Prova di esame.

Nome e Cognome: _____

Matricola: _____

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

```
int sum = 5, count = 2;
double mean= sum / count;

double mean2= mean / sum;

int mod= sum % count;
```

MEAN VALE 2.0 TIPO DOUBLE
MEAN 2 VALE 0.4 TIPO DOUBLE
MOD VALE 1 TIPO INT

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

```
int i= 1;

do {
  puts("OK");
}while(i--);

int a= 0, b= 5-2, c= 0xA;
printf("%d\n", a++-(b*=3));
printf("%d\n", ++a+c);
```

OK
ok
-9
12

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

```
int i= 0;
int j= 0;
int k= 0;

for(i=1;i<=7;i++){
  for(j=1; j<=i; ++j)
    printf("%d", j);

  for(k=7-i; k>=1; k--)
    printf("*");

  printf("\n");
}
```

1 * * * * * *
1 2 * * * * *
1 2 3 * * * *
1 2 3 4 * * *
1 2 3 4 5 * *
1 2 3 4 5 6 *
1 2 3 4 5 6 7

**3 punti** Elencare tra *abs, value, circ, radius, func, a, b, ..., f* quali sono definiti.

```
int abs(int value);
static double circ(double radius);
extern float a;
int b;
extern double c= 1;

int func(int d) {
  int e;
  static float f= 4;
  extern int g;
  // Comandi...
}
```

C
FUNC
D
E
F

È UN TENTATIVO DI DEFINIZIONE, IN QUESTO CASO È UNA DEFINIZIONE VERA E PROPRIA (PAG 158 LIBRO)

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

```
1 int func1(int a);
2 double b[];
3 extern int c;
4 static float* d;
5
6 static void func2(int e) {
7   int f;
8   extern int g;
9   static int h= 3;
10 }
```

FUNC 1 EXT
A NO LINKAGE
B EXT
C EXT
D INT

FUNC 2 INT
E NO LINKAGE
F NO LINKAGE
G EXT
H NO LINKAGE

6. [3 punti] Scrivere cosa stampa il seguente programma, sapendo che gli operatori unari hanno precedenza sugli operatori binari, che $a$ si trova in memoria all'indirizzo *0x7fff5ab1cb2c*, e sapendo che una variabile di tipo *int* occupa 4 byte in memoria.

```
1 int a = 7;
2 int b = !++*&a+1;
3 printf( "%d %d\n", a, b );
4
5 int* p= &a;
6 printf("%p %p %d\n", p, p+1, (*p)+2);
```

8 1
0x7fff5ab1cb2c 0x7fff5ab1cb30 10

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

```
1 #include <stdio.h>
2
3 static int a= 5;
4
5 int f1(void) {
6   a = 3;
7   printf("%d ", a);
8   return (a + 5);
9 }
10
11 int f2(int x, int* a) {
12   printf("%d ", *a);
13   return (x * (*a)++);
14 }
15
16 int main(void) {
17   int a= 0, b= 0, c= 0;
18   a= 7;
19   b= f1();
20   c= f2(a, &b);
21   printf("%d %d %d\n", a, b, c);
22 }
```

3 8 7 9 56

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

9. [3 punti] Come sopra con *int a=3, *const p=&a; float b= 3.0; float c= (a= !a) ? b*2: 0.0;* A. Alla fine $a$ vale 1; (B) L'inizializzazione di $b$ contiene una conversione di tipo; (C) $*p = 4$ è permesso; (D) L'espressione condizionale contiene una conversione di tipo; E. Alla fine $c$ vale 6.0; F. $p = p + 1$ è permesso.

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