Prova scritta Programmazione I - 15 Gennaio.

Nome e Cognome: _____

Matricola: _____

1. [3 punti] Scrivere il valore finale delle variabili *b*, *d* e *c* (il valore di 'c' come intero è 99).

```
1 int   a = 3, b= 0;
2 double c= 0, d= 153;
3 char e = 'c';
4
5 b = a + e;
6 d+= d / b;
7 c= (int) d / b;
```

b = 102
d = 154.5
c = 1.0

2. [3 punti] Scrivere cosa stampa il seguente programma.

```
1 int i= -2;
2
3 while(i++ + 1)
4   printf("OK\n");
5
6 int b= -2, c= 0xb -1;
7 printf("%d\n", i++-(b*=3));
8 printf("%d\n", ++i+c);
```

OK
6
12

3. [3 punti] Scrivere cosa stampa il seguente programma.

```
1 for(int i=1; i<=5; i++) {
2
3   for(int j=1; j<=5; j++)
4
5     if(i!=1 && i!=5 && j!=1 && j!=5)
6       printf(" ");  // uno spazio
7     else
8       printf("*");
9
10  printf("\n");
11 }
```

4. [3 punti] Quali tra *PI, circ, radius1, area, radius2, func, a, b, ..., f* sono definizioni di variabili o funzioni?

```
1 #define PI 3.14159
2 int a= 1;
3 extern int b;
4 extern double c= 1.0;
5 extern int circ(double radius1);
6 static double area(double radius2);
7
8 int* my_func(int d) {
9   static float e= 4.0;
10  double* f;
11  extern int g;
12  // Comandi...
13 }
```

a
c
MY_FUNC
d
e
f

5. [3 punti] Per tutti gli identificatori scrivere se hanno *external linkage*, *internal linkage* oppure *no linkage*.

```
1  static int a= 3;
2  extern int b;
3  float* c;
4  int my_func1(int);
5
6  static void my_func2(int d) {
7      int e;
8      extern int a;
9      extern int f;
10     static int g= 5;
11 }
```

a INTERNAL
b EXTERNAL
MY_FUNC 1 EXTERNAL
MY_FUNC 2 INTERNAL
d NO LINKAGE
e NO LINKAGE
c EXTERNAL
f EXTERNAL
g NO LINKAGE

6. [3 punti] Cosa stampa il seguente programma? Gli operatori unari hanno precedenza sugli operatori binari, l'operatore && ha la minima precedenza tra tutti (in questo esempio), la moltiplicazione ha precedenza sulla sottrazione, *a* si trova all'indirizzo *0x7fff51fecaf8*, un *int* occupa 4 byte e *short int* occupa 2 byte.

```
1 int a = 5, *b= &a;
2 int c = !!a && (++**&b)**b-36;
3 printf( "%d %d\n", a, c );
4
5 printf("%p %p %d\n", b, (short*) b +
      1, (*b)+2);
```

6 0
0x7fff51fecaf8 0x7fff51fecafa 8

7. [4 punti] Scrivere cosa stampa il seguente programma.

```
1  static int a= 5;
2
3  int f1(void) {
4      extern int a;
5      printf("%d ", a);
6      return (a--);
7  }
8
9  int f2(int* b, int* c) {
10     printf("%d ", (*b)++);
11     return (a * *b * (*c)++);
12 }
13
14 int main(void) {
15     int a= 8, b= 0, c= 0;
16     b= f1();
17     c= f2(&a, &b);
18     printf("%d %d %d\n", a, b, c);
19 }
```

5 8 9 6 180

8. [3 punti] Cerchiare le affermazioni vere dato *int a= 3, *p= &a; *p= a + 1;* A. *p* è un *lvalue*; B. **p* è un *rvalue*; C. 3 è un letterale; D. *p* è un identificatore; E. *&p* è un *rvalue*; F. Il valore finale di *a* è 4.

9. [3 punti] Dato *int a=-1, *const p= &a; const double b= 3.0, *q= &b; float c= ((a+= a) || a) ? *q*2: *q-1;*
A. Alla fine *a* vale −2; B. L'inizializzazione di *b* contiene una conversione di tipo; C. $*p = 4$ è permesso; D. $*q = 4$ è permesso; E. L'assegnamento dell'espressione condizionale a *c* porta ad una conversione di tipo; F. Alla fine *c* vale 6.0; G. $p = (int^*)q$ è permesso; H. La precisione di *b* è maggiore di quella di *c*.

10. [4 punti] Cerchiare le affermazioni vere dato *short int a[4]= {1,1,[3]=1}; int *p = (int*) a;* sapendo che un *int* occupa 4byte, uno *short int* occupa 2byte, e $65536 = 2^{16}$ (valori interi rappresentati in *little endian*).
A. $*a$ vale 0; B. $*p$ vale 65536; C. $*(a+2)$ vale 0; D. $*(p+1)$ vale 65536; E. $\&a[3] > (p+1)$ è vera.
F. $((int)(a+3)) - (int)(\&a[0])$ vale 3.