CERTAMEN #2 INF-154

Wenceslao Palma, Laura Griffiths

1. (30 ptos.) Utilizando la siguiente definición de nodo:

```
typedef struct node{
  struct node *sgte;
  int v;
}nodo;
```

escriba la función **void invertir(nodo **)**, la cual reorganiza los punteros de una lista. Por ejemplo, la lista:

```
INICIO
     1 --> 2 --> 3 --> 4 --> 5 --> 6 -->NULL
```

queda de la sgte manera luego de haber invocado la función **invertir(&INICIO)**:

```
                                          INICIO
NULL <-- 1 <-- 2 <-- 3 <-- 4 <-- 5 <-- 6
```

Importante: lo que se reorganiza son los punteros y no el contenido de los nodos.

```
  (a) uso de funciones 5 ptos.
  (b) uso de punteros  7 ptos.
  (c) solucion 18 ptos.

void invertir(nodo **INICIO)
{
  nodo *actual,*antecesor,*sucesor;

         antecesor=NULL;
         actual=*INICIO;

  while (actual!=NULL)
  {
       sucesor=actual->sgte;
       actual->sgte=antecesor;
       antecesor=actual;
       actual=sucesor;
  }
  *INICIO=antecesor;
}
```

2. (30 ptos.) En un archivo de texto, llamado "circunferencias.txt" se encuentra las coordenadas del centro (x, y) y radio r de circunferencias (una por línea). En otro archivo de texto llamado "puntos.txt" se encuentra las coordenadas (x, y) de puntos. Escriba las siguientes funciones:

   - **void crearListaCircunf(nodoC **);** la cual crea una lista enlazada, apuntada por ***C**, a partir de las datos almacenados en "circunferencias.txt".
   - **void crearListaPuntos(nodoP **);** la cual crea una lista enlazada, apuntada por ***P**, a partir de las datos almacenados en "puntos.txt".
   - **void pertenece(nodoC *, nodoP *);** la cual muestra por pantalla si un punto se encuentra al interior de alguna circunferencia. La salida, estará compuesta para cada punto, con un mensaje de la forma:
     Punto i **esta contenido en la figura** j.
     Si el punto no se encuentra contenido en ninguna figura el mensaje deberá ser:
     Punto i **no esta contenido en ninguna figura**.
     Puntos y figuras deben ser numeradas en el orden en el cual aparecen en sus respectivos archivos de entrada.

   Definición de nodos (4 ptos):

```
typedef struct circunferencia{
        int x;
        int y;
        int r;
        struct circunferencia *sgte;
}nodoC;

typedef struct punto{
        int x;
        int y;
        int r;
        struct punto *sgte;
}nodoP;
```

Función **void crearListaCircunf(nodoC **);** (8 ptos)

```
void crearListaCircunf(nodoC **C){
     FILE *fp;
     nodoC *nuevo,*tmp;

     fp = fopen("circunferencias.txt","r");
     if (fp==NULL){printf("error en la apertura del archivo .....\n");exit(0);}

     while (!feof(fp)){
            nuevo = (nodoC *)malloc(sizeof(nodoC));
            fscanf(fp,"%d %d %d \n",&nuevo->x,&nuevo->y,&nuevo->r);
            nuevo->sgte=NULL;

            if (*C==NULL)
                *C=nuevo;
            else{
                 tmp=*C;
                 while (tmp->sgte!=NULL)
                        tmp=tmp->sgte;
                 tmp->sgte=nuevo;
            }
     }
     fclose(fp);
}
```

Función **void crearListaPuntos(nodoP **);** (8 ptos)

```
void crearListaPuntos(nodoP **P){
     FILE *fp;
     nodoC *nuevo,*tmp;

     fp = fopen("puntos.txt","r");
     if (fp==NULL){printf("error en la apertura del archivo .....\n");exit(0);}

     while (!feof(fp)){
            nuevo = (nodoP *)malloc(sizeof(nodoP));
            fscanf(fp,"%d %d \n",&nuevo->x,&nuevo->y);
            nuevo->sgte=NULL;

            if (*P==NULL)
                *P=nuevo;
            else{
                 tmp=*P;
                 while (tmp->sgte!=NULL)
                       tmp=tmp->sgte;
                tmp->sgte=nuevo;
            }
     }
     fclose(fp);
}
```

Función **void pertenece(nodoC *, nodoP *);** (10 ptos)

```
void pertenece(nodeC *C, nodeP *P){
     nodoP *auxP;
     nodoC *auxC;
     int p,c;

     auxP=P;
     auxC=C;

     p=1;
     while (auxP!=NULL){
           auxC=C;
           c=1;
           flag=0;
           while (auxC!=NULL){
               if (sqrt(pow(auxP->x-auxC->y,2)+pow(auxP->y-auxC->y,2))<=auxC-r){
                  printf("Punto %d esta contenido en la figura %d\n",p,c);
                  flag=1;
               }
               auxC=auxC->sgte;
               c++;
           }

           if (flag==0)
              printf("Punto %d no esta contenido en ninguna figura \n",p);

           auxP=auxP->sgte;
           p++;
      }
}
```