donnez un nombre positif ***A*** = 34
Donnez une précision ***Epsilon*** = 0.000001
La racine carrée de ***A*** : 5.830952
Le nombre de termes nécessaires : 6

« Bon courage »

Matière : Algorithmique et structures de données I
Date : 08 Décembre 2022
Durée : 1h30
Semestre 1

Université Abou Bakr Belkaïd - Tlemcen
Faculté des Sciences
Département de Mathématiques
1ère Année LMD-Maths

Correction du Contrôle Continu

Aucun document n'est autorisé
Les appareils portables doivent être éteints et posés sur le bureau du surveillant
Les solutions doivent être rédigées en **C**

1 Affichage (10 pts)

Montrer l'historique d'exécution des deux programmes suivants :

```c
#include <stdio.h>
#include <stdlib.h>

void main ( ) {
 int i, toto=5;
 for (i=5 ; i>=0 ; i--)
 {
   switch ( toto )
   {  case 2 :  toto = i-3;
      case 4 :  toto = toto+1;  break ;
      case 8 :  toto = toto/10;  break ;
      case 10 :  toto = toto-8; break;
      case 6 || 0 :  toto = toto+7;  break ;
      default :  toto =toto+5;   break ;
   }
  printf (" toto = %d \n", toto );
}
}
```

```c
#include <stdio.h>
#include <stdlib.h>
void main( ) {
 int  x=1 , y=2 , z=4 , A;
 if  (!(x && !z))   A = x+y/z+2;
    else   A = x+y*z+2;
 printf("A=%d\n",A);
 for (int i=1 ; i<=z ; i++) {
     y=2;
     while(y<=2*z)  {
       if (x%2)
         {z--;
          x--;}
       else
         x++;
       y+=2;
     printf("x=%d, y=%d, z=%d\n",x,y,z);
                  }      }  }
```

Solution

```
toto = 10
toto = 2
toto = 1
toto = 8
toto = 0
toto = 5
```

```
A=3
x=0, y=4, z=3
x=1, y=6, z=3
x=0, y=8, z=2
x=1, y=4, z=2
x=0, y=6, z=1
```

2 Somme (4 pts)

Écrire un programme qui demande à l'utilisateur de saisir un nombre entier positif n puis calcule et affiche la somme des factorielles de tous les nombres inférieurs à n : $S=\sum_{k=1}^{n} k!$

Exemple : Pour n=1 : S=1!
Pour n=2 : S=1!+2!
Pour n=3 : S=1!+2!+3!

Solution

```
#include <stdio.h>
#include <stdlib.h>
void main( ) {
 int n;
 double F=1,S=0;
  printf("donnez un entier positif n: "); scanf("%d",&n);
  for(int i=1; i<=n; i++)
      {   F=1;
          for(int j=1; j<=i; j++)
              F=F*j;
          S=S+F;
      }
  printf("La somme des factorielles de tous les nombres inferieurs a %d est %.0lf",n,S);
                }
```

3 Racine carrée (6 pts)

La racine carrée d'un réel positif ***A*** peut être obtenue par une méthode itérative en utilisant la suite récurrente

$$\begin{cases} U_0 = A/2 \\ U_n = \left(U_{n-1} + \dfrac{A}{U_{n-1}} \right)/2 \end{cases}$$

qui converge vers la racine carrée de ***A.*** Le calcul s'arrête lorsque la condition suivante sera vérifiée :

$$\left| \frac{U_{n+1} - U_n}{U_{n+1}} \right| \leq \varepsilon \quad (\varepsilon \text{ donnée})$$

Écrire un programme qui calcule l'approximation de $\sqrt{A}$ avec une certaine précision ε et qui affiche le nombre de termes de la suite (U_n) nécessaires pour arriver au résultat.

Remarque :

- N'oubliez pas de contrôler la saisie de ***A.***
- En C, il existe une fonction **fabs** dans l'entête **math.h** permettant de calculer la valeur absolue d'un nombre réel.

Exemple d'éxecution:

Donnez un nombre positif ***A*** = -9
Donnez un nombre positif ***A*** = 34
Donnez une précision ***Epsilon*** = 0.000001
La racine carrée de ***A*** : 5.830952
Le nombre de termes nécessaires : 6

Solution

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main (){
double A,eps,U0,U1,D;
 int compt=0;
 do{
      printf("donnez un nombre positif A= ");
      scanf("%lf",&A);
     } while(A<=0) ;
 printf("Donnez une precision Epsilon= ");
 scanf("%lf",&eps);
 U0=A/2;
 do {
      U1=(U0+A/U0)/2;
      D=fabs((U1-U0)/U1);
      U0=U1;
      compt++;
     } while(D>eps);
printf("La racine carree de A : %lf\n",U0);
printf("Le nombre de termes necessaires : %d ",compt);
return 0;
}
```