RICORSIONE-bis

ricorsione su dati automatici

PRE=(dim>=0, A[0..dim-1] def., z def.)

int cerca(int *A, int dim, int z)

POST=(restituisce posizione di z e -1 se z non c'è)

```
int cerca(int *A, int dim, int z)
{ if(dim==0)

        return -1 ;
        else
        {if(A[0]==z)   return 0;
else
                { int v=cerca(A+1,dim-1,z);

                   if(v==-1) return -1;

                      else return v+1;
                }

            }
```

} POST=(restituisce prima posiz. di z e -1 se z non c'è)

(A[0]!=z e dim>0)

PRE_ric

```
int v=cerca(A+1,dim-1,z);
```

POST_ric

```
 if(v==-1) return -1;
     else return v+1;
```

POST=(restituisce prima posizione di z e -1 se z non c'è)

altra soluzione

```
int cerca(int *A, int pos, int dim, int z)
{ if(pos==dim) return -1 ;
else
{if(A[pos]==z)   return pos;
          else

                return cerca(A,pos+1,dim,z);

}
```

POST=(restituisce minima posiz. di z in A[pos..dim-1] e -1 se z non c'è)

e ancora altra !

```
int cerca(int *A, int pos, int z)
{ if(pos==-1) return -1 ;
else
{if(A[pos]==z)   return pos;
else

   return cerca(A,pos-1,z);
```

} POST=(restituisce massima posiz. di z A[0..pos] e -1 se z non c'è)

altro esempio:

PRE=(A[0..dim-1], y è def., A=vA)

int del(int*A, int dim, int y)

POST= (A=(vA-y) e restituisce DIM(vA-y))

dove (vA-y)= quello che resta di vA dopo aver eliminato gli y e ricompattato a sinistra e DIM(vA-y) la sua dimensione

soluzione iterativa:

```
int del(int*A, int dim, int y)
{ int z=0;
for(int x=0; x<dim; x++)\\R
  if(A[x]!=y)
      {A[z]=A[x]; z++;}
 return z;
}
```

R=(0<=z<=x<=dim,
A[0..z-1]=(vA[0..x-1]-y)

soluzione ricorsiva

PRE=(0<=z<=x<=dim, A[x..dim-1] def,A=vA)

```
int del(int*A, int dim, int x, int z, int y)
{ if(x==dim) return 0;
  if(A[x]!=y)
{A[z]=A[x];return 1+del(A,dim,x+1,z+1,y);}
  else
    return del(A,dim,x+1,z,y);
}
```

POST=(se restituisce k, A[z..(z+k)-1] = (vA[x..dim-1]-y))

problema simile: bandiera a 2 colori
array A con 2 valori (B e N) e vogliamo ordinarlo scambiando elementi

```
int p=0;
for(int s=0; s<dim; s++)
  if(A[s]==B)
    {scambia(A,s,p); p++;}
```

R= (0<=p<=s<=dim, A[0..p-1]=B, A[p..s-1]=N, A[0..s-1]= ord(vA[0..s-1]), A[s..dim-1]=vA[s..dim-1])

PRE=(0<=p<=s<=dim, A[0..p-1]=B, A[p..s-1]=N)

```
void ord(int*A,int p, int s, int dim)
{ if(s<dim)
   if(A[s]==B)
    {scambia(A,s,p); ord(A,p+1,s+1,dim);}
   else
     scambia(A,p,s+1,dim);
}
```

POST=(ha messo a posto anche la parte A[s..dim-1])

bandiera di 3 colori (Edsger Dijkstra)

```
int p=0, s=0, t=dim-1;
while(s<t) //R?
  if(A[s]==Green)
    {scambia(A,p,s); p++; s++;}
  else
    if(A[s]==Red)
     {scambia(A,s,t); t--;}
   else
     s++;
```