Prova scritta dell'esame di Programmazione e Laboratorio 18 Settembre 2006

1. Sia data una matrice di caratteri `A` di taglia fissata `n×m` e sia data una stringa `s` con `length(s)<m`. Si scriva una procedura o funzione che determini se la stringa `s` è presente in una riga di `A`, scritta da sinistra verso destra o da destra verso sinistra, e restituisca tramite degli opportuni parametri la posizione in cui tale stringa ha inizio. Nel caso in cui la stringa non figuri mai nella matrice, si dia a entrambi tali parametri il valore 0, e nel caso figuri più volte, si diano a tali parametri le coordinate della prima occorrenza della stringa procedendo dall'alto in basso, da sinistra a destra.

```
procedure find_string(mat: matrice; s: string; n,m: integer);
var i,j,k: integer;
  trovato,flag: boolean;
begin
k:=1;
trovato:=true;
flag:=false;

if flag<>true then begin
for i:=1 to n do
 for j:=1 to m do
  begin
  if length(s)<=m-j then begin

  if s[k]=mat[i,j]then begin
  k:=k+1;
   while (k<=length(s)) and (trovato=true) do begin

    if s[k]<>mat[i,j-k+1] then
      trovato:=false
    else
      k:=k+1;

   end;
   if k>length(s) then begin flag:=false; WRITELN(i,'     ',j-length(s)+1); end;
   end;

   if s[length(s)]=mat[i,j]then begin
    k:=length(s)-1;
   while (k<>0) and (trovato=true) do begin

    if s[k]<>mat[i,j-k+length(s)] then
      trovato:=false
    else
      k:=k-1;
   end;
   if k=0 then begin flag:=false; WRITELN(i,'     ',j+length(s)-1); end
   end;
   end;
```

```
  end;
end;
end;
```

2. Siano definiti i seguenti tipi:

```
type lista=^cella;
     cella=record
        asc:integer;
        ord:integer;
        next:lista;
           end;
```

Dove i campi `asc` e `ord` rappresentano l'ascissa e l'ordinata di un punto. Si scriva una procedura tale che, presi in input due interi `x` ed `y` (coordinate di un punto) ed una lista concatenata di celle definite come sopra, la modifichi in maniera tale da:

- eliminare quelle celle i cui campi `asc` e `ord` rappresentano l'ascissa e l'ordinata di un punto che ha distanza minore di 3 e dal punto di coordinate cartesiane `(x,y)`;
- se i campo `asc` e `ord` della cella osservata sono l'ascissa e l'ordinata di un punto a distanza maggiore di 10 dal punto `(x,y)`, si aggiunga subito dopo una cella i cui campi `asc` e `ord` rappresentano il punto simmetrico rispetto all'origine del punto rappresentato dalla cella osservata.

```
procedure punti(var l:lista; x,y: integer);
var q:lista;
begin
if l<>nil then
begin
  if sqrt((x-l^.asc)^2+(y-l^.ord)^2)>10 then
  begin
    new(q); q^.asc:=l^.asc; q^.ord:=l^.ord;
    q^.next:=l^.next;
    l^.next:=q;
    punti(l^.next^.next);
  end
  else
  if sqrt((x-l^.asc)^2+(y-l^.ord)^2)<3 then
  begin
    new(q); q:=l;
    l:=l^.next;
    dispose(q);
    punti(l);
  end
  else
    punti(l^.next);
end;
end;
```

3. Scrivere una procedura che, preso in input un albero binario, lo modifichi in maniera tale che:
   - Se un nodo `n` ha due figli, si scambiano i suoi due sottoalberi (il sottoalbero destro diventa sinistro e viceversa)
   - Se un nodo `n` ha un solo figlio, il figlio viene cancellato, e i figli destro e sinistro del nodo cancellato diventino rispettivamente i figli destro e sinistro del nodo `n`.

```
procedure cambia(var a: albero);
var temp: albero;
begin
if a<>nil then
begin
cambia(a^.sx); cambia(a^.dx);
  if (a^.sx<>nil) and (a^.dx<>nil) then
  begin
    new(temp);
    temp:=a^.sx;
    a^.sx:=a^.dx;
    a^.dx:=temp;
  end
  else
  begin
    if (a^.sx=nil) and (a^.dx<>nil) then
      begin
      a^.sx:=a^.dx^.sx;
      a^.dx:=a^.dx^.dx;
      end;

    if (a^.sx<>nil) and (a^.dx=nil) then
      begin
      a^.sx:=a^.sx^.sx;
      a^.dx:=a^.sx^.dx;
      end;
  end;
```

4. Si descriva una struttura dati per rappresentare un grafo non orientato i cui archi sono etichettati con dei caratteri e i cui nodi contengano un campo `info` di tipo `string`. Si scriva quindi una procedura che, preso in input un tale grafo e un nodo sorgente, scriva nel campo stringa di ogni nodo il valore della stringa che si ottiene concatenando le lettere degli archi che si percorrono nel cammino più corto dalla sorgente a quel nodo.

```
type lista=^cella;
          cella=record;
            nodo: num_nodi;
            arco: char;
            next: lista;
          end;
type istanza=record
          info: string;
          next: lista;
type grafo=array[1..num_nodi] of istanza;
```

```
procedure BFS(g: grafo; var dist: array[1..num_nodi]of string; var pred:array[1..num_nodi]of
integer; s: integer);
          var q:coda; j: integer; aux: lista;
begin
for i:=1 to num_nodi do
begin
  pred[i]:=0;
  dist[i]:=' ';
end;
//dist[s]:=grafo[s].info;
q.primo:=nil;
enqueue(q,s);
while q.primo<>nil do
begin
  j:=q^.info; write(j,'   ');
  aux:=g[j];
  while aux<>nil do begin
     if dist[aux^.nodo]<>' ' then
        begin
           dist[aux^.nodo]:=dist[j]+aux^.arco;
              pred[aux^.nodo]:=j;
              enqueue(q,aux^.nodo);
        end;
     aux:=aux^.next;
   end;
dequeue(q);
end;
end;

procedure renomina_nodi(var g: grafo; dist: array[1..num_nodi]of string);
          var i: integer;
begin
for i:=1 to num_nodi do
  g[i].info:=dist[i];
end;
```

Pierluigi Ballatore