

Nome e Cognome: _____

Matricola: _____

1. **3 punti** Riportare tutte le conversioni di tipo implicite. Dare il valore di finale di d dato $UINT_MAX = 4294967295$. In caso non sia possibile stabilire il valore di d , spiegare perché.

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

VEDI FIA A

2. **3 punti** Scrivere una funzione `main()` che disegna su terminale il seguente triangolo di lettere. Il numero di righe è preso come input da tastiera. *Suggerimento: utilizzare due cicli innestati.*

```
A
ABA
ABCA
ABCDCBA
ABCDEDCBA
```

VEDI FIA A

3. **3 punti** Scrivere cosa stampa il seguente programma, sapendo che a si trova all'indirizzo `0x7fee9289918`.

```
1 int a = 0xac, i = -13, *b = &a;
2
3 for (int* p = &i; (a -= 3, a--, (*p)++) ? (++(*p), (a--)-2) : ((*p)
4     += 3, a); (*p)++) {
5     --a;
6     printf("%d %d OK\n", a, *p);
7     if (*p > 0) {
8         a = (!(-a) || a++) ? 5 : 3;
9         break; } }
10 printf("%d %p %d\n", a, ((char*) b) + a, i);
```

16 -11 OK
160 -8 OK
154 -5 OK
148 -2 OK
142 1 OK
5 0x7fee928991d 1

4. **4 punti** Su foglio protocollo, Scrivere una funzione `main()` e una funzione `half_array()`. La funzione `main()` legge da tastiera un array di n posizioni (con n letto da tastiera) ciascuna di tipo `int`. L'array viene passato a `half_array()` che crea in memoria dinamica un nuovo array di `int` di lunghezza $n \div 2$. Il nuovo array viene riempito sommando aritmeticamente il primo elemento della prima metà dell'array passato, con il primo elemento della seconda metà dell'array passato (la somma è memorizzata in prima posizione del nuovo array); il secondo elemento della prima metà con il secondo elemento della seconda metà, e così via. Il nuovo array viene ritornato a `main()`, che lo stampa su schermo. In caso n sia dispari, non considerare l'ultimo elemento.
5. **3 punti** Su foglio protocollo, descrivere (anche con brevi esempi di codice) le differenze tra un linguaggio fortemente e debolmente tipato, e staticamente e dinamicamente tipato.
6. **3 punti** Su foglio protocollo, scrivere un solo file in cui siano presenti esattamente i) un tentativo di definizione che rimane dichiarazione, ii) un identificatore con *internal linkage*, iii) tre identificatori con *external linkage* (di cui un identificatore di funzione), iv) quattro identificatori con *no linkage*, v) una variabile locale che risieda in memoria permanente.

VEDI FIA A

7. **4 punti** Scrivere cosa stampa il seguente programma, sapendo che le variabili a , b , c in *main* sono memorizzate rispettivamente agli indirizzi $0x7fffeb7d296a$, $0x7fffeb7d297a$ e $0x7fffeb7d297b$.

```

1 static int a= 4;
2 int f(int* x, int y, int* z) {
3     int res;
4     if ((a++,a-=2) > 0) {
5         a ? res= (--y) + (*z+=2), (y+= *x) : 3;
6         printf("%d %d %p %p %d\n", *x, y, z, z+1, res);
7         res+= f(&y, *x, z);
8         return (res+1);
9     }
10    else
11        return res;
12 }
13
14 int main(void) {
15     int a= 4, b= 06, c= 0X3;
16     b= f(&b, a, &c);
17     printf("%d %d %d\n", a, b, c);
18 }

```

6 9 . - 8
 9 14 . 0 12
 14 22 . . 17
 4 40 9
 Istituzione A1 ..
 0x7fffeb7d297b
 0x7fffeb7d297f

8. **3 punti** Data la seguente *struct Node*, definire una funzione ricorsiva di nome *rec_print_sum* che stampi i) il campo *info* in successione dal primo all'ultimo elemento della lista, e, ii) per ogni elemento, la somma dei campi *info* di tutti i successivi elementi nella lista.

```

1 struct Node {
2     int info= 0;
3     struct Node* pNext= NULL;
4 }
5

```

VEDI FMA A

9. **4 punti** Cerchiare le affermazioni vere dato $\text{long long } a[3] = \{1536, -2, \text{LLONG_MIN} + 512\}$; $\text{short int } *p = (\text{short} *) a$; $\text{char } *q = (\text{char} *) a$; $p[1] = 4097$, $p[3] = 4095-2$, $*(q+15) = 73$, $p[9] = 4096*4+1$; sapendo che i tre tipi usati occupano 8, 2, e 1 byte, e $4096 = 2^{12}$ (valori rappresentati in *little endian* e complemento a due). Scrivere la mappa di memoria e giustificare le affermazioni (vere o false). Gli operatori $|$ e $\&$ ritornano rispettivamente l'*or* e l'*and* bit-a-bit dei due operandi, mentre \gg rappresenta l'operatore di *shift* di n posizioni a destra (ricordarsi che l'operazione è effettuata nei registri del processore...). Supporre che i tipi *char* siano tipi con segno: il bit più significativo uguale a 1 rappresenta un valore negativo.

- A. $(q[8] | q[2]) + q[2]$ B. $(q[1] \gg 2) - q[3] > 0$ C. $((q[18] < 1) \& (q[19])) == (q[20] \gg 5)$
 D. $((\text{int})(p+11) - (\text{int})(a+2)) + q[18] \% 7$ E. $((\&p[9] - \&p[2]) + p[8]) \% 2$ F. $*(p+5) - p[4] \% 2$