

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

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

1. 5 punti Elencare le conversioni di tipo (... da ... a). Sia *long int* che *int* sono rappresentati su 32 bit. Cosa stampa alla fine il programma?

```

1 int x = 0L, i = -2.5L;
2 char a= (char) 80, b= (char) 80, c= (
    char) 100;
3 a= (a*b) / c;
4 unsigned int limit = 10U;
5 long n = 20L;
6 if ( i < limit )
7     x = limit * n;
8 printf("%d %d\n", a, i);

```

**Tutte conversioni implicite eccetto dove riportato**  
**Linea 1: 0L da long in a int, -2.5L da long double a int**  
**Linea 2: 80 tre volte da int a char, conversioni esplicite**  
**Linea 3: a da char a int, b da char a int, c da char a int (integer promotion), risultato espressione da int a char**  
**Linea 6: i da int a unsigned int**  
**Linea 7: limit da unsigned int a unsigned long, n da long a unsigned long, risultato limit \* n da unsigned long a int**

**Stampa**  
**64 -2**

2. 6 punti Scrivere cosa stampa il seguente programma, sapendo che *a* si trova all'indirizzo 0x7ffee4399ffe.

```

1 int a= 0x1b, i= 3, *b= &a;
2
3 for (int* p= &i; (a-= 1) ? ((*p)++, --a)
    : ((*p)+=2, a); (*p)++) {
4     a= (a - i);
5     printf("%d %d OK\n", a, *p);
6     if (a <= 0) {
7         a= 1;
8         continue; }
9 }
10 printf("%d %p\n", a, ((short*) b) + 1);

```

**21 4 OK**  
**13 6 OK**  
**3 8 OK**  
**-9 10 OK**  
**0 0x7ffee439a000**

3. 6 punti Data la seguente *struct Node* definire su foglio protocollo una funzione di nome *ins\_se\_pari\_3\_posizione()* che prende un parametro di tipo *int* e che inserisce un nuovo elemento della lista in terza posizione solo se l'elemento in seconda posizione è pari (altrimenti non inserisce): se la lista originale è 7-4-18 e il valore passato è 11, la nuova lista sarà 7-4-11-18. Supporre un puntatore ad inizio lista globale di nome *pFirst*.

```

1 struct Node {
2     int info;
3     struct Node* pNext;
4 };

```

```

void ins_se_pari_3_posizione(int key) {
    if (pFirst != NULL && pFirst -> pNext != NULL) {
        if (pFirst -> pNext -> info % 2 == 0) {
            struct Node* pThird = pFirst -> pNext -> pNext;
            struct Node* pNew = (struct Node*) malloc(struct Node);
            pNew -> info = key;
            pNew -> pNext = pThird;
            pFirst -> pNext -> pNext = pNew;
        }
        else
            printf("L'elemento in seconda posizione non è pari\n");
    }
    else
        printf("La lista ha meno di due elementil\n");
}

```

4. **6 punti** Dire quali compilazioni provocano errore a causa del linker (e perché): 1) `gcc -c write.c`, 2) `gcc -o main main.c`, 3) `gcc -o write write.c`, 4) `gcc -c main.c`, 5) `gcc write.c main.c -o main`. In caso il punto 5) ritorni un errore, descrivere come può essere corretto. Dopo aver corretto l'errore, che tipo di *linkage* hanno `count`, `i`, e `mywrite`, ed in quale file sono definite? Cosa stampa il programma?

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

2) `mywrite` non definita, 3) funzione `main` e `count` non definiti  
 Nessun errore al punto 5) da correggere

count ext in entrambi i file	
mywrite ext in entrambi i file	
i ha no linkage	Stampa
	-1
	-6
count definito in main.c	-11
mywrite definito in write.c	-16
i definito in write.c	ciclo infinito

5. **7 punti** Cerchiare le affermazioni vere dato  $int\ a[5] = \{129, INT\_MIN, INT\_MIN | INT\_MAX, 262142, 262168\};$   $q[1] = 1;$   $short\ int\ *p = (short*)\ a;$   $char\ *q = (char*)\ a;$  sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e  $262144 = 2^{18}$  (valori rappresentati in *little endian* e complemento a due). Scrivere la mappa di memoria e giustificare le affermazioni (vere o false). Su foglio protocollo scrivere la mappa di memoria e giustificare brevemente tutte le affermazioni (sia se vere che false). A.  $(*(p+5) - p[6]) \% 1$  B.  $\&a[5] - (a+2) - q[1] - 2$  C.  $*((short*)(q+13)) == *((short*)(q+17))$

**A, B e C sono falsi**

100000001  
 (10000000) ← q[1]  
 q[1]

00000000  
 00000000

**B**  $\frac{2a[5] - (a+2)}{3} - q[1] - 2 == 0$   
 $\frac{2}{3} - 1$  FALSA

(00000000) ← a+1  
 (00000000) ← \*(p+2)

00000000  
 00000001

q[4] 11111111 ← a+2  
 11111111 ← a[2]  
 (11111111) ← \*(p+5) == p[5]  
 11111111

**A**  $\frac{*(p+5) - p[6]}{-1 - (-2)} \neq 1 == 0$   
FALSA

(01111111) ← p[6]  
 (11111111) ← ((sum+\*)(q+1))  
 (10000000)  
 (00000000) ← p[7]

**C**  $\frac{*((sum+*)(q+13))}{1023} == \frac{*((sum+*)(q+17))}{1024}$   
FALSA

(00011000) ← q[16]  
 (00000000) ← ((sum+\*)(q+17))  
 (00100000) ← q[18]  
 00000000

XXXXXXXX ← da[5]  
 P+10  
 XXXXXXXX  
 XXXXXXXX  
 XXXXXXXX