

Prova scritta Programmazione I - 13 Gennaio.

Nome e Cognome: _____

Matricola: _____

1. **3 punti** Elencare tutte le conversioni di tipo presenti; dire il valore finale di c.

```
1 double circ(float a) {
2     return (a*a);
3 }
4
5 int main(void) {
6     unsigned int a = -1LL;
7     float b = circ(2.0);
8     int c = 3.9;
9     unsigned short d = 1000;
10    d = -4;
11 }
```

~~ALLA DUE DUE DA DUE DA DUE DA DUE~~
 -1LL DA LONG LONG A UNSIGNED INT
 2.0 DA DOUBLE A FLOAT (PARAMETRO FLOAT)
 a*a DA FLOAT A DOUBLE
 RITORNO CIRC DA DOUBLE A FLOAT
 3.9 DA DOUBLE A INT
 1000 DA INT A UNSIGNED SHORT
 -4 DA INT A UNSIGNED SHORT

C=3

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

```
1 int a = 3;
2 while ((a+=1)? a-- : a) {
3     a--;
4     printf("OK\n");
5 }
6 printf("%d\n", a);
7 int b = 0x1a - 3;
8 b += a || a;
9 printf("%d", b);
```

OK
 OK
 OK
 OK
 0
 23

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

```
1 int n = 5, i = 8, count = 0, c = 0;
2 for (count = 2; count <= n+1;) {
3     for (c = 2; c <= i - 1; c++) {
4         if (i%c == 0)
5             break;
6         if (c == i) {
7             printf("%d\n", i);
8             count++;
9         }
10        i++;
11    }
12 }
```

11
 13
 17
 19
 23

4. **3 punti** Scrivere i nomi di tutti gli identificatori di variabile e funzione definiti.

```
1 extern int a = 3, b;
2 static double c;
3 typedef int d;
4 d e, f;
5 int f = 3;
6
7 int fun2(int g, int* h) {
8     int i;
9     int l = 5, m[e];
10    extern d e;
11    //Comandi...
12 }
```

A DERIVATA
 c
 f
 e
 FUNZ
 g

h
 i
 l
 m

5. **4 punti** Dire quali compilazioni provocano errore a causa del linker (e perché): 1) `gcc -o main main.c`, 2) `gcc -c write.c`, 3) `gcc -o write write.c`, 4) `gcc -o main write.c main.c`. In caso il punto 4) ritorni un errore, descrivere come può essere corretto. Dopo aver corretto l'errore, che tipo di *linkage* hanno `count`, `i`, e `write`, ed in quale file sono definite? Cosa stampa il programma?

```
main.c
static int count = 3;
void write();

int main(void) {
    extern int count;
    do {
        write();
    } while(count >= 0);
}

write.c
#include <stdio.h>
static int i = 1;
extern int count;

void write() {
    printf("%d\n", count = count - i);
}
```

1) ~~main~~ ~~WRITE~~ NON DEFINITA
 3) ~~PERCHÉ~~ COUNT E MAIN NON DEFINITA
 4) COUNT ESSENDO STATIC È PRIVATE IN MAIN.C
 LINKAGE INTERNO, COUNT E WRITE ESTERNO
 COUNT DEFINITA IN MAIN.C
 WRITE DEFINITA IN WRITE.C

2
1
0
-1

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

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

2 14 5
 2 15 30

7. **3 punti** Cerchiare se vero dato `int a = 2, *p = &a;` A. `*&a` è un *rvalue*; ☒ B. `**&p` è un *lvalue*; ☒ C. `a = 2, a++`; non genera warning; ☒ D. `a = 2, a++`; contiene 2 effetti collaterali; E. `a++ || a++`; genera warning; F. `a++ ? a++ : a++`; genera warning; G. `p` ha tipo `int`; H. `!*p == -2`; ☒ I. `&a == p`.
8. **2 punti** Dato `float a = 1.2, *const q = &a; const double b = 2.0, *p = &b;`
☒ A. L'inizializzazione di `a` contiene una conversione di tipo; B. `*p = 4.1` è permesso; ☒ C. `*q = 4.2` è permesso; D. `q = (float*)p` è permesso; E. La precisione di `a` è maggiore di quella di `b`; F. `++a == 2`.
9. **2 punti** Ordinare da 1 a 4 i moduli di cui si compone `gcc`, in base a come vengono richiamati:
☒ Linker; ☒ Compiler; ☒ Preprocessor; ☒ Assembler.
10. **4 punti** Cerchiare le affermazioni vere dato `long long a[3] = {9, 131072, 131071}; int *p = (int*) a; short int *q = (short*) a;` sapendo che i tre tipi usati occupano 8, 4, e 2 byte, e $131072 = 2^{17}$ (valori rappresentati in *little endian*). ☒ A. `q+3 > &p[1]` ☒ B. `*(p+1) == *(q+9)`; ☒ C. `*(q+5) - *(q+9) == 1`; D. `*(q+8) > 0`; ☒ E. `p[4] == 131071`. F. `((int)(&a[1]) - (int)(q+1)) < 6`.