Prova scritta Programmazione I - 19 Febbraio 2020 - FILA A

Nome e Cognome: _____ Matricola: _____

1. [3 punti] Elencare tutte le conversioni di tipo. Qual è il valore di b alla fine della funzione?

```
1 #define A 3.6
2 short int f(long long p1, double p2){
3   return (p1 >= p2 ? p1 : p2);
4 }
5
6 int main(void) {
7   int a = A;
8   int b = 2UL;
9   b %= f(b, a);
10 }
```

linea 7: 3.6 convertito da double a int
linea 8: 2UL convertito da unsigned long a int
linea 9: b convertito da int a long long, a da int a double
linea 3: p1 convertito da long long in double (p1>=p2)
linea 3: p1 convertito da long long a double (l'espressione condizionale ritorna double, il più alto dei due in ranking)
line 3: il risultato da double a short (la funzione ritorna short)
linea 9: risultato funzione convertito da short a int

alla fine b vale 2 (b = 2 % 3)

2. [3 punti] Scrivere cosa stampa la seguente porzione di codice.

```
1 int a= 0xf9;
2 while (a > 017 ? !!1: !!(a--, --a)) {
3   printf("%d \n", a);
4   if (a + 2 >= 0x10) {
5     a= -0x2;
6     continue; }
7   a+= 3; }
8 !(a+1) && a++;
9 printf("a: %d\n", a);
```

249
-4
-3
-2
-1
a: 0

3. [3 punti] Cerchiare se vero dato *int a= 2, *p= &a;* e motivare la risposta su foglio protocollo. A. $++(a++)$ non genera errore; B. $a+=2, a++;$ contiene 2 effetti collaterali; C. $*\&a$ è un *rvalue*; D. $**\&p$ è un *lvalue*; E. $a++ \&\& a++;$ genera warning; F. $*a$ genera errore; G. p ha tipo *int*; H. $!*p == -2;$ I. $\&a == p$.

4. [2 punti] Cerchiare le affermazioni vere dato **float** *a=1.2, *const q= &a; const double b= 2.0, *p= &b;* (motivare la risposta su foglio protocollo) 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 b è minore di quella di a; F. $++a == 2$.

5. [4 punti] Su foglio protocollo, scrivere la definizione di una funzione *arrayDiff* che prende due array di *int* in input, controlla che abbiano la stessa lunghezza, e ritorna la somma delle differenze tra elementi in posizioni corrispondenti: $a[0] - b[0] + a[1] - b[1] + \ldots$.

6. [3 punti] Data la seguente *struct*, su foglio protocollo definire una funzione di nome *parity* che prende in input una lista di tali elementi e ritorna 1 se il numero di elementi della lista è dispari, 0 altrimenti.

```
1 struct Node {
2   int info= 0;
3   struct Node* pNext= NULL;
4 }
```

7. [4 punti] Su foglio protocollo, scrivere il contenuto di due file, *main.c* e *secondary.c*. I due file devono essere compilati insieme producendo con successo l'eseguibile di nome *myexe*: scrivere il comando per fare ciò. I due file devono poter essere compilati anche indipendentemente per produrre un file oggetto; scrivere il comando per fare ciò. In *main.c* ci devono essere esattamente un tentativo di definizione che rimane dichiarazione, una definizione di variabile e una di funzione entrambi con *external linkage*, una dichiarazione di funzione con *external linkage*. In *secondary.c* ci devono essere esattamente una definizione di funzione con *external linkage*, una variabile con *internal linkage* e tre variabili con *no linkage*.

8. [4 punti] Scrivere cosa stampa il seguente programma.

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

2
-4 2
-8
1 3
0
4 2

9. [4 punti] Cerchiare le affermazioni vere dato *int a[5]= {129, INT_MIN, INT_MIN | INT_MAX, 262142, 262168};* short int *p = (short*) a; char *q= (char*) a; *q[1]= 1;* sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e $262144 = 2^{18}$ (valori rappresentati in *little endian* e complemento a due). Scrivere la mappa di memoria e giustificare le affermazioni (vere o false). A. $^*((short^*)(q+13)) == ^*((short^*)(q+17))$ B. $(^*(p+5)-p[6])\,\%\,1$ C. $a[2] + ^*(q+1)$ D. $(((int)(p+10) - (int)(a+1)) + q[16] - q[18])\,\%\,6$ E. $\&a[5] - (a+2) - q[1] - 2$ F. $^*(p+2)\,?\,p[7] : !(p[4] + p[5] + p[7])$ **TUTTI FALSI**

ESERCIZIO 3

A: ++(a++) : a++ è un rvalue, sul quale non si può applicare un operatore ++, se a vale 7 equivarrebbe a fare ++7

B: a+ = 2, a++; contiene due effetti collaterali, entrambi su a, prima lo incrementa di 2 e poi di 1

C: se prima applico l'operazione indirizzo (&) e poi quello di dereferenziazione (*), ritorno ad a, che è un lvalue (è uan variabile)

D: l'espressione equivale a *p, che è sinonimo della variabile a, che è un lvalue

E: non genera warning perché && separa i due effetti collaterali su a: a && corrisponde un sequence point

F: * vuole come operando un puntatore, e non una variabile comune come a

G: p a tipo "puntatore a int"

F: *p == 2 e negato fa 0 che è diverso da -2

H: l'indirizzo di a è contenuto in p, quindi sono identici

ESERCIZIO 4

A: 1.2 ha tipo double, che viene convertito a float (il tipo di a)

B: p è un puntatore a costante, quindi non si può modificare il valore della variabile puntata

C: q è un puntatore costante, quindi posso modificare il valore della variabile puntata, ma non il valore di q

D: q è un puntatore costante, quindi non ne posso modificare il valore

E: la precisione di un double è più alta rispetto a quella di un float

F: ++a equivale a 2.2