

Nome e Cognome: _____

Matricola _____

1. 4 punti Scrivere il valore finale delle variabili *mean*, *mean2*, e *mod*, ed il loro tipo.

```

1 int sum = 10, count = 2;
2 double mean = sum / count;
3
4 double mean2 = mean / sum;
5 int mod = sum % count;
```

2. 5 punti Scrivere cosa stampa il seguente programma.

```

1 int i=0;
2 int j=0;
3 int k=0;
4
5 for(i=1; i<=7; i++){
6     for(j=1; j<=i; ++j)
7         printf("%d",j);
8
9     for(k=7-i; k>=1; k--)
10        printf("*");
11
12    printf("\n");
13 }
```

3. 2 punti Scrivere cosa stampa il seguente programma.

```

1 int a = 20;
2 int b = 10;
3 int c = 15;
4 int d = 5;
5 int e;
6
7 e = (a + b) * c / d;
8 printf("Value of (a + b) * c / d is :
9      %d\n", e );
10
11 e = ((a + b) * c) / d;
12 printf("Value of ((a + b) * c) / d is
13      : %d\n" , e );
14
15 e = (a + b) * (c / d);
16 printf("Value of (a + b) * (c / d) is
17      : %d\n", e );
18
19 return 0;
```

4. 4 punti Scrivere cosa stampa il seguente programma.

```
1
2 int main(void) {
3     int n=5, i, p, s, k=3;
4     s = 0;
5     p = 1;
6     for (i=1; i<=n; i++) {
7         p = p*k;
8         s = s+p;
9     }
10    printf("%d\n", s);
11    return(0);
12 }
```

5. 4 punti Scrivere cosa stampa il seguente programma.

```
1
2 int fibonacci(int i) {
3
4     if(i == 0) {
5         return 0;
6     }
7     if(i == 1) {
8         return 1;
9     }
10    return fibonacci(i-1) + fibonacci(i-2);
11 }
12 int main() {
13     int i;
14     for (i = 0; i < 10; i++) {
15         printf("%d\t\n", fibonacci(i));
16     }
17     return 0;
18 }
```

6. 2 punti Scrivere cosa stampa il seguente programma e indicare dove g è dichiarata come variabile locale e dove globale.

```
1 int g = 20;
2 int main () {
3     int g = 10;
4     printf ("value of g = %d\n", g);
5
6     return 0;
7 }
```

7. 4 punti Scrivere cosa stampa il seguente programma.

```
1 int a = 20;
2
3 int main () {
4     int a = 15;
5     int b = 14;
6     int c = 0;
7     printf ("il valore di a in main() =
8         %d\n", a);
9     c = sum( a, b);
10    printf ("il valore di c in main() =
11        %d\n", c);
12    return 0;
13 }
14 int sum(int a, int b) {
15     printf ("il valore di a in sum() =
16         %d\n", a);
17     printf ("il valore di b in sum() =
18         %d\n", b);
19     return a + b;
20 }
```

8. 3 punti Scrivere cosa stampa il seguente programma.

```
1 int main () {
2     int a, b;
3     a = b = 4;
4     b = a++;
5     printf ("%d %d %d %d", a++, --b, ++
6         a, b--);
7 }
```

9. 3 punti Scrivere cosa stampa il seguente programma.

```
1 int main () {
2     int a[4] = { 25, 16 };
3     printf ("%d %d", a[0] & a[1], a[1]
4         | a[2]);
5 }
```

10. 4 punti Scrivere cosa stampa il seguente programma.

```
1 int main () {
2     static int num = 8;
3     printf ("%d", num = num - 2);
4     if (num != 0)
5         main ();
6 }
```

11. 4 punti Scrivere cosa stampa il seguente programma.

```
1 int main() {
2     int m = -10, n = 20;
3     n = (m < 0) ? 0 : 1;
4     printf("%d %d", m, n);
5 }
```

True or False

12. Only character or integer can be used in switch statement. True — False
13. The return type of *malloc* function is void. True — False
14. *define* is known as *preprocessor* compiler directive. True — False
15. *sizeof()* is a function that returns the size of a variable. True — False
16. *continue* keyword skip one iteration of loop. True — False
17. A do-while loop is used to ensure that the statements within the loop are executed at least twice. True — False