11/10/18

Given an array return. max and the position where max resides.

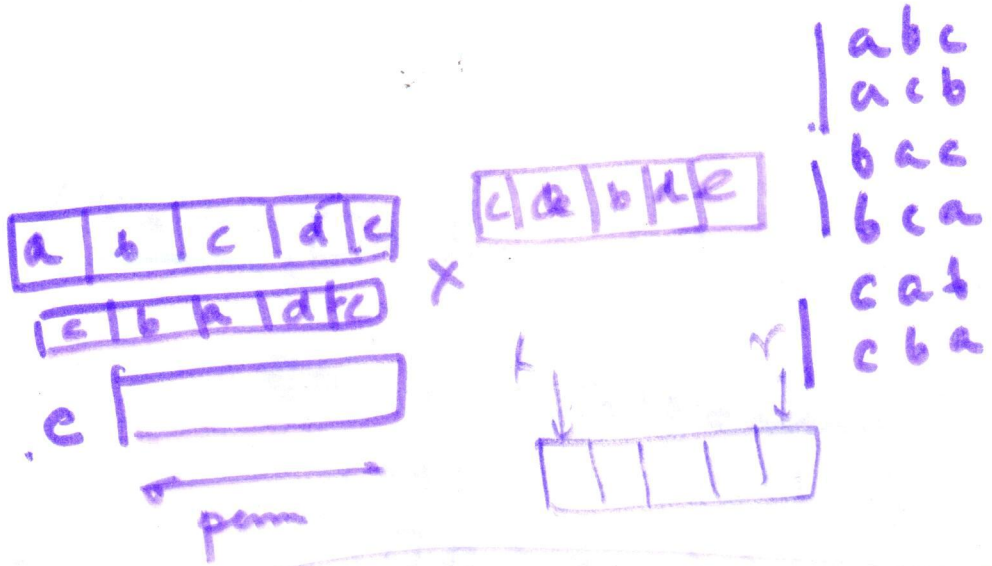
write a function

```
int   Findmax (int n, int a[], int *X)
{  max = a[0], maxi = a[0];
   for (i=0; i<n; i++),
       if (max < a[i])
         { max = a[i]; maxi = i; }
   *x = max.i
   return (maxi);
}

main()
{ int Y, j

   j = Findmax (n, a, &Y);
   printf("%d %d", j, Y);
}
```

char* p = "abc"

↓

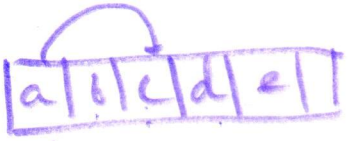
want to find all the permutation of abc.




permin = ✗ swap (
first position + (permutation on smaller

permutation (x, l, r, a).

{ If (l == r).
printf("%s", a);




abcde.

cbade

push (~~a~~ l, i, ~~j~~).

for (i = l, i < r; i++).

{ push

~~swap (a[l], a[i])~~;

permutation (l+1, r, a);

~~swap (a[i], a[l])~~; } p abcde.

anti push (l, i, a).

} anti push.

→ abcde.

## ANAGRAM

HARY → ARMY.

```
int findanagram (char *a, char *b, )
    n = strlen(a)
{   for(i=0; i<n; i++)
    {  x = a[i]
       for(j=0; j<n; j++)
          if (b[j] == x)
             b[j] = -1
             continue;
       flag = 0; break;
    }
    return (flag);
}
```

```
if (strlen(a) ==
{ strlen(b)
```

maya mary

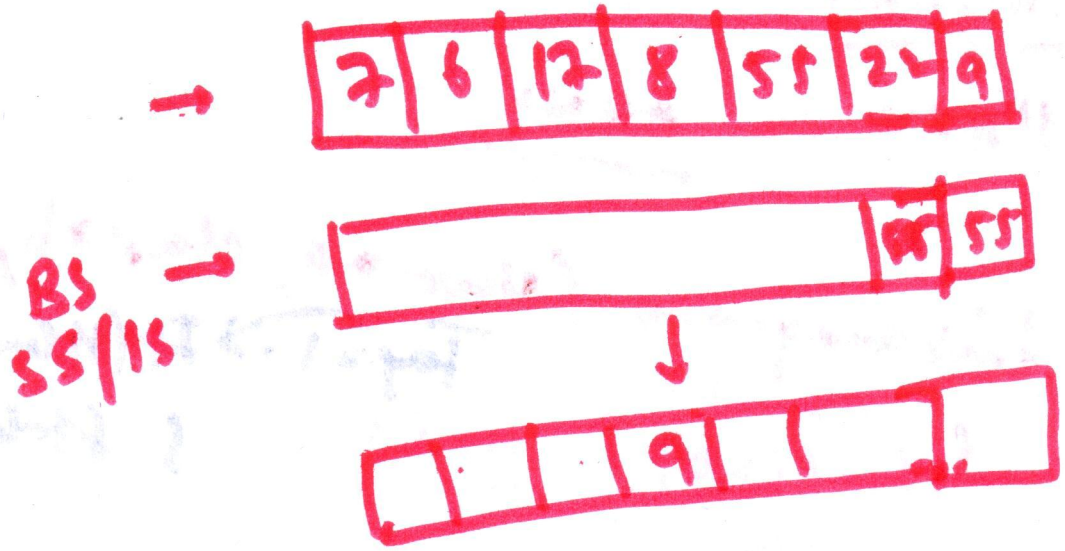
0 = Strcmp (a, b) ✓

Bubble sort = $O(n^2)$

$T(n) = T(n/2) + k.$

$T(n) = T(n-1) + n.$

$T(n) = T(n/2) + 1$

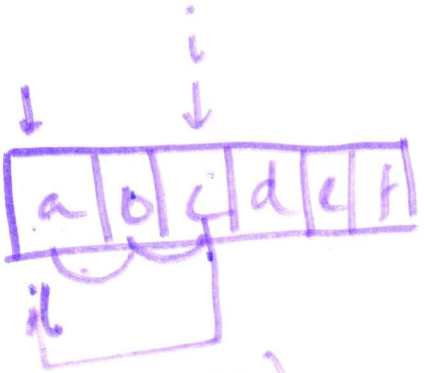



$T(n) = 2 \cdot T(n/2) + k \cdot n.$

$2 \cdot T(n/2) = 2^2 (T(n/4) + n.$

$T(n) = n \log n$

```
push(int l, int i, int a[])
{  x = a[i];
   for(j=l; j<i; j++).for(j=i-1;j>=l;j--)
      a[j+1] = a[j]
   a[l] = x;
}
}
```




```
push(int l, int i, char a[])
{  x = a[i];
   for(j = i-1; j >= l; j--)
      a[j+1] = a[j];
   a[l] = x;
}
```

```
anti-push(int l, int i, char a[])
{  x = a[l];
   for(i = l+1; j <= i; j++)
      a[j-1] = a[j];
   a[i] = x;
}
```