

Nome e Cognome: _____

Matricola: _____

1. **3 punti** Riportare le conversioni di tipo implicite, dare il valore di finale di d dato $INT_MAX = 2147483647$. In caso non sia possibile stabilire il valore di d , spiegare perché. Scrivere la gerarchia dei tipi interi ed in virgola mobile (*tipo1* ">" *tipo2* significa che *tipo1* è più alto in gerarchia).

```
1 double f(float a) {return (a - 1);}
2
3 int main(void) {
4     unsigned a = 3;
5     int b = -1;
6     float c = f((b < a) ? b : a);
7     float d = INT_MAX - 3;
8 }
```

VEDI FILA B

2. **3 punti** Scrivere cosa stampa il seguente programma.

```
1 int r = 6, c = 5, s, i, j;      8      c = 2;
2 for(i=0; i<r; i++) {           9      else
3     for(s=1; s <= r-i; s++)    10      c = c*(i-j+1)/j;
4     printf("-");              11     printf("-%d", c);
5                                12 }
6     for(j=0; j <= i; j++) {    13 printf("\n");
7     if (j==0 || i==0)         14 }
```

```
-----2
-----2-2
-----2-4-2
-----2-6-6-2
---2-8-12-8-2
--2-10-20-20-10-2
```

3. **3 punti** Scrivere cosa stampa il seguente programma, sapendo che a si trova all'indirizzo $0x7ffce4399ffe$.

```
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
-3 10 OK
0 0x7ffce4399ffe
```

4. **4 punti** Su foglio protocollo, scrivere la definizione di una funzione *matrix_sum_diagonals* che somma gli elementi sulle due diagonali di una matrice $n \times n$ (matrice di *int*) e ritorna un valore intero 1 se la somma è uguale, 0 o altrimenti. Se la matrice è *identica* (tutti gli elementi della diagonale principale sono costituiti dal numero 1, mentre i restanti elementi sono 0), la funzione stampa a schermo un messaggio appropriato.
5. **3 punti** Su foglio protocollo, descrivere cosa sono i *lvalue* e i *rvalue*, compresi i *lvalue* non modificabili. Fornire esempi di espressioni che li contengono. VEDI FILA B
6. **3 punti** Data la seguente struttura, definire una funzione *ricorsiva* di nome *recursive_list* che crea in memoria dinamica una lista di strutture di lunghezza l , valore di tipo *unsigned int* preso come parametro.

```

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

```

VEDI PUA B

7. **4 punti** Scrivere cosa stampa il seguente programma, sapendo che le variabili a , b , c in *main* sono memorizzate rispettivamente agli indirizzi $0x7ffeeb7d2964$, $0x7ffeeb7d2970$ e $0x7ffeeb7d2974$.

```

1 static int a= 3;
2 int fl(int* x, int y, int* z) {
3     int res= 1;
4     if (a-- > 0) { \\ a meno meno
5         a ? res= ++y + ++(*z), (y+= *x) : 2;
6         printf("%d %d %p %d\n", *x, y, z, res);
7         res+= fl(&y, *x, z);
8         return (res);
9     }
10    else
11        return res; }
12
13 int main(void) {
14     int a= 3, b= 4, c= 4;
15     b= fl(&b, a, &c);
16     printf("%d %d %d\n", a, b, c);
17 }

```

VEDI PUA B

8. **4 punti** Per ogni identificatore di variabile e funzione ad ogni linea, scrivere se corrisponde ad una definizione o dichiarazione, ed il tipo linkage associato. Scrivere un file *file1.c* compilabile senza errori con il file sottostante (di nome *file2.c*) attraverso il comando `gcc -o eseguibile file1.c file2.c`. In *file1.c* si deve chiamare la funzione *my_func*. *Suggerimento: evitare errori di linkage, la logica della funzione non è di interesse.*

```

1 static double a;
2 extern double area(double);
3 typedef int pippo;
4 int b;
5 extern pippo b= 1;
6 extern double c;
7
8 extern void my_func(int d, int* e) {
9     static float f;
10    double g, h[d];
11    extern pippo b;
12    g= area(c); // Comandi...
13 }

```

VEDI PUA B

9. **4 punti** Cerchiare le affermazioni vere dato $\text{int } a[5] = \{129, \text{INT_MIN}, \text{INT_MIN} | \text{INT_MAX}, 262142, 262168\};$ $q[1] = 1$; $\text{short int } *p = (\text{short} *) a$; $\text{char } *q = (\text{char} *) a$; 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). A. $a[2] + *(q+1)$ B. $(*(p+5) - p[6]) \% 1$ C. $((\text{int})(p+10) - (\text{int})(a+1)) + q[16] - q[18] \% 6$ D. $\&a[5] - (a+2) - q[1] - 2$ E. $*(p+2) ? p[7] : !(p[4] + p[5] + p[7])$ F. $*((\text{short}*)(q+13)) == *((\text{short}*)(q+17))$

TUTTI FALSI

VEDI PUA B