

```
#include <iostream>
using namespace std;
int main(){
    int num;
    int* pInteri;
    num = -587;
    pInteri = &num;
    cout << "valore di num : " << num <<endl;
    cout << "indirizzo di num : " << &num <<endl;
    cout << "valore del puntatore : " << pInteri <<endl;
    cout << "indirizzo del puntatore : " << &pInteri <<endl;
    cout << "valore variabile puntata:" << *pInteri <<endl;
}
```

programma : esp159\_pInteri.cpp

```
Prompt dei comandi
>>esp159_pInteri
valore di num : -587
indirizzo di num : 0x6dfefc
valore del puntatore : 0x6dfefc
indirizzo del puntatore : 0x6dfef8
valore variabile puntata:-587
>>
```

```
int main(){
    int A[7]={3, 5, -1, 8}; //-- vengono inizializzati solo : A[0],A[1],A[2 ],A[3]
    int* pA; //-- A[4],A[5],A[6] vengono posti a zero
    for (int i=0; i<7; i++)
        cout << "indirizzo di A["<<i<<"]:\t" << &A[i] <<endl;
    cout<<endl;
    for (int i=0; i<7; i++)
        cout << "indirizzo A+"<<i<<":\t" << A+i <<endl;
    cout<<endl;
    pA = A; //-- A rappresenta l'indirizzo di partenza dell'array A
    for (int i=0; i<7; i++)
        cout << "indirizzo pA+"<<i<<":\t" << pA+i <<endl;
    cout<<endl;

    cout << "val puntato da A :\t" << *pA <<endl;
    pA = &A[1]; //-- indirizzo dell'elemento di A di posto 1
    cout << "val puntato da &A[1]:\t" << *pA <<endl;
    cout << "val puntato da pA+2 :\t" << *(pA+2) <<endl;
    pA = A+3; //-- indirizzo dell'elemento di A di posto 3
    cout << "val puntato da A+3 :\t"<< *pA <<endl;
    cout << "\ncontenuto di A[] :\t" ;
    for (int i=0; i<7; i++)
        cout << A[i] <<" ";
    cout << "\nvalori puntati da pA:\t" ;
    pA = A;
    for (int i=0; i<7; i++)
        cout << *pA++ <<" "; //-- prima mostra valore puntato da pA (*)
    cout<<endl; //-- poi incrementa il puntatore (operatore ++)
```

programma: esp161\_pArrayInt.cpp

```
Prompt dei comandi
>>esp161_pArrayInt
indirizzo di A[0]: 0x6dfec8
indirizzo di A[1]: 0x6dfed0
indirizzo di A[2]: 0x6dfed4
indirizzo di A[3]: 0x6dfed8
indirizzo di A[4]: 0x6dfedc
indirizzo di A[5]: 0x6dfee0
indirizzo di A[6]: 0x6dfee4

indirizzo A+0: 0x6dfec8
indirizzo A+1: 0x6dfed0
indirizzo A+2: 0x6dfed4
indirizzo A+3: 0x6dfed8
indirizzo A+4: 0x6dfedc
indirizzo A+5: 0x6dfee0
indirizzo A+6: 0x6dfee4

indirizzo pA+0: 0x6dfec8
indirizzo pA+1: 0x6dfed0
indirizzo pA+2: 0x6dfed4
indirizzo pA+3: 0x6dfed8
indirizzo pA+4: 0x6dfedc
indirizzo pA+5: 0x6dfee0
indirizzo pA+6: 0x6dfee4

val puntato da A : 3
val puntato da &A[1]: 5
val puntato da pA+2 : 8
val puntato da A+3 : 8

contenuto di A[] : 3 5 -1 8 0 0 0
valori puntati da pA: 3 5 -1 8 0 0 0
>>
```

NOTE:  
ogni elemento di A occupa 4 byte (sono di tipo int)  
gli elementi di un array occupano posizioni adiacenti

indirizzo di A[0]  
6DFECC +  
4 =  
-----  
6DFED0  
indirizzo di A[1]