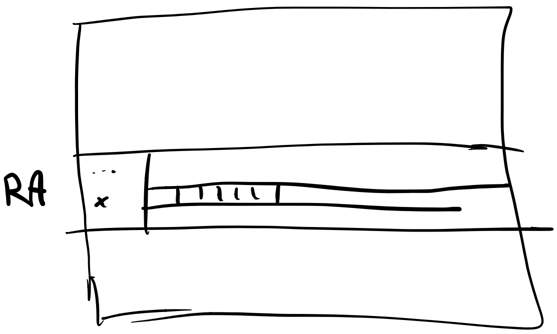
LEZIONE EXTRA

- VENERDÌ 15/1
  DALLE 8:30 (PUNTUALI)
  ALLE 10:00

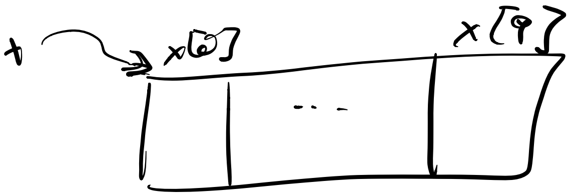
```
int x[10];

int x[n];
```



stdlib.h
stdint.h

```
int *p;
int x[10];
```

$$x \equiv \&(x[0])$$



```
p = x
x[0] = 15        *p = 15
```

```
void f(int x[], int n)
```

I

```
void f(int *x, int n)
```

II

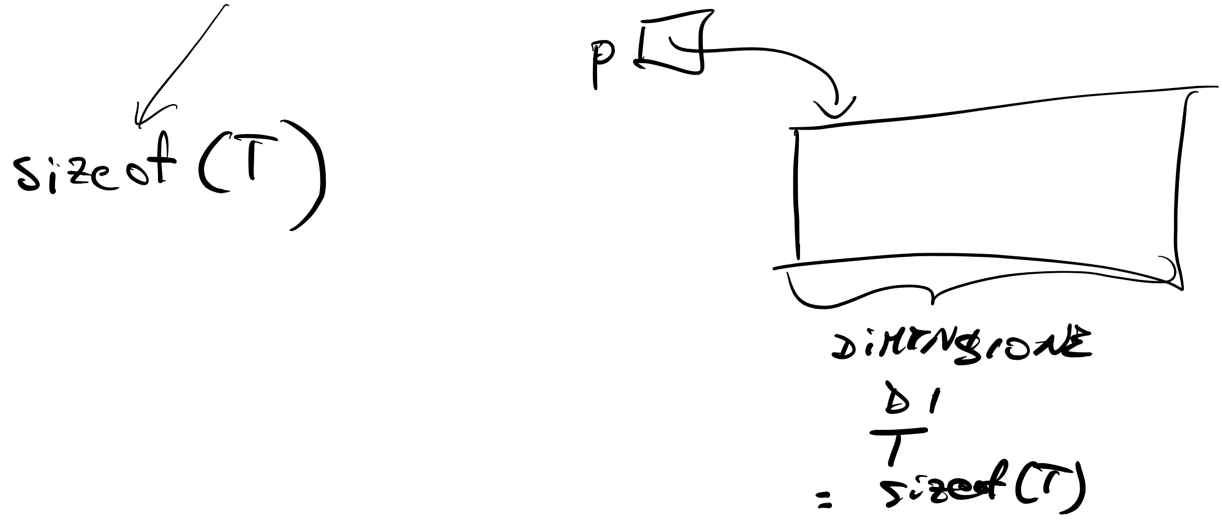
- FUNZIONE CHE DATO UN ARRAY DI INTERI E DATO UN INTERO K , CONTA QUANTI SONO I MULTIPLI DI K NELL'ARRAY

```
int    conta(int a[], int n, int k){
        int i,c;
        c =0;
        for (i=0; i<n; i++)
            if (a[i] % k==0)
                    c++;
        return c

}
```

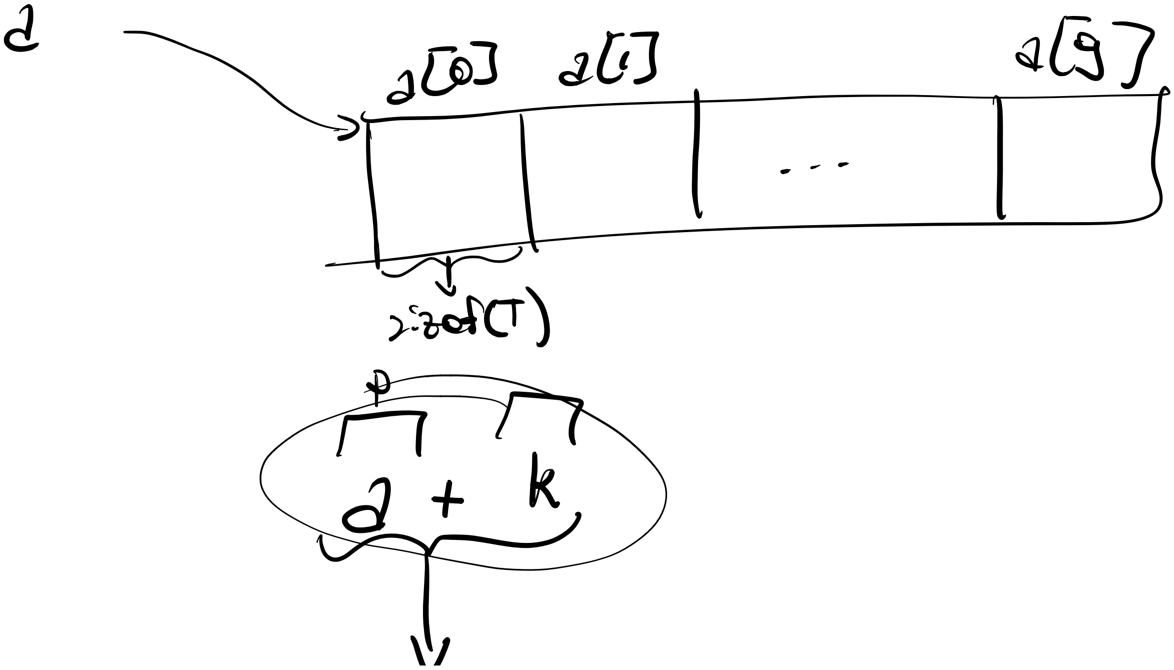
```
func    conta(a []int, k int)    int {
            c:=0
            for i :=0; i< len(a); i++ {
                    if    a[i] % K==0 {
                            c++

                    }
            }
            return  c
}
```

ARITMETICA DEI PUNTATORI

T *p;



T a[10];



locazione che si
trova k* sizeof(T) byte
dopo a

$$\underline{a+3} \equiv \&(a[3])$$

$$\underline{a[5]=7} \rightsquigarrow \underline{*(a+5)=7}$$

```
int   conta(int a[], int n, int k){
      int i,c;
      c =0;
      for (i=0; i<n; i++)
           if (a[i] % k==0)
                  c++;
      return c
}
```

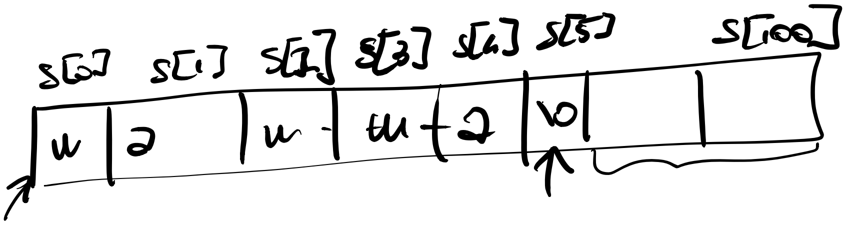
```
int   conta (int *a, int n, int k){
      int c = 0;
      int *p;
      for(p=a; p - a < n; p++)
            if *p % k == 0
                  c++;

      return c ;
}
```

STRINGHE IN C

char ⟷ 1 byte

```
char e;          |   rune r;
c = 'A';         |   r= 'A';
```

```
char  s[100]; ←
```



```
printf("%s", s);
```

string.h

```
strlen ( — )
strcpy
strcat
strstr
strrchr
  ⋮
```

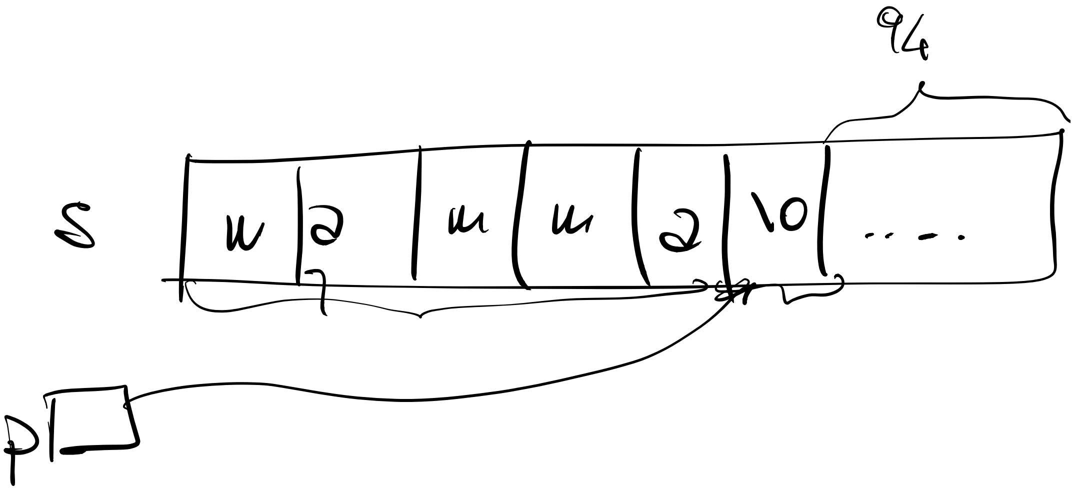
man string
man stdio

man strrchr

strlen

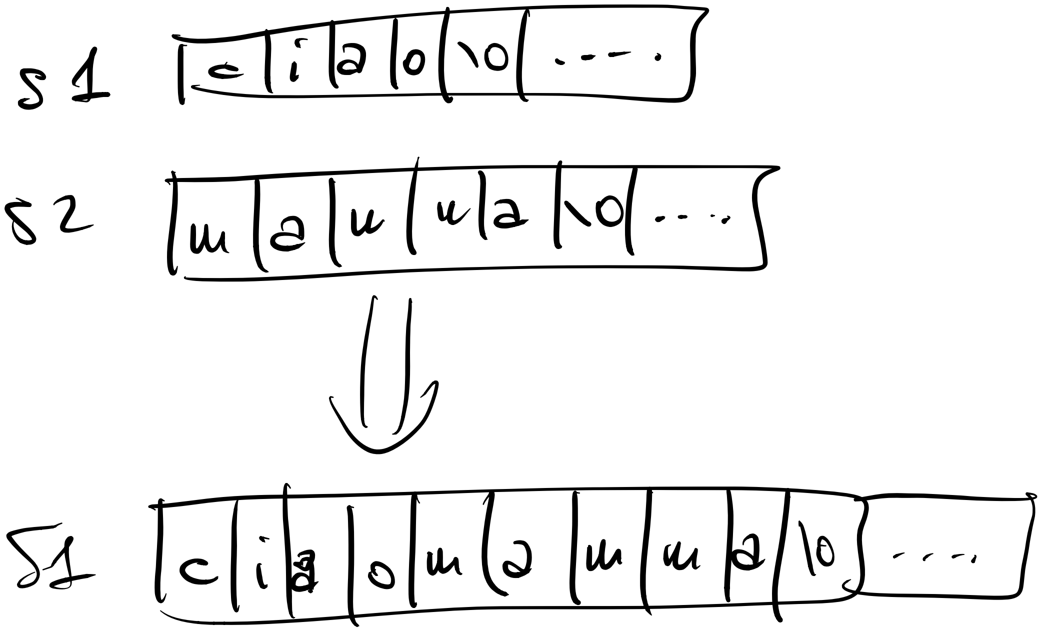
```
int   strlen (char  *s) {
        int   *p;
        for (p=s; *p; p++);
        return  p-s;

}
```

```
int     main()  {
        char  s[100];
      → scanf("%s", &s);
        n= strlen(s);
        printf("%d\n", n);

}
```



strcat (s1, s2)

concatena s2 in fondo a s1


s1
c | i | a | o | \0 | ....
s2
m | a | u | u | a | \0 | ....
s1
c | i | a | o | m | a | m | m | a | \0 | ....

```
void strcat(char *s1, char *s2){
    char *p, *p2;
    for (p=s1; *p; p++);
    for (p2=s2; *p2; p2++, p++)
            *p = *p2;
    *p=0;
}
```

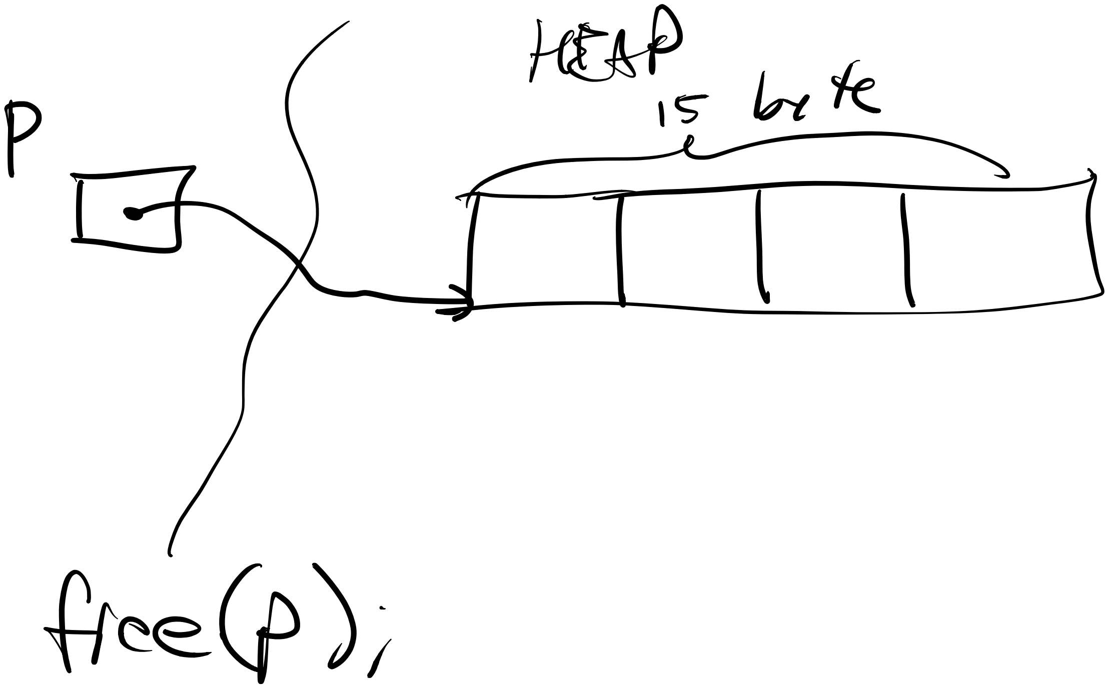
```
char *strchr(char *s, char c){
    for (; *s; s++)
        if (*s== c)
            return s;
    return NULL;
}
```

```
char   s1[100];

char   *p;
  ...
→ p = strchr (s1, '/');
  printf ("%s", p+1);
```

ALLOCAZIONE DINAMICA

(stdlib.h)

```
void *   malloc( n° byte )

    int  *p;             int  p[15];

p=(int *) malloc ((15 * sizeof(int));

p[0]=7 ;
```


HEAP
15 byte
p
free(p);

```
char *strdup(char *s) {
    int n, i;
    n = strlen(s);
    char *p;
    p = (char *) malloc(n+1);
    for (i=0; i<=n; i++)
          p[i]= s[i];

    return p;

}
        char *p;                char p[10];

        p= ...                  p= ...
          :
          :
        p=...
```