01.10.2021

ADS

Swap Pointers

```
void swap_string (char str1[], char str2[]) {
char *tmp;
tmp = str1;
str1 = str2;
str2 = tmp;
return;
}
```

Swap pointers pointing to those strings

if `char S1[n];` S1 is a constant pointer

if `char *S1;` S1 can be exchanged

Swap Strings

```
void swap_string (char str1[], char str2[]) {
char tmp[n];
strcpy(tmp, str1);
strcpy(str1, str2);
strcpy(str2, tmp);
return;
}
```

Sizeof()

ASCII rappresentation is on 1 byte (8 bits).

```
char s1[]="string";
char s2[6]={'s','t','r','i','n','g’};
char *p1="string";
```

```
strlen(s1)==6 //correct
sizeof(s1)==7 //correct
sizeof(s2)==6 // correct
sizeof(p1)==6 //wrong 4 (32bit) or 8(64bit) it's a pointer
sizeof(*p1)==6 //wrong it's a character so it's 1
```

in = "This is a very loooong string"

out undefined

n undefined

```
void myf (char *in, char *out, int *n) {

  char *tmp1, *tmp2;
  int l;
  out[0] = '\0’;

  tmp1 = in;
  while (*tmp1!='\0’) {
  while (*tmp1==' ‘) {
  tmp1++;
  }
  tmp2 = tmp1;
  while (*tmp2!=' ' && *tmp2!='\0’) {
  tmp2++;
  }
  l = tmp2 - tmp1; //basically the length ot "This"
  if (l > strlen(out)) {
  *n=l;
  strncpy (out, tmp1, l);
  out[l] = '\0’;
  }
  tmp1=tmp2;
  }
  return;
}
```

out = loooong

n = 7

s = "This 12345 is a string“

s = “Thisisastring”

T	**h**	**i**	**s**	**i**	**s**	**a**	**s**	**t**	**r**	**i**	**n**	**g**

```
void f (char *s) {
  int i, j;
  i = 0;
  while (i < strlen(s)) {
    if (s[i]==' ' || (s[i]>='0' && s[i]<='9')) {
      for (j=i+1; j<strlen(s)+1; j++)
      s[j-1] = s[j];
    } else {
      i = i + 1;
    }
  }
  return;
}
```

> characters are small integers

clicling

1	2	4	7	11
3	5	8		
6	9			
10				

```
display (float **mat, int n);
{
  for(int i=0; i<n; i++ );{
    for(int j=i; j<n; j++ );
    {
      ...
    }
  }
}
```

Palindrome

```
int palindrome (char *str);
```

```
void substring (char *str, int *letter, int *digit);
```