IF121
Contrle continu n^o 3a

La durée de l'épreuve est de 40 minutes.

On considère le programme suivant décrit en pseudo-langage. Il vous est demandé d'écrire le programme C−− correspondant.

Une grande attention du *CORRECTEUR* sera portée sur l'indentation de votre programme et au fait que les differentes parties fonctionnent ou pas. Exceptionellement, vu le temps impartis, **il ne vous est pas demandé de reporter les commentaires déjà présent dans le pseudo.** Mettez simplement votre nom en commentaire au debut du programme pour permettre de s'y retrouver, et ceux que vous jugez necessaires.

Sujet :

```
      Constantes
         dim = 20

      Types
         ligne = tableau de dim entiers
         ecran = tableau de dim lignes


      Procedure init(reference ecran e, reference entier x,y)

         Variables i,j : deux entiers

         Debut init

           x <- dim/2
           y <- dim/2

           pour i de 1 a dim faire
              pour j de 1 a dim faire
              e[i][j] <- 0
              finpour
           finpour
         fin init


      Procedure affiche(valeur ecran e, valeur entier x,y)

         Variables i,j : deux entiers

         Debut affiche
           pour i de 1 a dim faire
              pour j de 1 a dim faire
                si i=y et j=x faire afficher ``#''
                sinon
                   si e[i][j]=1 faire afficher ``+''
                   sinon afficher ``.''
                   fin si
                finsi
              finpour
              afficher endl
           finpour
           afficher endl endl endl
         fin affiche

// Parmis les quatre fonctions suivantes, il n'est exige que d'en ecrire une, a votre choix

      Fonction avance_haut(valeur nb_pas entier,reference e un ecran, valeur x,y deux entiers) : retourne entier

        debut avance_haut
          tantque( y>1 et nb_pas>0) faire
```

```
            e[y][x] <- 1
            y <- y -1
            nb_pas <- nb_pas -1
        fin tantque
      retourner y
      fin avance_haut


    Fonction avance_droite(valeur nb_pas entier,reference e un ecran, valeur x,y deux entiers) : retourne entier

      debut avance_droite
        tantque( x<dim et nb_pas>0) faire
            e[y][x] <- 1
            x <- x + 1
            nb_pas <- nb_pas -1
        fin tantque
      retourner x
      fin avance_droite


    Fonction avance_bas(valeur nb_pas entier,reference e un ecran, valeur x,y deux entiers) : retourne entier

      debut avance_bas
        tantque( y<dim et nb_pas>0) faire
            e[y][x] <- 1
            y <- y + 1
            nb_pas <- nb_pas -1
        fin tantque
      retourner y
      fin avance_bas


    Fonction avance_gauche(valeur nb_pas entier,reference e un ecran, valeur x,y deux entiers) : retourne entier

      debut avance_gauche
        tantque( x>1 et nb_pas>0) faire
            e[y][x] <- 1
            x <- x - 1
            nb_pas <- nb_pas -1
        fin tantque
      retourner x
      fin avance_gauche


// Le programme principal

     Programme tortue

         Variables
            e ecran
            x,y,direction,nb_pas quatres entiers

         debut tortue

            init(e,x,y)
            affiche(e,x,y)

            faire
              afficher ``combien de pas ?''
              lire nb_pas
              afficher ``dans quelle direction ?''
              lire direction

// parmis les 4 lignes suivantes, n'ecrivez d'abord que celle qui
// correspond a la fonction que vous avez choisit d'ecrire.

              si direction=1 faire y=avance_haut(nb_pas,e,x,y) fin si
              si direction=2 faire y=avance_droite(nb_pas,e,x,y) fin si
              si direction=3 faire y=avance_bas(nb_pas,e,x,y) fin si
              si direction=4 faire y=avance_gauche(nb_pas,e,x,y) fin si

              affiche(e,x,y)

           tantque (nb_pas!=0)

        fin tortue
```

```
#include <iostream.h>

const int dim = 20;

typedef int ligne[dim];
typedef ligne ecran[dim];  // convention : ecran[y][x]


void affiche(ecran e,int x, int y){
  int i,j;
  for (i=0;i<dim;i++){
    for (j=0;j<dim;j++){
      if ((i==y) && (j==x) ) {
cout << "#";
      }
      else{
if (e[i][j]==1) {
  cout << "+";
}
else cout << ".";
      }
    }
    cout << endl;
  }
  cout << endl << endl << endl;
}


void init(ecran e, int &x, int &y){
  int i,j;
  x=dim/2;
  y=dim/2;
  for(i=0;i<dim;i++){
    for(j=0;j<dim;j++){
      e[i][j]=0;
    }
  }
}

int avance_haut(int nb_pas,ecran e,int x,int y){
  while(y>0 & nb_pas >0) {
    e[y][x]=1;
    y--;
    nb_pas--;
  }
  return y;
}

int avance_bas(int nb_pas,ecran e,int x,int y){
  while(y<dim-1 & nb_pas >0) {
    e[y][x]=1;
    y++;
    nb_pas--;
  }
  return y;
}

int avance_droite(int nb_pas,ecran e,int x,int y){
  while(x<dim-1 & nb_pas >0) {
    e[y][x]=1;
    x++;
    nb_pas--;
  }
  return x;
}

int avance_gauche(int nb_pas,ecran e,int x,int y){
  while(x>0 & nb_pas >0) {
    e[y][x]=1;
    x--;
    nb_pas--;
  }
  return x;
}


void main(){
  ecran e;
  int x,y;
  int direction, nb_pas;

  init(e,x,y);
  affiche(e,x,y);

  do{
    cout << "combien de pas ? ";
    cin >> nb_pas;

    cout << "quelle direction ? ";
    cin >> direction;
```

```
    if(direction==1) y=avance_haut(nb_pas,e,x,y);
    if(direction==2) x=avance_droite(nb_pas,e,x,y);
    if(direction==3) y=avance_bas(nb_pas,e,x,y);
    if(direction==4) x=avance_gauche(nb_pas,e,x,y);

    affiche(e,x,y);

  }
  while(nb_pas!=0);


}
```