

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

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

1. **5 punti** Elencare le conversioni di tipo implicite (... da ... a). Dato  $USHRT\_MAX = 65535$ , scrivere il valore finale della variabile *c*. Che precisione ha *b*?

```
1 int fun1(long x) {
2     char a= 'e';
3     return (x - a); }
4
5 int main(void) {
6     unsigned short a = 'd';
7     double b = fun1(a);
8     unsigned short c = - (int) (b+2);
9 }
```

linea 6: 'd' da int a unsigned short  
 linea 7: a da unsigned short a long  
 linea 7: ritorno funzione da int a double  
 linea 8: 2 da int a double, risultato da int a unsigned short

linea 2: 'e' da int a char  
 linea 3: a da char a long  
 linea 3: (x-a) da long a int

Il Valore finale di *c* è 65535: -1 + (USHRT\_MAX + 1)  
 b ha precisione doppia, 15 cifre

2. **6 punti** Scrivere cosa stampa la seguente porzione di codice.

```
1 int a= 0xfe - 0361;
2 printf("%d\n", a);
3 while(a > 10 ? !!!0: !(a+9, a-=10)) {
4     if (a + 2 < 0x10) {
5         a--;
6         printf("%d\n", a);
7         continue;
8     }
9     break; }
10 !(a) || a++;
11 printf("a: %d\n", a);
```

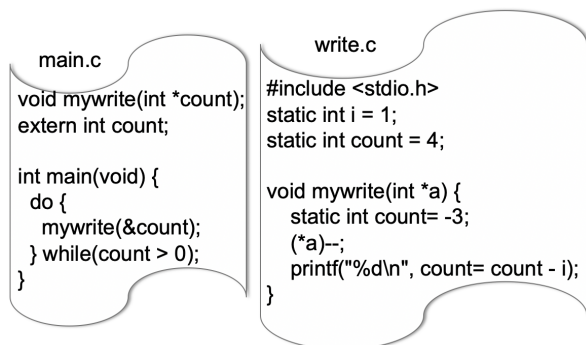
13  
 12  
 11  
 10  
 a: 0

3. **6 punti** Data la seguente *struct* definire una funzione di nome *stampa\_pari* che prende come parametro una lista e stampa su video il valore del campo *info* per tutti gli elementi in posizione pari della lista: supposto il primo elemento in posizione 1, stampa invece quelli in posizione 2, 4, 6, etc (se presenti...).

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

vedere sotto

4. **6 punti** Dire quali compilazioni provocano errore a causa del linker (e perché): 1) `gcc -o main main.c`, 2) `gcc -o write write.c`, 3) `gcc write.c main.c -o main`. In caso il punto 3) 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?



1) errore: manca definizione mywrite e count  
 2) errore: manca definizione main  
 3) errore: count è dichiarato in main.c ma ha linkage interno in write.c e quindi non visibile in main.c  
 Per risolvere, cancellare static davanti a static int count = 4;  
 - count ha linkage esterno in main.c  
 - i ha linkage interno in write.c (definita)  
 - count globale ha linkage esterno in write.c (definita),  
 dopo aver corretto il punto 3)  
 - count locale alla funzione ha no linkage (definita)  
 - mywrite() definita in write.c

Stampa:  
 -4  
 -5  
 -6  
 -7

5. 7 punti Cerchiare le affermazioni vere dato  $\text{long long } a[5] = \{9, 131072, 131053, -2, \text{LLONG\_MAX}\};$   
 $\text{int } *p = (\text{int} *) a;$   $\text{short int } *q = (\text{short} *) a;$  sapendo che i tre tipi usati occupano 8, 4, e 2 byte, e  $131072 = 2^{17}$  (valori rappresentati in *complemento a due* e *little endian*). Rappresentare la zona di memoria in cui è memorizzato l'array. A.  $q+3 > \&p[1];$  B.  $*(q+5) - *(q+9) == 1;$  C.  $*(q+8) > 0;$  D.  $((\text{int})(\&a[1]) - (\text{int})(q+1)) < 6.$  E.  $(p+7) - (p+2) > 0x000000f$

#### Mapa di memoria esercizio 5:

```

10010000
00000000
00000000    q+1
00000000
00000000    &p[1]
00000000
00000000    q+3
00000000

00000000    &a[1] e p+2
00000000
01000000    *(q+5)
00000000
00000000
00000000
00000000
00000000

10110111    *(q+8)
11111111
10000000    *(q+9)
00000000
00000000
00000000
00000000
00000000

01111111
11111111
11111111
11111111
11111111    p+7
11111111
11111111
11111111

11111111
11111111
11111111
11111111
11111111
11111111
11111111
11111110

```

#### Esercizio 3

```

void stampa_pari(struct Node* lista) {
    int count = 1;
    if (lista == NULL) {
        printf("Lista vuota!\n");
        return;
    }
    struct Node* pScan = lista;
    while (pScan != NULL) {
        if (count % 2 == 0)
            printf("Valore: %d\n", pScan->info);
        count++;
        pScan = pScan->pNext;
    }
}

```

#### Esercizio 5

- A vero: il primo puntatore è due byte più grande in memoria rispetto al secondo  
 B vero:  $2^{-1} == 1$   
 C falso: il bit significativo è 1, quindi sicuramente negativo (vale -19)  
 D falso: la differenza in byte tra i due puntatori è 6  
 E falso: 5 non è maggiore di 15