

Nome e Cognome: _____

Matricola: _____

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

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

vedere FILA A

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

```
1 int i, j, N= 4;
2 for(i=1; i<=N; i++) {
3     for(j=1; j<i; j++)
4         printf("-");
5     printf("%d", i);
6     for(j=1; j <= ((N - i) * 2
7         - 1); j++)
8         printf("-");
9     if(i != N)
10        printf("%d", i);
11        printf("\n"); }
12 for(i=N-1; i>=1; i--) {
13     for(j=1; j<i; j++)
14         printf("%d", i == j / 2? 1
15             : i+j);
16     for(j=1; j <= ((N - i) * 2
17         - 1); j++)
18         printf("-");
19     printf("%d\n", i); }
```

1----1
-2---2
--3-3
---4
--6-3
-4---2
2-----1

3. **3 punti** Scrivere cosa stampa la seguente porzione di codice sapendo che c si trova all'indirizzo 0x7ffee7419e00.

```
1 int a= 0x3d, b = 06, *c= &a;
2 do {
3     for (int i= 2; i ? --i : !!0 ; i--) {
4         a/=2; b = a;
5         printf("%d %d\n", a, i);
6         if (a == b) break; }
7 } while (a--, a && b);
8 printf("%d %p\n", a, ((short*) c) + a);
```

30 1
14 1
6 1
2 1
0 1
-1 0x7ffee7419dfe

4. **3 punti** Su foglio protocollo, scrivere la definizione di una funzione *arrayCreation* che ha come parametro solamente *unsigned int lunghezza*. Questa funzione crea un nuovo array, lo inizializza con valori da 0 a *lunghezza - 1*, e ritorna l'array alla funzione che lo ha chiamato in modo che lo possa utilizzare. Non utilizzare variabili globali o *static*; tutto il codice necessario deve essere contenuto all'interno della funzione.

vedere FILA A

5. **4 punti** Data la seguente struttura, definire una funzione di nome *del_head_if_in* che prende come parametro *int infoToRemove*, e rimuove l'elemento in testa alla lista solamente se il campo *info* di questo elemento ha valore uguale a *infoToRemove*. In caso invece tale valore sia differente, aggiunge in testa un nuovo elemento della lista con campo *info* uguale a *infoToRemove*.

vedere FILA A

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

6. **3 punti** Su foglio protocollo, scrivere una funzione ricorsiva di nome *stringLength* che calcola la lunghezza di una stringa (di lunghezza sconosciuta alla funzione) passata come parametro. Per esempio, la funzione ritorna 12 se viene passata la stringa "Hello, World". **vedere FILA A**

7. **2 punti** Scrivere una espressione con 2 effetti collaterali su una variabile *a* e 1 effetto collaterale su *b*, che *NON* generi un warning *multiple unsequenced modifications*. Scrivere una seconda espressione con 3 effetti collaterali su *a* che invece generi tale warning. Evidenziare in entrambi i casi i *sequence points*.

b = a++, a+=1



a = a++ + ++a

8. **4 punti** Scrivere cosa stampa il seguente programma.

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

5 17 6 16
5 17 5 23
5 17 4 23
4 23 17

9. **4 punti** Cerchiare le affermazioni vere dato $\text{long long } a[3] = \{1537, -67, (LLONG_MAX + 1) + 512\};$
 $\text{int } *p = (\text{int}^*) a; \text{char } *q = (\text{char}^*) a; p[1] = INT_MAX, p[4] += 2048, q[19] = \sim q[19];$ sapendo che i tre tipi usati occupano 8, 4, 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). Gli operatori *|* e *&* ritornano rispettivamente l'*or* e l'*and* bit-a-bit dei due operandi, *~* è la negazione bit a bit.

- A. $((q[17] | q[20]) + q[1]) \% 4$ **B.** $(\sim (p[3] \& p[1])) == p[5]$ **C.** $((\text{int})(p+12) - (\text{int})(a+4)) - q[1] < \sim q[23]$
D. $((\text{long long})(\&p[1])) < ((\text{long long})(\&p[2]))$ **E.** $((\text{long long})(\&p[1])) < ((\text{short int})(\&p[2]))$
F. $((p+3) - p[2]) \% 11$

10. **1 punti** Quali affermazione/i sono vere? Il linguaggio C è ...

- ☐ non tipato; ☐ fortemente e staticamente tipato; ☐ sotto tipato. ☒ staticamente tipato;
☐ retro tipato. ☒ debolmente tipato.