

ESERCIZI PER CASA:

scrivere il risultato dei printf (dove richiesto) nei seguenti codici:

```
*****
#include <stdio.h>
int main(){
    int* pc, c;
    c = 5;
    pc = &c;
    printf("%i", *pc); //Risultato:____
}
*****
*****

#include <stdio.h>
int main(){
    int* pc, c;
    c = 5;
    pc = &c;
    c = 1;
    printf("%i", c); //Risultato:____
    printf("%i", *pc); //Risultato:____
}
*****
*****

#include <stdio.h>
int main(){
    int* pc, c;
    c = 5;
    pc = &c;
    *pc = 1;
    printf("%i", *pc); //Risultato:____
    printf("%i", c); //Risultato: ____
}
*****
*****

#include <stdio.h>
int main(){
    int* pc, c, d;
    c = 5;
    d = -15;
    pc = &c;
    printf("%i", *pc); //Risultato:____
    pc = &d;
    printf("%i", *pc); //Risultato:____
}
*****
```

```

*****
#include <stdio.h>
int main() {
    int c = 5;
    int *p = &c;
    printf("%i", *p); //Risultato:____
    return 0;
}
*****
*****

#include <stdio.h>
int main(){
    int* pc, c;
    c = 22;
    printf("Indirizzo di c: %p\n", &c);
    printf("Valore of c: %i\n", c); //Risultato:____
    pc = &c;
    printf("Indirizzo a cui punta pc: %p\n", pc);
    printf("Valore della variabile puntata da pc: %i\n\n", *pc); //Risultato:____
    c = 11;
    printf("Indirizzo a cui punta pc: %p\n", pc);
    printf("Valore della variabile puntata da pc: %i\n\n", *pc); //Risultato:____
    *pc = 2;
    printf("Indirizzo di c: %p\n", &c);
    printf("Valore di c: %i\n\n", c); //Risultato:____
    return 0;
}
*****
*****

#include <stdio.h>
int main(){
    int x[5] = {1, 2, 3, 4, 5};
    int* ptr;
    // ptr e' assegnato all'indirizzo del terzo elemento
    ptr = &x[2];
    printf("*ptr = %i \n", *ptr); //Risultato:____
    printf("*ptr+1 = %i \n", *ptr+1); //Risultato:____
    printf("*ptr-1 = %i", *ptr-1); //Risultato:____
    return 0;
}
*****
*****

#include <stdio.h>
int main(void){
    char a[7]={'a', 'b', 'c', 'd', 'e', 'f', 'g'};
    printf("%c\n", *(a+3)); //Risultato:____
    return 0;
}
*****

```

```

*****
#include <stdio.h>
#define N 4
int main( )
{
    int *p;
    int val[N] = { 10, 12, -3, 4} ;
    p = val;
    for ( int i = 0 ; i<N ; i++ )
    {
        printf("val[%i]: il valore e' %i\n", i, *(p+i)); //Risultato:____
        printf(" e l'indirizzo e' %p\n", (p+i));
    }
    return 0;
}
*****
*****

#include <stdio.h>
#define N 5
int main(void){
    char pvocali[N];
    int i;

    pvocali[0] = 'A';
    pvocali[1] = 'E';
    *(pvocali + 2) = 'I';
    pvocali[3] = 'O';
    *(pvocali + 4) = 'U';

    for (i = 0; i < N; i++) {
        printf("%c ", pvocali[i]); //Risultato:____
    }
    printf("\n");
}
*****
*****

```

Data una matrice

```
int matrix[3][3]
```

dire che cosa restituiscono i seguenti puntatori

```

*(matrix)      => _____
*(matrix + 0)  => _____
*(matrix + 1)  => _____
**matrix       => _____
*(*(matrix + 0)) => _____
*(*(matrix + 0) + 0) => _____
*(*(matrix + 1)   => _____
*(*(matrix + 0) + 1) => _____
*(*(matrix + 2) + 2) => _____

```

```

*****
#include <stdio.h>
#include <stdlib.h>
#define ROWS 3
#define COLS 4
int main(void) {
    int total_cells=ROWS*COLS;
    // 2d array
    int num[3][4] = {
        {1, 2, 3, 4},
        {5, 6, 7, 8},
        {9, 10, 11, 12}
    };
    int *ptr = &num[0][0];
    printf("%i\n", *(ptr + 7)); //Risultato: ____
    return 0;
}
*****
*****

```

Esercizio:

Scrivere un programma che allochi in un vettore di dimensione $N=100$, 100 numeri interi random compresi tra $[0,20]$. Calcolare la somma dei numeri usando i puntatori (ossia bisogna definire un puntatore per la somma e usare il nome dell'array come puntatore per scorrere gli indici dell'array). Controllare che il programma restituisca i risultati corretti, testando il caso di un array contenente tutti 1.

Esercizio:

Scrivere un codice C in cui:

- Venga creata una matrice $N \times M$ (N e M interi a vostra scelta) i cui elementi sono inizializzati a zero.
- Si definisca un puntatore ad interi che punti all'elemento $(0,0)$ della matrice.
- Si scriva un algoritmo che sposti il puntatore di i righe e j colonne, dove i e j sono numeri interi random (con i tra $[0,N)$ e j tra $[0,M)$).
- Si ripeta l'operazione di spostamento del puntatore un numero di volte definito nel programma o introdotto dall'utente e si incrementi di 1 ogni elemento della matrice che viene puntato dal puntatore.
- Si stampino su schermo le coordinate di tutti i movimenti del puntatore (ossia le coppie (i,j)) e la matrice ottenuta alla fine di tutte le operazioni.
- Si controlli che la somma degli elementi della matrice sia uguale al numero di operazioni di spostamento del puntatore e in caso, si stampi un messaggio che dica che il programma ha eseguito le operazioni correttamente. Per effettuare la somma degli elementi della matrice usare il nome della matrice come puntatore.