

Prova scritta Programmazione Procedurale - 15 Febbraio 2022 - FILA A

Nome e Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_

- 5 punti Cerchiare se vero dato `int a= 2, *p= &a;` e motivare brevemente la risposta su foglio protocollo.  
A. ++ (a++) non genera errore; **B. a++ = 2, a++;** contiene 2 effetti collaterali; C. \*&a è un *rvalue*;  
D. a++ && a++; genera warning; E. p ha tipo *int*;
- 6 punti Scrivere cosa stampa la seguente porzione di codice.

```
1 int a= 0x3e, i= -06, *b= &a;
2 for (int* p= &i; (a++, (*p)++) ? (++(*p),
  (a--)-1) : ((*p)+=3, a-1); (*p)++) {
3   --a;
4   printf("%d %d \n", a, *p);
5   if (*p > 3) {
6     a= (!(-a) && a++) ? 3 : 2;
7     break;
8   }
9 else continue; }
10 printf("%d\n", a);
```

61 -4  
60 -1  
60 4  
2

- 6 punti Data la seguente *struct* definire una funzione di nome *cancella\_per\_valore* che ha come parametro un intero *i* e cancella tutti gli elementi della lista in cui il campo *info* è uguale a *i*. Supporre che il puntatore ad inizio lista *pFirst* sia globale (non passarlo come parametro).

```
1 struct Node {
2   int info= 0;
3   struct Node* pNext= NULL;
4 }
```

- 6 punti Dire quali comandi provocano errore a causa del linker (e perché): 1) `gcc -c main.c`, 2) `gcc -o write write.c`, 3) `gcc write.c main.c -o output`, 4) `gcc -o main main.c`. In caso il punto 3) ritorni un errore, descrivere come può essere corretto. Elencare tutte le definizioni e dichiarazioni di variabile e funzione per ogni file. Cosa stampa *output*?

main.c	write.c
<pre>int count; int count= 3; void mywrite(int count); int main(void) { extern int count; do {   mywrite(count); }while(count &gt;=0); }</pre>	<pre>#include &lt;stdio.h&gt; int i= 1; static int count= 4;  void mywrite(int a) {   a--;   printf("%d\n", count= count - i); }</pre>

2) Errore, funzione main non definita  
4) Errore, funzione mywrite non definita  
1) e 3) non ritornano nessun errore

	Viene stampato
In main, count viene prima dichiarato (tentativo	3
definizione) e sotto definito, myWrite viene	2
dichiarata, main viene definita, count viene	1
dichiarata dentro main (corrisponde a count	0
globale in main.c)	-1
In write.c i viene definita, count viene definita,	-2 ed entra in un
myWrite viene definita, a viene definita	ciclo infinito

- 7 punti Cerchiare le affermazioni vere dato `int a[5]= {62, -62, 65536*2, SHRT_MAX, SHRT_MIN};`  
`short int *p = (short*) a;` `char *q = (char*) a;` `*(q+3) = 5;` `a[2] += 1;` sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e  $65536 = 2^{16}$  (valori rappresentati in *complemento a due* e *little endian*). Rappresentare anche la zona di memoria in cui è memorizzato l'array. A. `q+3 > &a[1];` **B. `*(q+15) - q[0] < 55;`** C. `!(q+8);`  
**D. `(&p[3] - &p[1])%2 == 0`**

## Esercizio 1

- A FALSO `a++` è un valore, mentre l'operatore `++` si applica ad un lvalue  
B VERO `a+=2` ha un effetto collaterale su `a`, `a++` ha un altro effetto collaterale su `a`  
C FALSO con `&*`, `&` restituisce prima l'indirizzo di `a` e poi `*` restituisce il valore contenuto a quell'indirizzo, il risultato è `a`, che è un lvalue  
D FALSO l'espressione non genera un warning perché c'è un sequence point tra i due effetti collaterali su `a`  
E FALSO `p` ha tipo "puntatore a int"

## Esercizio 5

```
/* Mappa di memoria
01111100
00000000
00000000    &p[1]
10100000    q+3

01000011    &a[1]
11111111
11111111    &p[3]
11111111

10000000    *(q+8)
00000000
01000000
00000000

11111111
11111110
00000000
00000000    *(q+15)
00000000
00000001
11111111
11111111
*/
```

Correzione simile allo scritto del 1 Febbraio 2022  
 $B * (q+15) - q[0]$  equivale a  $0 - 62 = -62$

## Esercizio 3

```
void cancella_per_valore(int key) {
    if (primo == NULL) {
        printf("Lista vuota\n");
        return;
    }
    struct Node* pScan = pFirst;
    if (pScan -> info == key) {
        struct Node* pTemp = pScan -> pNext;
        free(pScan);
        primo = pTemp;
    }
    while (pScan -> pNext != NULL) {
        if ((pScan -> pNext) -> info == key) {
            struct Node* pTemp = (pScan -> pNext) -> pNext;
            free(pScan -> pNext);
            pScan -> pNext = pTemp;
        }
        else
            pScan = pScan -> pNext;
    }
}
```