

Nome e Cognome: _____

Matricola: _____

1. 5 punti Elencare le conversioni di tipo (... da ... a). Dato $USHRT_MAX = 65535$, se possibile scrivere il valore finale esatto della variabile b sapendo che il carattere a ha valore 97 (in ASCII) e le altre lettere seguono in ordine alfabetico. Spiegare perché possibile scrivere il valore, oppure non possibile (causa precisione).

```

1 long long int fun2(long int p) {
2     return (p + 'f' - 'b'); }
3
4 long int fun1(long p) {
5     char a= 'h';
6     return fun2(p - a + 'c'); }
7
8 int main(void) {
9     unsigned short a = -2LL;
10    float b= fun1(a);
11    printf("%f\n", b);
12 }
```

b vale alla fine 65533.0
 un valore sicuramente rappresentabile con un float perché ha 5 cifre significative, meno della precisione del float (6 cifre)

linea 9: -2LL da long long a unsigned short
 linea 10: a da unsigned short a long
 linea 5: 'h' da int a char
 linea 6: a da char a long, 'c' da int a long
 linea 2: 'f' da int a long, 'b' da int a long
 linea 2: p + 'f' - 'b' da long int a long long (ritorno funzione fun2)
 linea 6: risultato fun2 da long long a long (ritorno funzione fun1)
 linea 10: risultato fun1 da long a float

linea 9: a è inizializzato a 65534, -2 + (USHRT_MAX + 1)

2. 6 punti Scrivere cosa stampa il seguente programma e motivare su foglio protocollo.

```

1 int a= 0x2d - 036;
2 printf("%d\n", a);
3 while(-a % 2 ? (a++, a / 2): !!1) {
4     printf("%d \n", a);
5     if(a == 2 && a++) {
6         a <<= 1;
7         break; }
8     a >>= 1;
9     continue;
10 }
11 printf("a: %d\n", a);
```

15
 16
 8
 4
 2
 a: 6

3. 6 punti Data la seguente *struct Node* definire su foglio protocollo una funzione di nome *cancella_se_3_posizione()* che prende un parametro di tipo *int* e che cancella l'elemento della lista in terza posizione se il suo campo *info* è uguale al valore passato come parametro: se la lista originale è 7-4-11-18 e il valore passato è 11, la nuova lista sarà 7-4-18. Supporre un puntatore ad inizio lista globale di nome *pFirst*.

```

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

Esercizio 3:

```

void cancella_se_3_posizione(int key) {
    if (pFirst != NULL && pFirst -> pNext != NULL && pFirst -> pNext -> pNext != NULL) {
        if (pFirst -> pNext -> pNext -> info == key) {
            pSecond = pFirst -> pNext;
            pFourth = pSecond -> pNext -> pNext;
            free(pSecond -> pNext);
            pSecond -> pNext = pFourth;
        }
        else
            printf("L'elemento in terza posizione è diverso da %d\n", key);
    }
    else
        printf("La lista ha meno di tre elementil\n");
}
```

4. 5 punti Per ogni identificatore di variabile e funzione scrivere se è definito o dichiarato, ed il suo linkage.

```

1 typedef int intero;
2 intero a= 5;
3 intero a;
4 extern int b;
5 extern int c = 1;
6 extern int cmp(float , float);
7
8 int* my_func(int d) {
9     static double e= 4.0;
10    double* f= &e;
11    register int q= 4;
12    // Altri comandi
13 }
```

linea 2: a definita, external linkage
 linea 3: a tentativo di definizione che rimane dichiarazione, external linkage
 linea 4: b dichiarata, external linkage
 linea 5: c definita, external linkage
 linea 6: cmp dichiarata, external linkage
 linea 8: my_func definita, external linkage
 linea 8: d definita, no linkage
 linea 9: e definita, no linkage
 linea 10: f definita, no linkage
 linea 11: q definita, no linkage

5. 8 punti Cerchiare le affermazioni vere dato $int\ a[5] = \{INT_MAX - 7, 1287, INT_MIN + 528, -10, 312\}$; $short\ int\ *p = (short\ int*)\ a$; $char\ *q = (char*)\ a$; $p[3] = SHRT_MAX$, $p[5] += 2048$, $q[18] = \sim q[19]$; sapendo che i tre tipi usati occupano 4, 2, e 1 byte, con valori rappresentati in *little endian* e complemento a due. Su foglio protocollo scrivere la mappa di memoria e giustificare brevemente tutte le affermazioni (sia se vere che false).

A. $(q[0] | q[1]) + q[17]$ B. $*((short*)\ &q[17]) > 0$ **C.** $((int)(q + 13) - (int)(a + 1)) - q[1] \% 3$

00011111 q[0]
11111111 q[1]
11111111
11111110

11100000 ← a+1
10100000
11111111
11111110

00001000
01000000
00000000
00010001

01101111 ← q+13
11111111
11111111
11111111

00011100
10000000
11111111
00000000

B: *((short*) &q[17]) è minore di zero, visto il bit più significativo uguale a 1 FALSA

A: q[0] | q[1] = 11111111
che equivale a -1
q[17] == 1
-1 + 1 == 0 FALSA

C: (int)(q+13) - (int)(a+1) == 9
q[1] == -1
9 - (-1) % 3 == 1 VERA