

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

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

1. **3 punti** Elencare le conversioni di tipo (... da ... a). Dato  $USHRT\_MAX = 65535$ , scrivere il valore finale della variabile  $b$  sapendo che il carattere  $a$  ha valore 97 e le altre lettere seguono in ordine alfabetico.

```
1 long int fun2(int p) {
2     return (p + 'l' - 'c'); }
3
4 int fun1(long p) {
5     char a= 'o';
6     return fun2(p - a + 'b'); }
7
8 int main(void) {
9     unsigned short a = -1LL;
10    double b= fun1(a);
11 }
```

**b vale 65532.0**

**per le conversioni vedere FILA A**

2. **3 punti** Scrivere cosa stampa la seguente porzione di codice.

```
1 int i, s, c= 0, c1= 0, a= 4, k=0;
2 for(i=1; i<a; ++i, k=0) {
3     for(s=1; s<a-i; ++s) {
4         printf("--");
5         ++c; }
6
7 while(k != 2*i+1) {
8     if (c < a-1) {
9         printf("*--");
10        c+=1; continue;}
11    else {
12        ++c1;
13        printf("%d--", (i+k-2*c1));}
14        ++k; }
15    c1 = c = k = 0;
16    printf("\n"); }
```

```
--1--2--3-
--*-0--1--2--3--4-
***-1-0--1--2--3--4--5-
```

3. **3 punti** Scrivere cosa stampa la seguente porzione di codice.

```
1 int a= 0xfd - 010;
2 while(a > 9 ? !!!0: !(a+1, a-=2, --a))
3     {
4         printf("%d \n", a);
5         if (a + 2> 0x10) {
6             a= -3;
7             continue; }
8         a+= 4;
9     }
10 a++ || a++;
11 printf("a: %d\n", a);
```

```
245
-6
-5
-4
-3
-2
-1
a: 2
```

4. **4 punti** Su foglio protocollo, scrivere la definizione di una funzione *crea\_m* che prende come parametri *righe* e *colonne* (*int*), e crea una matrice *righe*×*colonne* riempiendo i valori da tastiera (*scanf*). Infine, la funzione ritorna la matrice. *Suggerimento: creare la matrice come un semplice array di righe*×*colonne* di tipo *int*, ed utilizzare l'aritmetica dei puntatori per indicizzare l'array come una matrice.

**Vedere correzione esercizio 4 FILA A**

5. **3 punti** Data la seguente *struct* definire una funzione di nome *print\_pari* che prende come parametro una lista e stampa su video il valore del campo *info* per tutti gli elementi in posizione pari della lista: supposto il primo elemento in posizione 1, stampa invece quelli in posizione 2, 4, 6, etc (se presenti...).

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

**Vedere correzione esercizio 5 FILA A**

6. **3 punti** Evidenziare con una freccia ciascun sequence point nel codice sottostante. Quanti effetti collaterali ci sono su *a* (supponendo *a* = 1) in totale? Scrivere una espressione con 3 effetti collaterali su una variabile *a* e 1 effetto collaterale su *b*, che NON generi un warning *multiple unsequenced modifications*, e una espressione che lo generi. Evidenziare i sequence point in entrambe le espressioni.

```
1 while (a++) {
2     a > 0? a+=1 : a-= 1;
3     a++ || a++;
4     a, a--, a+= 1;
5 }
```

**Vedere correzione esercizio 6 FILA A**

7. **4 punti** Dire quali compilazioni provocano errore a causa del linker (e perché): 1) *gcc -c write.c*, 2) *gcc -o main main.c*, 3) *gcc -o write write.c*, 4) *gcc -c main.c*, 5) *gcc write.c main.c -o main*. In caso il punto 5) ritorni un errore, descrivere come può essere corretto. Dopo aver corretto l'errore, che tipo di *linkage* hanno *count*, *i*, e *mywrite*, ed in quale file sono definite? Cosa stampa il programma?

main.c	write.c
<pre>int count= 4; void mywrite(int count);  int main(void) {     int count= 4;     do {         mywrite(count);     }while(count &gt;=0); }</pre>	<pre>#include &lt;stdio.h&gt; extern int count;  void mywrite(int a) {     static int i= 1;     printf("%d\n", count=         count - (a + i)); }</pre>

**Vedere correzione esercizio 7 FILA A**

8. **5 punti** Cerchiare le affermazioni vere dato  $\text{int } a[5] = \{13+2*32, 129, [2] = \text{INT\_MIN} + 47, 130939, 131072*2 + 65\}$ ;  $\text{short int } *p = (\text{short}*) a$ ;  $\text{char } *q = (\text{char}*) a$ ;  $q[2] = -1$ ;  $p[3] = 128*2$ ; sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e  $131072 = 2^{17}$  (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, mentre *>>* (*<<*) rappresenta l'operatore di *shift* di *n* posizioni a destra (sinistra), inserendo 0 nelle posizioni eliminate (operazione fatta nel processore).

A.  $(*(p+8) - *(q+2) - q[6]) > 65$     B.  $(\&a[4] - (a+1)) + *(q+18) - 7$     C.  $(q[13] - q[14] + *(p+7))\%2$   
D.  $(*(q+2) | q[8]) + (q[7] \& q[13])$     E.  $((\text{int})(a+7) - (\text{int})(p+3)) + q[16]\%87$     F.  $((q[8] >> 3) + \sim q[4])\%130$

9. **2 punti** Su foglio protocollo, descrivere le differenze tra *dichiarazione* e *definizione* di variabile, con esempi.