

Carrazza	<input type="checkbox"/>
Mereghetti	<input type="checkbox"/>
Spoletini	<input type="checkbox"/>
Tamascelli	<input type="checkbox"/>

INFORMATICA – 28 gennaio 2019

Cognome _____ Nome _____ Matr _____

Lab (voto/quando) _____ / _____ Firma _____

1) Siano le definizioni

```
struct elem {
    int dato;
    int *p;
};

elem *X = new elem[100];
```

- Come assegnare al campo **dato** dell'elemento di indice **3** di **X** la somma del contenuto del campo **dato** dell'elemento di indice **0** di **X** più il contenuto della variabile puntata dal campo **p** dell'ultimo elemento di **X**?

- Scrivere l'istruzione ***((X+3) -> p) = 5** in una forma equivalente:

- Stabilire il valore di verità delle seguenti uguaglianze:

X[0].p == X -> p	vero <input type="checkbox"/>	falso <input type="checkbox"/>
(X+3) != &X[2] + 1	vero <input type="checkbox"/>	falso <input type="checkbox"/>

2) Sia la funzione

```
void swap(int &x, int &y) { int t = x; x = y; y = t; }
```

e sia il vettore **int A[7] = {1, 2, 3, 4, 5, 6, 7}**. Specificare il contenuto di **A** dopo l'esecuzione dei frammenti di codice:

```
for (int i = 0; i < 7; i++)
    swap(A[i], A[7-i-1]);
```

A = { _____ }

```
for (int i = 0; i < 7/2; i++)
    swap(A[i], A[7-i-1]);
```

A = { _____ }

3) Sia il frammento di codice

```
int X[10]; int k = 1;
for (int i = 0; i < 10; i++, k++)
    X[i] = 10/k;
```

Specificare il valore delle seguenti variabili dopo l'esecuzione del codice:

x[0]: _____ **x[7]:** _____ **x[9]:** _____ **k:** _____

4) La struttura

```
struct rett {  
    float infx, infy, supx, supy;  
}
```

rappresenta un rettangolo nel piano cartesiano il cui vertice in basso a sinistra ha coordinate **(infx,infy)** mentre quello in alto a destra ha coordinate **(supx,supy)**. Scrivete un frammento di codice che utilizzi la struttura **rett** per memorizzare **100** rettangoli chiesti l'uno dopo l'altro all'utente (non è necessario controllare la correttezza delle coordinate dei due vertici che definiscono il rettangolo). Successivamente il codice deve stampare i soli rettangoli contenenti l'origine degli assi cartesiani.

5) Scrivete la funzione

```
int occurrence(int *X, int *Y, int dim)
```

che accetta in ingresso due array di interi positivi **X** e **Y** di dimensione **dim** e restituisce quanti elementi di **X** occorrono in **Y**.

Esempio: assumendo **X = {5, 6, 28, 4, 13, 13}** e **Y = {5, 5, 7, 4, 13, 13}** l'invocazione **occurrence(X, Y, 6)** deve restituire **4**.