

**Titolo:** gestione dei puntatori ad array dinamicamente allocati

Argomenti trattati:

- ✓ Esempi di uso di puntatori come parametri di function



Prerequisiti richiesti: programmazione C (array, puntatori, allocazione dinamica)

# Esempio 0a: passaggio dei parametri in C

In C i parametri di output vanno passati per riferimento (indirizzo)

```
#include <stdio.h>
void void_def_var(float* pt)  { *pt = 1.5; }

main()
{ float A=0.f, B=0.f; float* point;
  void_def_var(&A);
  printf("\nA = %g\n",A);
  point = &B; void_def_var(point);
  printf("\nB = %g\n",B);
}
```



A = 1.5



B = 1.5

## Esempio 0b: se la function restituisce un valore...

più semplice!

```
#include <stdio.h>
float def_var() {return 2.5f;}

main()
{float A=0.f, B=0.f; float* point;
  A=def_var();
  printf("\nA = %g\n",A);
  point = &B; *point=def_var();
  printf("\nB = %g\n",B);
}
```



A = 2.5

B = 2.5

# Esempio Oc: se il parametro di output è un puntatore

```
#include <stdio.h>

void def_var2(short* count, float* pA, float* pB, float** pt)
{
    ((*count)++) NON *count++ perché "*" e "++" hanno uguale priorità!
    if (*count%2 == 0)
    {
        *pt=pA;
        printf("\nEnter a float value: "); scanf("%f",*pt);
    }
    else
    {
        *pt=pB;
        **pt=(float)*count;
    }
}

main()
{
    short count=1; float A=0.f, B=0.f; float *point;

    def_var2(&count,&A, &B, &point);

    printf("\nA = %g,\tB = %g\n",A,B);
    def_var2(&count,&A, &B, &point);
    printf("\nA = %g,\tB = %g\n",A,B);
    ...
}
```

se count è **pari** modifica A,  
altrimenti modifica B



```
count = 2
Enter a float value: 1.5
A = 1.5,           B = 0
count = 3
A = 1.5,           B = 3
count = 4
Enter a float value: 2.7
A = 2.7,           B = 3
```

# Esempio 0d: se il parametro di output è un puntatore

```
#include <stdio.h>
float* def_var3(short* count, float* pA, float* pB)
{
    float* pt;
    (*count)++;
    if (*count%2 == 0)
    {
        pt = pA;
        printf("\nEnter a float value: "); scanf("%f",pt);
    }
    else
    {
        pt = pB;
        *pt=(float)*count;
    }
    return pt;
}
```

più semplice!

```
main()
{
    short count=1; float A=0.f, B=0.f; float *point;
    point = def_var3(&count,&A, &B);
    printf("\nA = %g,\tB = %g\n",A,B);
    point = def_var3(&count,&A, &B);
    printf("\nA = %g,\tB = %g\n",A,B); ...}
}
```



# Esempio 1: array statico mediante puntatore (nel main)

```
1  #include <stdio.h>
2
3
4  int main()
5  {
6      float A[10]; float *pA;
7      short i, N=10;
8
9      pA = A;
10
11     for (i=0; i<N; i++)
12         *(pA+i) = (float)i;
13
14     puts("\n array statico A =\n");
15     for (i=0; i<N; i++)
16         printf("  %g", *(pA+i));
17     puts("\n\n");
18 }
```

equivale a: `pA = &A[0];`

array statico A =

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

# Esempio 2a: array statico mediante puntatore (in function)

```
1  #include <stdio.h>
2
3  void crea_array(short N, float A[], float *pA)
4  {
5      short i;
6      pA=A;
7
8      for (i=0; i<N; i++)
9          *(pA+i) = (float)i;
10 }
11
12 int main()
13 {
14     float *pA, A[10];
15     short i, N=10;
16
17     //crea array(N,A,pA);
18     crea_array(N,A,&pA);
19
20     puts("\n array statico A =\n");
21     for (i=0; i<N; i++)
22         printf("  %g", *(pA+i));
23     puts("\n\n");
24 }
```

NON funzionano!

PERCHÉ?

Line	Message
18	In function 'main': error: cannot convert 'float**' to 'float*' for argument '3' to 'void crea_array'
=== Build finished: 1 errors, 0 warnings ===	

## Esempio 2b: array statico mediante puntatore (in function)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void crea_array(short N, float A[], float **pA)
4  {
5      short i;
6      *pA=A;
7
8      for (i=0; i<N; i++)
9          *(*pA+i) = (float)i;
10 }
11
12 int main()
13 {
14     float *pA, A[10];
15     short i, N=10;
16
17     crea_array(N,A,&pA);
18
19     puts("\n array statico A =\n");
20     for (i=0; i<N; i++)
21         printf("  %g", *(pA+i));
22     puts("\n\n");
23 }
```

OK! ora funzionano!

array statico A =

0 1 2 3 4 5 6 7 8 9



# Esempio 3a: costruire un array dinamico (in function)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void crea_array_dinamico1(int N, float **pt)
5  {
6      *pt = (float *)malloc(N*sizeof(float));
7
8      int i;
9      for (i=0; i<N; i++)
10         *(*pt+i) = (float)i;
11 }
12
13 int main()
14 {
15     short i, N=10; float *pA;
16
17     crea_array_dinamico1(N, &pA);
18
19     puts("\n array dinamico A mediante pA = \n");
20     for (i=0; i<N; i++)
21         printf("    %g", *(pA+i));
22
23     puts("\n\n");
24 }
```

array dinamico A mediante pA =
0 1 2 3 4 5 6 7 8 9

## Esempio 3b: costruire un array dinamico (in function)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  float *crea_array_dinamico1(int N)
5  {
6      float *pt;
7      pt = (float *)malloc(N*sizeof(float));
8
9      int i;
10     for (i=0; i<N; i++)
11         *(pt+i) = (float)i;
12     return pt;
13 }
```

```
14 int main()
15 {
16     short i, N=10; float *pA;
17     pA = crea_array_dinamico1(N);
```

```
18
19     puts("\n array dinamico A mediante pA = \n");
20     for (i=0; i<N; i++)
21         printf("  %g", *(pA+i));
22
23     puts("\n\n");
24 }
25
```

**più semplice!!!**

array dinamico A mediante pA =									
0	1	2	3	4	5	6	7	8	9