Prova scritta Programmazione I - 2 Febbraio.

Nome e Cognome: _____ Matricola: _____

1. [3 punti] Elencare tutte le conversioni di tipo presenti, riportando i tipi coinvolti (*da ... a*).

```
1 int i = 7l;
2
3 float x = 0.5;
4 i = x;
5 x += 2.5;
```

71 DA LONG INT A INT
0.5 DA DOUBLE A FLOAT
X DA FLOAT A INT
X DA FLOAT A DOUBLE, X.+2.5 DA DOUBLE A FLOAT

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

```
1 int i= 5%3, a= 2;
2
3 do
4   printf("OK\n");
5 while((a= (--i + 1)));
6
7 int c= 0x10 -1;
8 printf("%d\n", i+=-(a+=3));
9 printf("%d\n", i+c--);
```

OK
ok
OK
-4
11

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

```
1 for(int i=1;i<=5;i++) {
2   for(int j=5;j>=1;j--)
3     if(i==j)
4       printf("*");
5     else {
6       printf(" ");
7       continue;
8     }
9
10  for(int j=1;j<=5;j++)
11    if(i==j)
12      printf("*");
13    else
14      printf(" ");
15
16  printf("\n");
17 }
```

\- - - - * - - - -
- - - * - - * - - -
- - * - - - - * - -
- * - - - - - - * -
* - - - - - - - - *

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

```
1 static double a= 3.5;
2 extern float area(double);
3 typedef int pippo;
4 pippo b= 1;
5
6 extern int* my_func(int c, int* d) {
7   static float e= 4.0;
8   int f=5, g[f];
9   extern pippo b;
10  // Comandi...
11 }
```

A DEFINIZIONE
AREA DICHIARAZIONE
B DEFINIZIONE
MY_FUND DEFINIZIONE
C DEFINIZIONE
D DEFINIZIONE
E DEFINIZIONE
F DEFINIZIONE
G DEFINIZIONE
B (DENTRO MY_FUNC) DICHIARAZIONE

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

```
1 #define NUM 10
2 static  double a[NUM];
3 extern int b;
4 float* c;
5 int my_func1(int d);
6
7 void my_func2(int e) {
8   int f;
9   extern int g;
10  static int h;
11 }
12
13 int g = 3;
```

A INTERNAL
B EXTERNAL
C EXTERNAL
MY_FUNC1 EXTERNAL
D NO LINKAGE
MY_FUNC2 EXTERNAL
E NO LINKAGE
F NO LINKAGE
G EXTERNAL
H NO LINKAGE

6. [3 punti] Cosa stampa il seguente programma? Gli operatori relazionali e virgola sono associativi da sinistra verso destra, *a* si trova all'indirizzo *0x7fff58c35afc*, un *int* occupa 4 byte e un *long long* occupa 8 byte.

```
1 int a = 0x12, *b= &a;
2 int c = (a++,a-=3) < (++*&*&*b) >=
      (!0);
3 printf( "%d %d\n", a, c );
4
5 printf("%p %p %lu\n", b, (long long*)
      b + 1, sizeof(*b));
```

17 1
0x7fff58c35afc 0x7fff58c35b04 4

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

```
1 #include <stdio.h>
2 int f(int* b, int c, int* d) {
3   static int a= 3;
4   return (++a * (*b)++  * c++ * (*d)
      ++);
5   printf("%d", a);
6 }
7
8 int main(void) {
9   int a= 2, b= 3, c= 1;
10  b= f(&a, b, &c);
11  printf("%d %d %d\n", a, b, c);
12  b= f(&a, b, &c);
13  printf("%d %d %d\n", a, b, c);
14 }
```

3 24 2
4 720 3

8. [3 punti] Cerchiare le affermazioni vere dato *int a= 2, *p= &a; *p= a + 1;* (A.) $a+1$ è un *rvalue*; (B.) p è un *lvalue*; C. il comando $a = a{+}{+}$; non genera errore; D. il comando $(a{+}{+}, a{+}{=}a)$; genera errore; E. 2 è un identificatore; F. p ha tipo *int*.

9. [3 punti] Cerchiare le affermazioni vere dato *int a=-1, *const p= &a; const double b= 3.0, *q= &b; float c= ((a-= !!a, a + 1) >= 0) ? *q*2: 3.0f;* (A.) Alla fine a vale -2; B. L'inizializzazione di b contiene una conversione di tipo; C. $p = q$ è permesso; (D.) $q = NULL$ è permesso; E. $a + 1$ è un comando; F. $\&(\&a)$ è permesso;

10. [3 punti] Cerchiare le affermazioni vere dato *long long a[3]= {3,65535.65536}; int *p = (int*) a; short int *q = (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 != p; (B.) $^*p == {}^*q$; C. $^*(q+4)$ è maggiore di 0; (D.) $^*(q+9) - q[0]$ vale -2; E. $(int)(\&a[1]) - (int)(p+1)$ vale 2. (F.) $^*(p+4) > {}^*(p+2)$.