

Nome e Cognome: _____

Matricola: _____

1. **3 punti** Elencare tutte le conversioni di tipo (esempio: *da int a double*). Riportare i valori finali di *i*, *x*, *mod*.

```
1 int i = 7ul;
2 float x = 0x11;
3 i = x;
4 i += 2.5;
5 int mod = i % (int) x;
```

19 7 DA UNSIGNED LONG A INT
17.0 0x11 DA INT A FLOAT
2 X DA FLOAT A INT
i DA INT A DOUBLE, i+2.5 DA DOUBLE A INT
X DA FLOAT A INT

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

```
1 int a= 1, i= -3;
2
3 for (; i <= 3;) {
4     printf("OK\n");
5     ++i;
6     while (!(a+= i));
7 }
8
9 printf("%d\n", a + i);
```

OK
OK
OK
OK
OK
OK
OK
12

3. **4 punti** Scrivere cosa stampa il seguente programma.

```
1 int i, space, star=0, rows= 5;
2 for(i = 1; i < rows; i++) {
3     for(space = 1; space <= rows-i;
4         space++) {
5         }
6     while(star != (2*i - 1)) {
7         if(star==0 || star==2*i-2)
8             printf("*");
9         else
10            printf(" ");
11        star++;
12    }
13    star=0;
14    printf("\n");
15 }
```

```

-----*
--*--*
-*--*--*
-*--*--*

```

4. **3 punti** Per ogni identificatore di variabile e funzione scrivere se definizione o dichiarazione.

```
1 #define PI 3.14159
2 static double a= 4.2;
3 extern float area(double b);
4 typedef int c;
5 c d= 1;
6
7 extern int myFunc(float e, int* f) {
8     static float g= 4.1;
9     int h=5, i[*f];
10    extern c d;
11    // Comandi...
12 }
```

MyFunc,

a, d, e, f, g, h, i DEFINITI
area, b, d (DENTRO MYFUNC) DICHIARATI

5. **3 punti** Cosa stampa il seguente programma? Gli operatori relazionali e virgola sono associativi da sinistra verso destra, a si trova all'indirizzo $0x7fff5280fab$, un *int* occupa 4 byte e un *long long* occupa 8 byte.

```
1 int a = 0x11, *b = &a;
2 int c = (a++, a += 3) < (++*b) >= (!0)
3 ;
4 printf( "%d %d\n", a, c );
5 printf("%p %p %lu\n", b, (long long*)
   b + 2, sizeof(*b));
```

22 1
 $0x7fff5280fab$ $0x7fff5280face$ 4

6. **4 punti** Scrivere cosa stampa il seguente programma.

```
1 int f1(int* b, int c, int* d) {
2     static int a = 3;
3     return (++a * (*b)++ * c++ * (*d)
4         ++);
5     printf("%d", a);
6 }
7 int f2(int* b, int c) {
8     int a = 1;
9     return (++a + (*b)--- - c--- );
10 }
11
12 int main(void) {
13     int a = 2, b = 3, c = 1;
14     b = f1(&a, b, &c);
15     printf("%d %d %d\n", a, b, c);
16     b = f1(&a, b, &c);
17     printf("%d %d %d\n", a, b, c);
18     c = f2(&b, c+b);
19     printf("%d %d %d\n", a, b, c);
20 }
```

3 24 2
 4 720 3
 4 719 -1

7. **3 punti** Descrivere le funzionalità dei quattro principali moduli in cui è suddiviso *gcc*, riportando come ciascuno di essi trasforma il codice passato come input. (su foglio protocollo)
8. **3 punti** Scrivere in linguaggio C una funzione ricorsiva che moltiplica i primi n valori interi; la funzione prende in input e restituisce in output un valore di tipo *int*. Se l'input è $n = 5$, la funzione restituisce 120, cioè $5 \times 4 \times 3 \times 2 \times 1$. (su foglio protocollo)
9. **4 punti** Cerchiare le affermazioni vere dato $\text{long long } a[3] = \{3, 65535, 65536\}$; $\text{int } *p = (\text{int}*) a$; $\text{short int } *q = (\text{short}*) a$; sapendo che i tre tipi usati occupano 8, 4, e 2 byte, e $65536 = 2^{16}$ (valori rappresentati in *little endian*). A. $q + 2 \neq p + 1$; B. $*(p + 1) == *(q + 2)$; C. $*(q + 4) == *(p + 2)$; D. $*(q + 8) - q[0] \leq 0$; E. $*(q + 9) - *(q + 8) == 2$; F. $((\text{int})(q + 3) - (\text{int})(p + 3))/2 == -3$.

E e' FALSA! Sono vere solo B D F