

Prova scritta Programmazione I, FILA B - 3 Febbraio.

Nome e Cognome: _____

Matricola: _____

1. **3 punti** Elencare tutte le conversioni di tipo; dare il valore finale di *a* e *d* (USHRT_MAX = 65535 e UINT_MAX = 4294967295).

```
1 int fun(int a) {
2     return (a*3.0);
3 }
4
5 int main(void) {
6     unsigned int a = -2U;
7     float b = fun(2.0);
8     long int c = 3 + 'a';
9     unsigned short d = -3;
10 }
```

2.0 DA DOUBLE A INT (PARAMETRO FUN)
 a DA INT A DOUBLE
 a*3.0 DA DOUBLE A INT
 FUN(2.0) DA INT A FLOAT
 1a da long A INT (integer promotion)
 a = 4294967294
 d = 65533
 (Umax+1)+n

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

```
1 int a = 07;
2 while ((a-=1)? a++:a--) {
3     a-=2;
4     printf("OK\n");
5 }
6 int b = 0xfa;
7 b+= a && a++;
8 printf("%d %d\n", a, b);
```

linea 8: 3 + 'a' da int a long int
 linea 9: -3 da int a unsigned short

OK
 OK
 OK
 0 251

3. **4 punti** Scrivere cosa stampa il seguente programma.

```
1 int i, s, r = 5, k = 0, c = 0, cl = 0;
2 for(i=1; i<=r; ++i) {
3     for(s=1; s <= r-i; ++s)
4         { printf("--"); ++c;}
5
6     while(k != 2*i-1) {
7         if (c <= r-1)
8             {printf("%d ", i+k); ++c;}
9         else
10            {++cl; printf("%d ", (i+k-2*cl)
11              + i);}
12         ++k;
13     }
14     cl = c = k = 0;
15     printf("\n");
16 }
```

-----1
 -----2 3 4
 ---3 4 5 7 6
 --4 5 6 7 10 9 8
 5 6 7 8 9 13 12 11 10

4. **3 punti** Cosa stampa il programma? Gli operatori unari hanno precedenza massima, *a* si trova all'indirizzo di memoria 0x7fff54824fff, uno *short* occupa 2 byte ed un *int* 4 byte.

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

2 1
 0x7fff54824fff 0x7fff54825001 2

5. **4 punti** Scrivere cosa stampa il seguente programma.

```

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

```

```

2
3 4
8
4 8 3
1 3 2
4 5

```

6. **3 punti** Scrivere una espressione con 3 effetti collaterali sulla stessa variabile *a*, che compilata NON produca un warning: *unsequenced modification and access to 'a'*. Scrivere un'espressione con 3 effetti collaterali che invece produca tale warning. **(1)** $a+=3, a++, a--$ **(2)** $a+=(a++)+(a++)$
7. **3 punti** Su foglio protocollo, descrivere in modo esauriente i vari tipi di linkage (con esempi).
8. **4 punti** Su foglio protocollo, scrivere una funzione di nome *min* che prende due parametri: un *int** array ed un *int n*. Essi rappresentano un array e la sua lunghezza (numero elementi di tipo *int* contenuti nell'array); la funzione ritorna un *int*. La funzione calcola e ritorna il minore degli elementi dell'array (supporre che gli elementi possano essere positivi e negativi).
9. **4 punti** Cerchiare le affermazioni vere dato $\text{int } a[3] = \{65539, 258, 255\}$; $\text{char } *p = (\text{char}*) a$; $\text{short int } *q = (\text{short}*) a$; sapendo che i tre tipi usati occupano 4, 1, e 2 byte, e $65536 = 2^{16}$ (valori rappresentati in little endian). **(A)** $((\text{int})(q+5) - (\text{int})(a+5)\%2) \neq 0$; **(B)** $(\text{int})\&a[2] > (\text{int})(q+3)$; **(C)** $\&a[2] - a > 1$; **(D)** $*(q+4) - p[4] - p[5] == 252$. **E.** $q[0] - q[2] > 8$. **F.** $*(p+8) + *(p+5) == -1$;

```

11000000
00000000
10000000
00000000
01000000
10000000
00000000
00000000
11111111
00000000
00000000
00000000

```

65536

```

INT MIN (INT* ARRAY, INT N)
{
    IF (N > 0) {
        INT MIN = ARRAY[0];
        FOR (INT i = 1; i < N; i++)
            IF (ARRAY[i] < MIN)
                MIN = ARRAY[i];
        RETURN MIN;
    }
    ELSE {
        PRINTF("Errore");
        RETURN 0;
    }
}

```