

Prova scritta Programmazione I - 9 Luglio 2018.

Nome e Cognome: _____

Matricola: _____

1. **3 punti** Riportare tutte le conversioni di tipo implicite ed esplicite (operazione di *cast*), indicando per ciascuna *imp* o *esp*. Scrivere il valore finale di *a* e *x*.

```
1 int x = 0, i = -1;
2 char a = (char) 64, b = (char) 64,
   c = (char) 1;
3 a = (a+b) - c;
4 unsigned int limit = 10;
5 long n = 20;
6 if ( i < limit )
7   x = limit * n;
```

DAVE NON ESP LE ALTE SONO IMPLICITE

LINIA 2: 64 DA INT A CHAR (ESP) SIA PER A CHE PER B
 1 DA INT A CHAR (ESP)
 LINIA 3: a E b DA CHAR A INT (INTEGER PROMOTION)
 RISULTATO DA INT A CHAR
 LINIA 4: 10 DA INT A UNSIGNED
 LINIA 5: 20 DA INT A LONG
 LINIA 6: -1 DA INT A UNSIGNED INT
 LINIA 7: LIMIT DA UNSIGNED A UNSIGNED LONG (SUDDESU)
 A = 128 X = 0

2. **3 punti** Scrive cosa stampa la seguente porzione di codice

```
1 int i, q, v = 0x5, star = 0;
2 for (i = 1; i < v; i++) {
3   for (q = 1; q <= v-i; q++) {
4     printf("-");
5     while (star != (2*i - 1)) {
6       if (star == 0 || star == 2*i - 2)
7         printf("*");
8       else
9         printf("%d", q+1);
10      star++;
11    }
12    printf("\n");
13    for (i = 0; i < 2*v-1; i++) {
14      printf("*");
15    }
```

linea 7: anche n convertito a unsigned long.
 Il risultato convertito da UL a int

```

- - - - *
- - - * 5 *
- - * 4 4 4 *
- * 3 3 3 3 *
* * 2 2 * * 2 2 *

```

3. **3 punti** Scrive una funzione *main()* che disegni su terminale la seguente composizione di numeri. Il numero di righe è preso come input da tastiera.

```

0
0 1
0 2 4
0 3 6 9
0 4 8 12 16

```

<http://www.techcrashcourse.com/2016/02/c-program-to-print-multiplication-table-pattern.html>

4. **4 punti** Su foglio protocollo, descrivere cosa sono i lvalue e i rvalue, compresi gli lvalue non modificabili. Fornire esempi di espressioni che li contengono, sottolineando lvalue ed rvalue. Scrivere una espressione con esattamente 4 effetti collaterali sulla stessa variabile *a* e due su una variabile *b*, che compilata NON produca un *warning*: *unsequenced modification and access to 'a'* (stesso risultato per *b*). Scrivere un espressione con lo stesso numero di effetti collaterali su *a* e *b* che invece produca tale *warning* (sia su *a* che su *b*).
5. **4 punti** Su foglio protocollo, scrivere la definizione di una funzione *matrix_sum_diagonals* che somma gli elementi sulle due diagonali di una matrice *n x n* (matrice di *int*, ricevuta come parametro dalla funzione) e ritorna un valore intero 1 se la somma è uguale alla somma di tutti gli elementi sulla prima e sull'ultima ultima riga della matrice, o 0 altrimenti.

VEDERE COMMENTI ESERCIZIO 4
 23 FEBBRAIO 2018

Exercise 4

$a = b++$, $a++$, $a++$, $a++$, $b++$; NO WARNING

$a = a++$, $b = b++$, $a++$, $a++$; WARNING PLAS
ACUSU B

6. 4 punti Scrivere cosa stampa il seguente programma.

```

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

```

4	24	5
5	1	6
120	2	7

7. 3 punti Scrivere una funzione ricorsiva *GCD* che calcola il massimo comune divisore tra due valori di tipo *int*. Qual è il numero massimo di *frame* sullo stack se la funzione viene chiamata tra 18 e 3?

8. 4 punti Cerchiare le affermazioni vere dato $\text{long long } a[3] = \{1537, -10, \text{LLONG_MIN} + 511\};$
 $\text{int } *p = (\text{int} *) a; \text{char } *q = (\text{char} *) a; p[1] = \text{INT_MAX}, p[4] += 4096, *(q+19) = 24, q[4] = -2;$
 sapendo che i tre tipi usati occupano 8, 4, e 1 byte, e $4096 = 2^{12}$ (valori rappresentati in *little endian* e complemento a due). Scrivere la mappa di memoria e giustificare le affermazioni (vere o false). Gli operatori $|$ e $\&$ ritornano rispettivamente l'*or* e l'*and* bit-a-bit dei due operandi, mentre \gg rappresenta l'operatore di *shift* di n posizioni a destra. Supporre che i tipi *char* siano tipi con segno: il bit più significativo uguale a 1 rappresenta un valore negativo.

A. $(q[17] | q[20]) + q[2] == 0$ B. $(*(p+3) - p[2]) \% 3$ C. $((\text{int})(p+12) - (\text{int})(a+4)) - q[21]) \% 1$
 D. $*((\text{int} *) (\&q[9])) - p[3] > 0$ E. $*((\text{long long} *) (\&p[1])) > *((\text{long long} *) (\&p[2]))$ F. $((q[19] < < 1) \& q[19]) + q[17]$

9. 2 punti Sottolineare le espressioni per ciascuno dei seguenti punti. Se qualcuno di essi contiene qualche errore sintattico, riportarlo su foglio protocollo.

A. if (a > b); B. do {a++; b++;} while (a > 0); C. for (int i = 0; i > 0; i++); D. a + b + c
 E. if (;); F. void fun (int a, int b, int c);

↑
 errore
 dentro if deve
 esserci espressione, non comando (;)

Exercício 8

100000000

011000000

000000000

000000000

$q[2]$

$+(lowc - lowc^*)(dp[1])$

011111111

111111111

111111111

111111110

011011111

111111111

111111111

111111111

111111111

111111111

100010000

000000000

000000000

000110000

$p[2]$

$dp[9]$

$q[12]$

$+(p+3)$

000000000

000000000

000000000

000000001

$q[10]$

(A) $q[17] \mid q[10] + q[1] == 17$
 $10001000 \mid 00000000 + 0$ **FALSE**

(B) $+(p+3) - p[2] \times 3 == 0$
 $-1 - -10$ **FALSE**

(C) $(low +)(p+12) - (low)(a+1) == 16$
 Exercício $p == 1000$

$p+12 = 1012$ e $a+1 = 1032$

$q[21] == 0 \mid 16 \times 1 = 0$

FALSE

(D) $+(low +)(dp[9]) - p[5] == 0$
 $-1 - -1$ **FALSE**

(E) $+(lowc - lowc^*)(dp[1]) \mid +(lowc - lowc^*)(dp[2])$
 $==$

FALSE

011111111

111111111

111111111

111111111

011011111

111111111

111111111

111111111

>

011011111

111111111

111111111

111111111

111111111

111111111

111111111

111111111

(F) $q[13]$
 $00011000 \leq 1$ e $q[14] == 00011000$
 00110000 00011000