

Prova scritta Programmazione Procedurale

Nome e Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_

1. 1 punti Riportare le conversioni di tipo implicite e scrivere quanto valgono alla fine le variabili  $a$ ,  $b$ ,  $c$ .

```
1 char a= (char) 2, b= (char) 3, c= (char) 4;
2 a= 2 * a * b / c;
3 if ( a < b )
4 b = c;
```

2. 2 punti Riportare le conversioni di tipo implicite e scrivere quanto valgono alla fine le variabili  $a$ ,  $b$ ,  $c$ .

```
1 unsigned int a = 5U;
2 char b = (char) 2;
3 float c = 2.0;
4 c = (float) a / b;
5 b = a / b;
6 a = c - b;
```

3. 4 punti Riportare le conversioni di tipo implicite e scrivere quanto valgono alla fine le variabili  $a$ ,  $b$ ,  $c$ .

```
1 float a = 2.5;
2 long b = 2L;
3 int c = 2U;
4 char d = (char) 2.5;
5 char *p = &d;
6 void *v = p;
7 if ( c < (int) a )
8     b = b*a;
9 else
10     b = a * d / c;
11 a = c * d;
12 c = b / a;
13 d = a + b + c + d;
```

4. 1 punti Si fornisca un esempio di effetto collaterale in C, evidenziando i *sequence points* e spiegando perché genera o non genera un warning *multiple unsequenced modifications*, fornendo un controesempio.

5. 2 punti Si definisca una funzione *centro* che ha come parametro un array di *int* e ritorna la media tra il valore massimo e il valore minimo tra gli  $n$  elementi presenti nell'array.

6. 2 punti Scrivere una funzione che prenda in input una stringa di lunghezza massima 30 e la stampi invertita.

7. 1 punti Scrivere una funzione di inserimento in coda ad una lista in maniera iterativa.

8. 2 punti Scrivere una funzione di inserimento in coda ad una lista in maniera ricorsiva.

9. 4 punti Data la seguente struttura, definire una funzione di nome *positivi* che rimuove gli elementi il cui campo *val* ha valore minore di 0.

```
1 struct Lista {  
2     int val;  
3     struct Lista* pNext;  
4 }
```

10. 1 punti Indicare quali tra i seguenti sono l-value, dati *int s[3]; int \*x = s, \*y = x + 3*.

☐  $x + 3$ ;   ☐  $*(y + 3) = 3$ ;   ☐  $*x = y$ ;   ☐  $\&y$ ;   ☐  $a[3] - 3$ ;   ☐  $y = \&x$ .

11. 1 punti Si scriva un programma che permetta di scambiare il valori di due variabili x e y integrando l'utilizzo di puntatori.

12. 2 punti Si scriva una funzione *matrmalloc*, in grado di allocare dinamicamente una matrice rettangolare di float, le cui dimensioni sono ricevute come parametri. Inizializzare la matrice azzerando tutte le celle.

13. 4 punti Si scriva una funzione *matrptr* che riceva come parametri dimensioni e puntatore a una matrice rettangolare di float. Inizializzare la matrice azzerando tutte le celle. Spiegare le differenze con l'esercizio precedente evindenziando pro e contro.

14. 4 punti Scrivere cosa stampa la seguente porzione di codice sapendo che *y* si trova all'indirizzo 0xf0ff010.

```
1  int x= 0xae, a = 12, *y= &x;
2  for (int i= 0; i<x ; i++) {
3      x = x/2 - (--a);
4      printf("%d %d\n", x, --a);
5      if (2*a>x) break;
6  }
7  printf("%d %p\n", x, ((long*) y) + a);
```

15. 4 punti Dato  $int\ s[3] = \{511, -666, INT\_MAX\};$   
 $int\ *x = (int*)\ s;$   $char\ *y = (char*)\ s;$   $x[1] = INT\_MIN - 3;$   $y[2] = \sim y[2] - 64;$  sapendo che i due tipi occupano 4 e 1 byte, con valori rappresentati in *little endian* e complemento a due, scrivere la mappa di memoria. L'operatore  $\sim$  è la negazione bit a bit.