

Nome e Cognome: _____

Matricola: _____

1. **3 punti** Riportare le conversioni di tipo implicite e scrivere quanto valgono alla fine le variabili a , x , i ($INT_MAX == 2147483647$).

$a == 4$, $x == 0$, $i == -2147483646$ ($INT_MAX + 1$ fa INT_MIN , $+3$ fa $INT_MIN + 3 == -2147483645$ dato che $INT_MIN == -2147483648$)

```
1 int x = 0U, i = (int) (INT_MAX + 4);
2 char a= (char) 10, b= (char) 40, c= (
    char) 100;
3 a= (a*b) / c;
4 unsigned int limit = 10U;
5 long n = 5L;
6 if ( i < limit )
7     x = 3.0;
```

linea 1: 0U da unsigned int a

linea 3: a, b, c da char a int per integer promotion, il risultato da int a char

linea 6: i convertito da int a unsigned int

linea 7: 3.0 da double a int

La condizione alla linea 6 è falsa: il valore di i diventa -2147483645 + (UINT_MAX + 1) che è sicuramente maggiore del valore di limit

regola 1 slide 11 su conversione di tipo, e slide 20

2. **4 punti** Scrivere cosa stampa il seguente programma.

```
1 int i, j, s, r=6, st=0;          12 } else {
2 for(i = 1; i <= r; i++) {      13     for(j = 0; j < 9; j++) {
3     if(i <= 4){                14         if((int)(j/3) == 1)
4         for(s = 1; s <= 5-i; s++)15             printf("o");
5             printf("-");        16         else
6         while(st != (2*i - 1)) {17             printf("*");
7             printf("*");        18     }
8             st++;              19     printf("\n");
9     }                          20 }
10     st=0;                     21 }
11     printf("\n");
```

```
----*
---***
--*****
_*****
***ooo***
***ooo***
```

3. **3 punti** Scrivere cosa stampa il seguente programma.

```
1 int a= 0xb1 - 0251;
2 while ((a-=1)? a++:a--) {
3     a-=1;
4     printf("OK\n");
5 }
6 int b= 0xfc;
7 b+= a && a++;
8 printf("%d %d\n", a, b);
```

**OK
OK
OK
OK
OK
OK
OK
0 253**

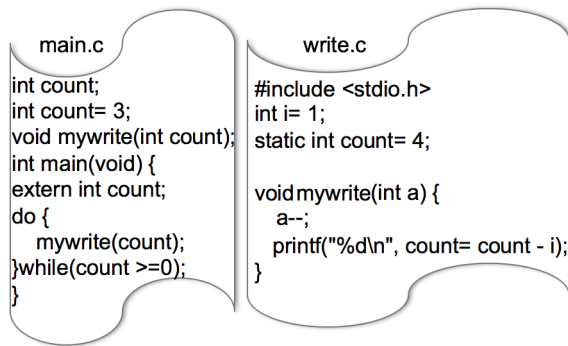
4. **4 punti** Su foglio protocollo, scrivere la definizione di una funzione *inverti_righe12* che prende come parametri una matrice di interi m , le *righe*, le *colonne*, e scambia gli elementi della prima riga con quelli dell'ultima riga (controllare prima che la matrice abbia per lo meno due righe).

5. **3 punti** Data la seguente *struct* definire una funzione di nome *print_dispari* che prende come parametro una lista e stampa su video il valore del campo *info* per tutti gli elementi in posizione dispari della lista: supposto il primo elemento in posizione 1, stampa gli elementi in posizione 1, 3, 5, etc (se presenti...).

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

6. **4 punti** Dire quali compilazioni 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 main`, 4) `gcc -o main main.c`. In caso il punto 3) ritorni un errore,

descrivere come può essere corretto. Infine, che tipo di *linkage* hanno *count*, *i*, e *mywrite*? Cosa stampa il programma *output*? Elencare tutte le definizioni e dichiarazioni in ogni file.



2) Errore, non esiste file main da compilare insieme a file main.c
4) Manca la definizione di mywrite

1) non ritorna nessun errore

In main, count viene prima dichiarato (tentativo definizione) e sotto definito, myWrite viene dichiarata, main viene definita, count viene dichiarato dentro main
In write.c i viene definita, count viene definita, myWrite viene definita

Viene stampato 3
2
1
0
-1
-2 e così via

7. **3 punti** Per ogni identificatore di variabile e funzione scrivere se è definito o dichiarato, ed il suo linkage; riportare eventuali tentativi di definizione.

```

1 #define A 3
2 int a= A;
3 int a;
4 extern int b;
5 extern int cmp(float , float);
6 static double area(double);
7
8 int* my_func(int c) {
9     static double e= 4.0;
10    double* f= &e;
11    register int q= 4;
12    extern int a;
13    // Altri comandi ...
14 }
    
```

a linea 2 definita/ext
a linea 3 tentativo di def rimane dichiarata/ext, b dichiarata/ext
cmp dichiarata/ext
area dichiarata/int
my_func definita/ext
c definita/nolinkage
e definita/nolinkage
f definita/nolinkage
q definita/nolinkage
a linea 12 dichiarata/ext (a globale)

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

A. $(q[0] | q[1]) + q[17]$ B. $((\text{short } *) \&q[17]) > 0$ C. $((\text{int})(q + 13) - (\text{int})(a + 1)) - q[1] \% 3$ D. $((p + 7) \&p[9]) \% 5$ E. $q[9] + q[11] > -120$ F. $\sim (\&p[5] - \&p[1] - q[9] * 2)$

9. **1 punti** Per ciascuna delle seguenti fasi di *gcc*, dare l'ordine da 1 a 4 in cui vengono eseguite. ④ linker; ① preprocessor; ② compiler; ③ assembler.

10. **2 punti** Su foglio protocollo, descrivere le proprietà di una variabile descritte dal suo tipo.

Un tipo determina

- 1) Quanto spazio una variabile occupa in memoria
- 2) I valori che una variabile può assumere
- 3) Le operazioni che possono essere eseguite sulla variabile

00011111 q[0]
 11111111
 11111111 — &p[1]
 11111110

11100000 — a+1
 10100000
 11111111
 11111110

00001000
 01000000 — &p[5]
 00000000
 00010001

q+13 — 01101111
 11111111
 11111111
 11111111

*(p+7) — 00011100
 10000000
 11111111
 00000000

p[9]

A: $q[0] \mid q[1] = 11111111$
 che equivale a -1
 $q[17] == 1$
 $-1 - (-1) == 0$ FALSA

C: $(\text{int})(q+13) - (\text{int})(a+1) == 9$
 $q[1] == -1$
 $9 - (-1) \% 3 == 1$ VERA

D: $11111111 \mid 11111111 == *(p + 7)$
 $11111111 \mid 00000000 == p[9]$
 in and bit a bit il risultato è p[9] che equivale a $255 \% 5 == 0$ FALSA

E: $q[9] == 2$,
 per q[11] applicare l'algoritmo slide 27 sui tipi: sottrarre 1 (11100001) e
 poi flippare i bit (00011110) ci dice che q[11] vale -120
 $2 + (-120) == -118 > -120$ VERA

B: $*((\text{short}^*) \&q[17])$ è minore di zero, visto il bit più significativo uguale a 1 VERA

F: $\&p[5] - \&p[1] == 4$ (restituisce il numero di short int tra i due puntatori)
 $q[9] * 2 == 4$
 $\&p[5] - \&p[1] - (q[9] * 2) == 0$, che negato bit a bit porta a VERA