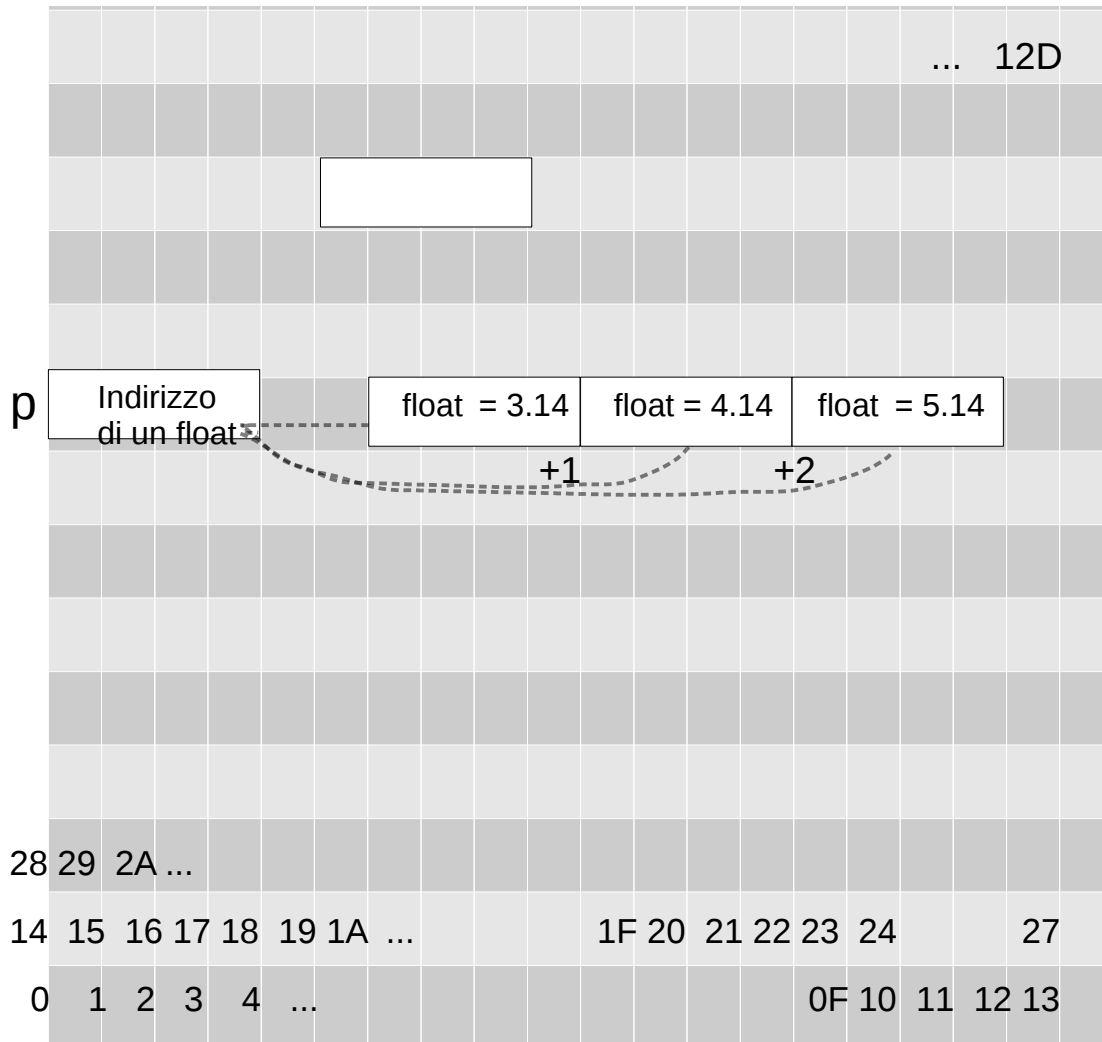


Strutture dinamiche in heap nel linguaggio C

E.Mumolo DIA

Salviamo dei valori float in un array a di float in heap:

Memoria heap



```
float a=4.5; stack
```

```
float *p;
```

```
p=malloc(3*sizeof(float));
```

```
*p=3.14;
```

```
*(p+1)=4.14;
```

```
*(p+2)=5.14;
```

In alternativa:

```
float *p=malloc(3*sizeof(float));
```

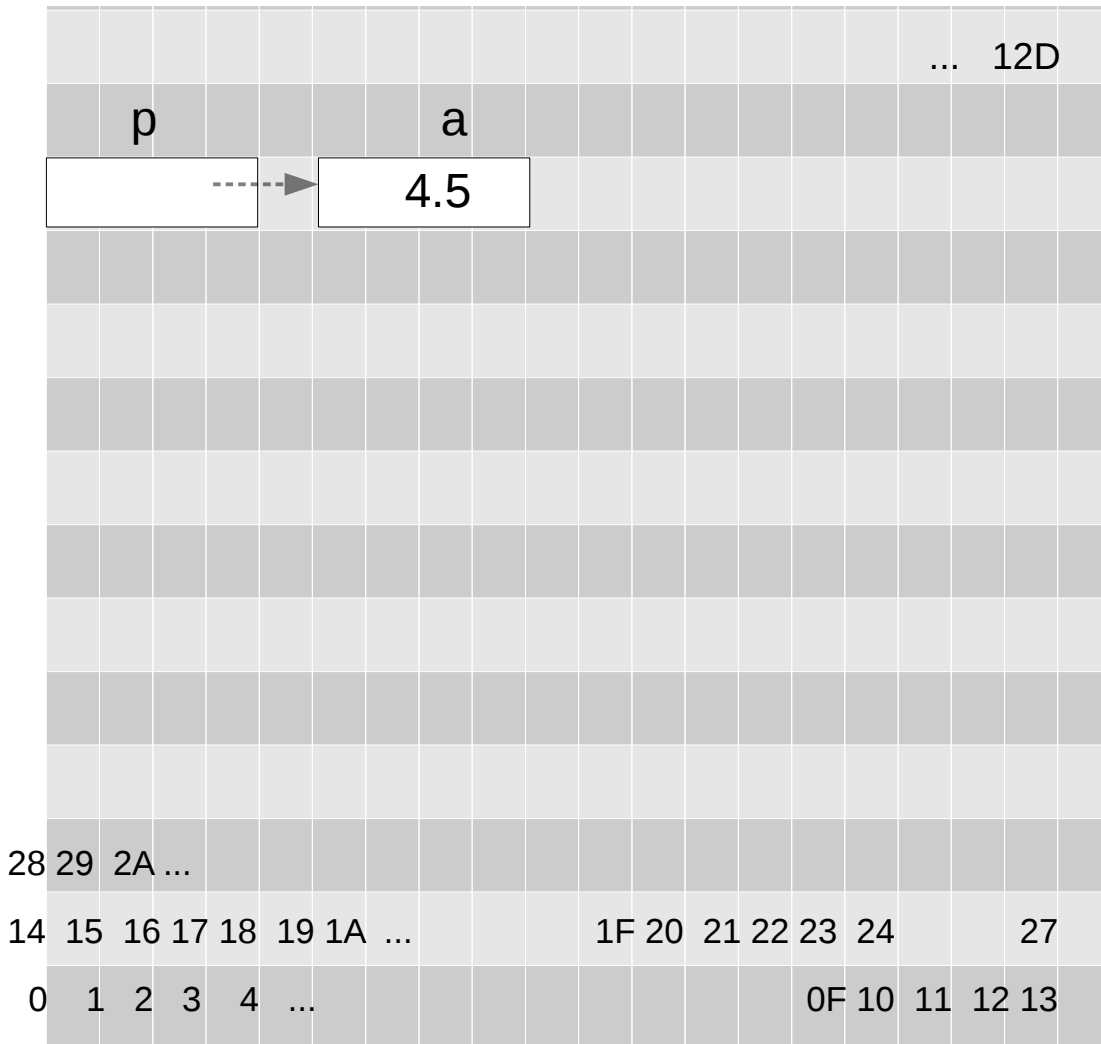
```
p[0]=3.14;
```

```
p[1]=4.14;
```

```
p[2]=5.14;
```

***p è un array p[] !!**

Memoria stack



```
float a=4.5; /* 4 byte in stack*/
```

```
float * p; /* 4 byte in stack */
```

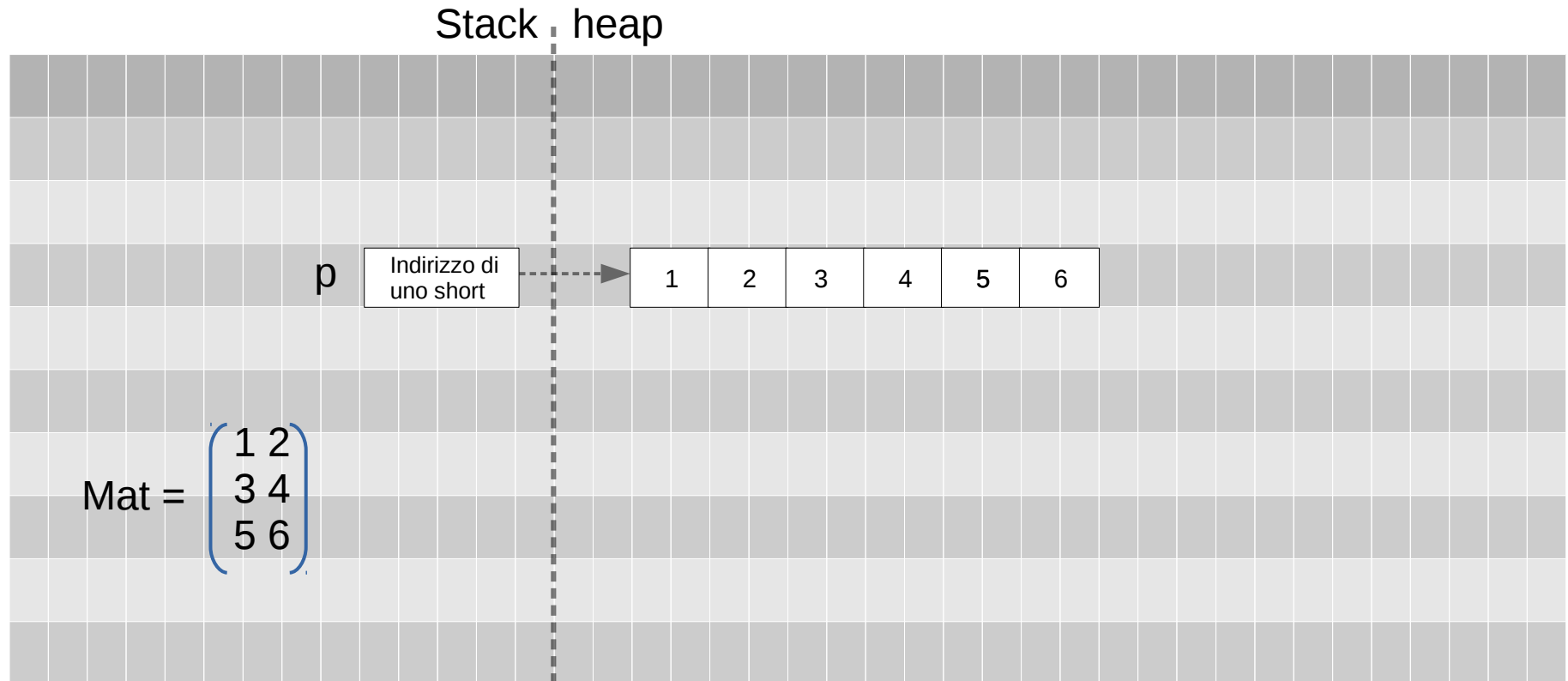
```
p = &a; /* 4 byte in stack*/
```

```
printf("%f allocato in %d\n", a, p);
```

```
printf("%f allocato in %d\n", *p, p);
```

```
printf("%f allocato in %d\n", a, &a);
```

Memorizziamo una matrice NxM in un array monodimensionale in heap:



```
short Mat[3][2]={{{1,2},{3,4},{5,6}}};
```

```
short *p=malloc(3*2*sizeof(short));
```

```
*(p+0) = 1;
```

```
*(p+1) = 2;
```

```
*(p+2) = 3;
```

```
*(p+3) = 4;
```

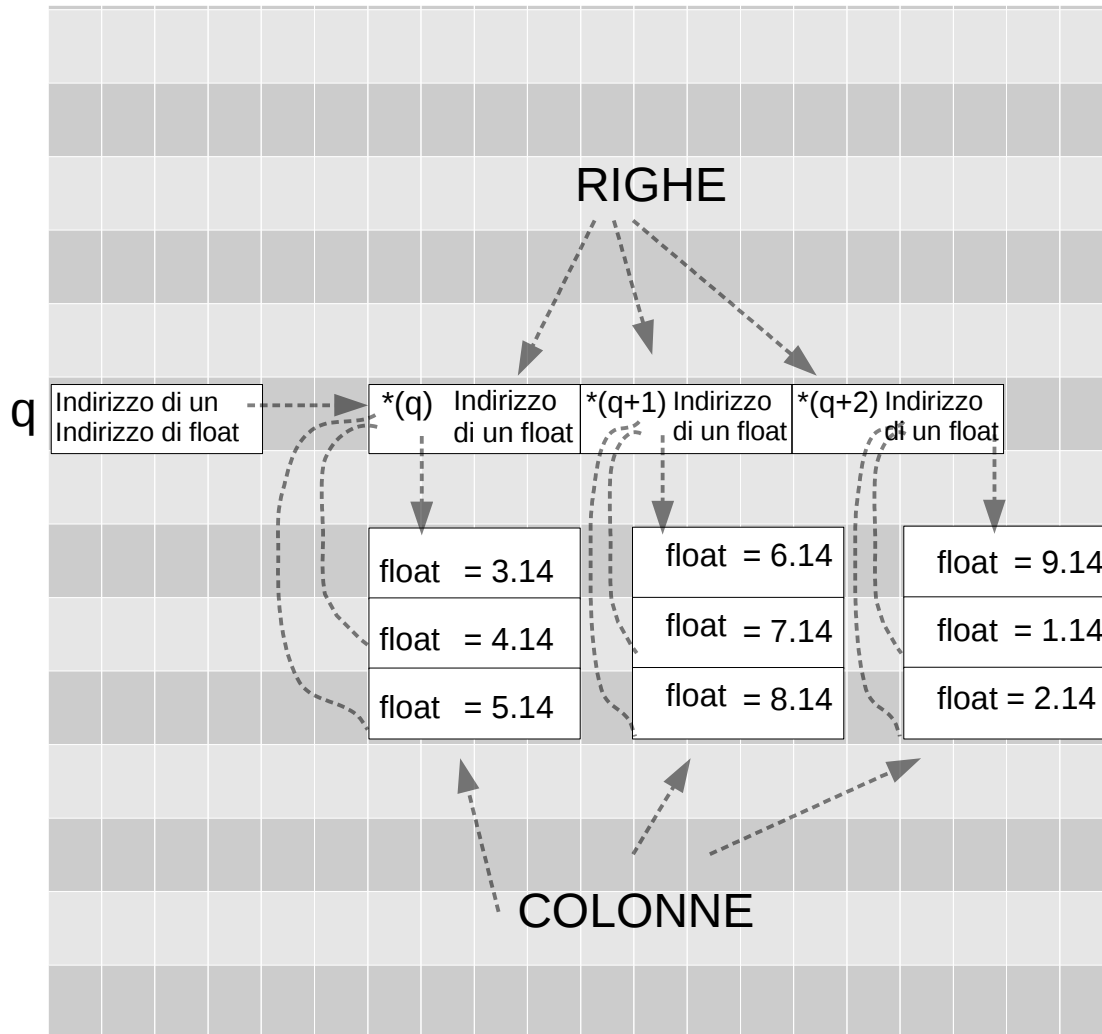
```
*(p+4) = 5;
```

```
*(p+5) = 6;
```

algoritmo

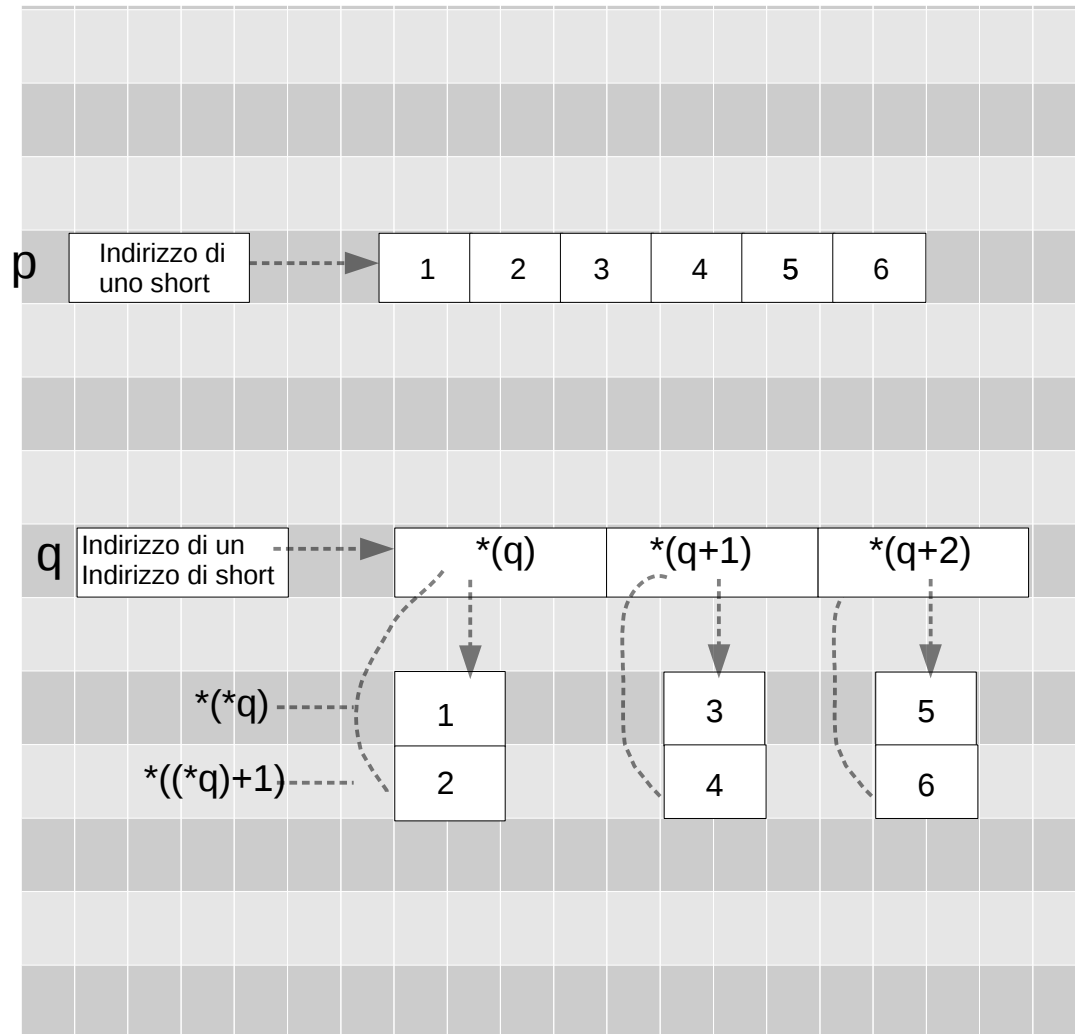
```
for(i=0;i<3;i++)  
  for(j=0;j<2;j++)  
    *(p+i*2+j)=Mat[i][j];
```

Salviamo dei valori float in un array Bidimensionale 3x3 di float in heap:



```
float **q;  
q=malloc(3*sizeof(float*));  
*q=malloc(3*sizeof(float));  
*(q+1)=malloc(3*sizeof(float));  
*(q+2)=malloc(3*sizeof(float));  
  
*(*q)=3.14;  
  
*(*q+1)=4.14;  
  
*(*q+2)=5.14;  
  
*(*q+1)=6.14;  
  
*(*q+1+1)=7.14;  
  
*(*q+1+2)=8.14;  
  
*(*q+2)=9.14;  
  
*(*q+2+1)=1.14;  
  
*(*q+2+2)=2.14;
```

Copia di una matrice 3x2 memorizzata in un array monodimensionale di short in un array Bidimensionale di short in heap:



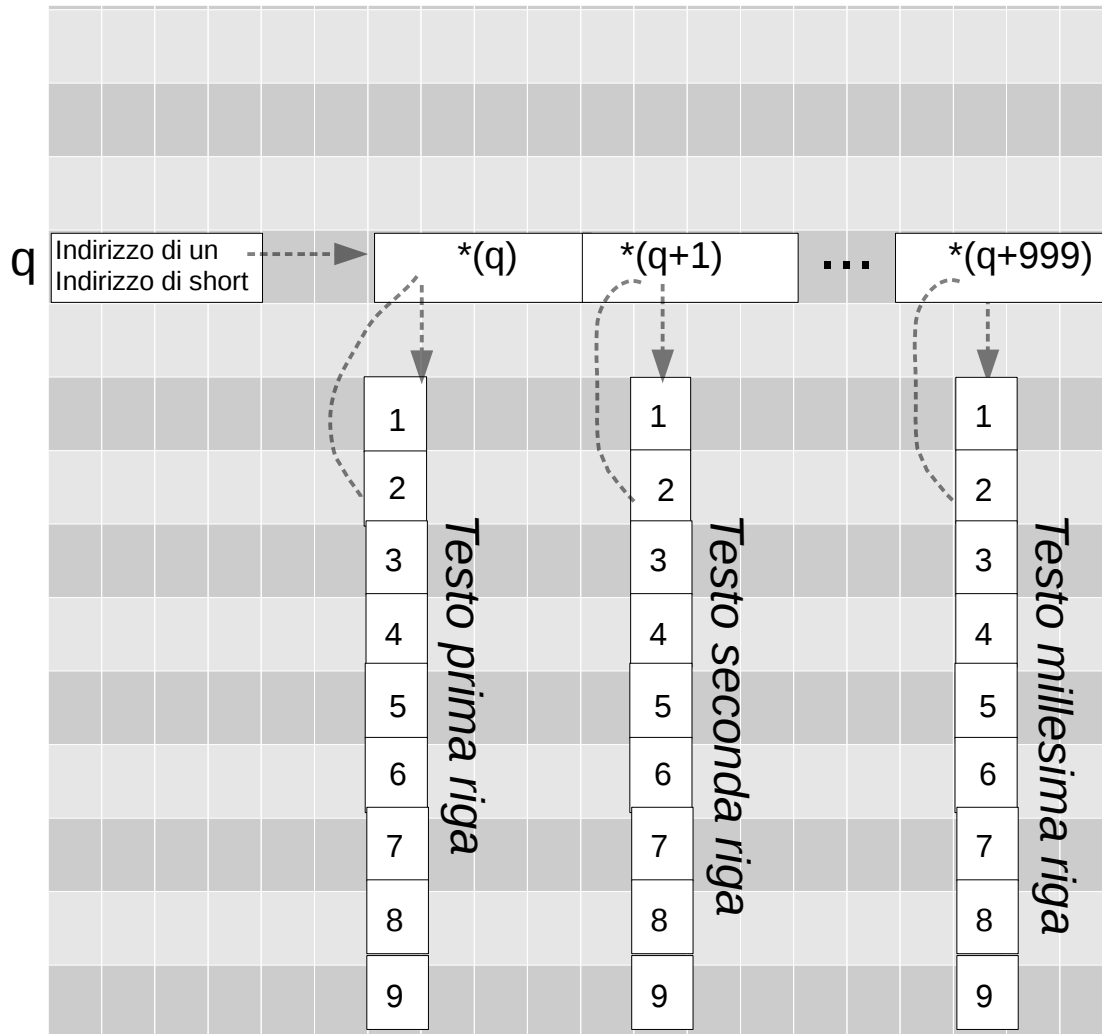
```
short *p;
p=malloc(6*sizeof(short));
/* inizializza i valori */
```

```
short **q;
q=malloc(3*sizeof(short*));
*(q)=malloc(2*sizeof(short));
*(q+1)=malloc(2*sizeof(short));
*(q+2)=malloc(2*sizeof(short));
*(*q)=*(p+0*2+0);
*((*q)+1)=*(p+0*2+1);
*(*q+1)=*(p+1*2+0);
*(*q+1)+1)=*(p+1*2+1);
*(*q+2)=*(p+2*2+0);
*(*q+2)+1)=*(p+2*2+1);
```

Algoritmo:

```
short **p=malloc(3*sizeof(short*));
short *p1=malloc(3*2*sizeof(short));
for(i=0;i<3;i++){
    p[i]=malloc(2*sizeof(short));
    for(j=0;j<2;j++)
        *(*p+i)+j)=*(p1+i*2+j);
}
```

Legge per righe un file di stringhe e le salva un array Bidimensionale di stringhe in heap:



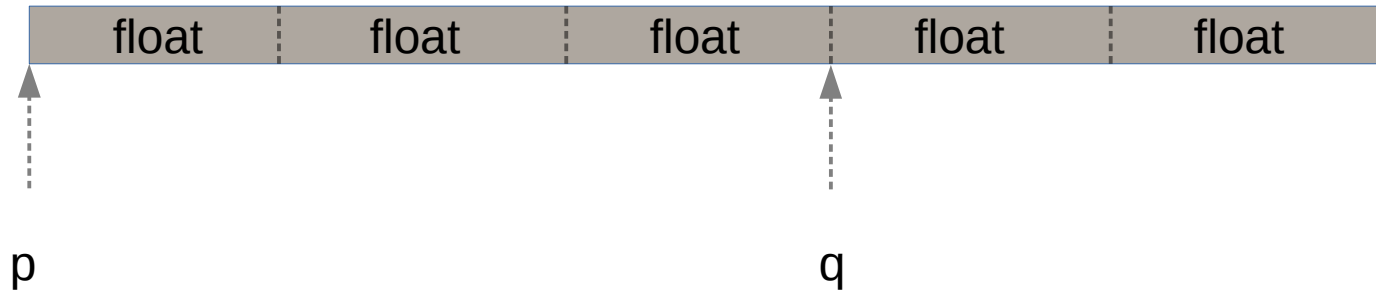
```
char **q;  
q=malloc(1000*sizeof(char*));  
fd=fopen("testmed.txt","r");  
fgets(testo, size, fp);  
*(q)=malloc(strlen(testo));  
strcpy(*(q),testo);  
fgets(testo, size, fp);  
*(q+1)=malloc(strlen(testo));  
strcpy(*(q+1),testo);  
...  
fgets(testo, size, fp);  
*(q+999)=malloc(strlen(testo));  
strcpy(*(q+999),testo);
```

Algoritmo:

```
char **data=malloc(1000);  
fp=fopen("testmed.txt","r");  
for(i=0;i<10;i++){  
    fgets(testo, size, fp);  
    data[i]=malloc(strlen(testo));  
    strcpy(data[i],testo);  
}  
close (fd);
```

Quanti elementi ci sono tra due puntatori?

```
float *p, *q;
```



I puntatori sono l'indirizzo del byte puntato cioè sono interi.
La differenza tra puntatori è il numero di byte compreso:
Il numero di byte compreso tra q e p è $(q - p)$

Quanti elementi sono compresi? Dipende dal tipo di dato.
Se sono float: numero elementi =

$$\frac{(q-p)}{\text{sizeof(float)}}$$

In generale:

$$\frac{(q-p)}{\text{sizeof(type)}}$$

Scrivere un main in C che legge un file di testo per righe e carica ogni riga in un array bidimensionale di stringhe. Cerca il punto iniziale della prima parola 'oggi' e la sostituisce con la parola 'ieri' direttamente sul file usando `lseek()` usando `write()` per scrivere sul file.

Usare la funzione `char *strstr(const char *str, const char *sub)` che cerca la prima occorrenza della stringa 'sub' in str. Usare `lseek(fd,offsett,SEEK_SET)` cioè dall'inizio del file.

Nel File:

Prima riga Seconda riga Terza riga ...
--

```
void main(int argc, char *argv[])
{
    int i,fd, numero_righe,newlen,oldlen, size=100, n;
    char **righe, *str="ieri", testo[100], *p;
    FILE *fp;
    /*spazio per 100 righe*/
    righe=(char**)malloc(100*sizeof(char*));
    numero_righe=atoi(argv[2]);

    fd=open(argv[1],O_WRONLY);
    fp=fopen(argv[1],"r");

    oldlen=0;
    for(i=0;i<numero_righe;i++){
        fgets(testo, size, fp);          /* leggi la riga del file*/
        newlen=strlen(testo);           /* lunghezza riga */
        righe[i]=malloc(strlen(testo)); /* alloca spazio per riga */
        strcpy(righe[i],testo);         /* copia stringa in array 2D */
        p = strstr(righe[i], "oggi");   /* cerca punto inizio */
        n = p-righe[i]+oldlen;          /* punto inizio nel file*/
        lseek(fd,n,SEEK_SET);           /* file pointer */
        write(fd,str,4);                /* rimpiazza stringa nel file */
        oldlen+=newlen;
    }
    close (fp);
    close(fd);
}
```

