

Prova scritta Programmazione I - 17 Settembre 2018.

Nome e Cognome: _____

Matricola: _____

1. **3 punti** Elencare le conversioni di tipo, specificando se implicite o esplicite. Dati $UINT_MAX = 4294967295$ e $USHRT_MAX = 65535$, scrivere i valori di a e d prima dell'uscita da *main*.

```
1 double pow3(int a) {
2     return (a * a * a); }
3
4 int main(void) {
5     unsigned a = -2;
6     float b= pow3(a);
7     double *p= malloc(sizeof(double));
8     unsigned short d = -3;
9 }
```

linea 5: -2 da int a unsigned int (i) (i)= implicita
 linea 1: a da unsigned a int (i)
 linea 2: a * a * a da int a double (i)
 linea 6: risultato pow da double a float (i)
 linea 7: risultato malloc da void* a double* (i)
 linea 8: -3 da int a unsigned short (i)

a: 4294967294
 d: 65533

2. **3 punti** Scrivere cosa stampa il seguente programma. Il valore intero in tabella ASCII del carattere '*' è 42

```
1 int i,j, r= 5;
2 for(i=1; i<=r; i++){
3     for(j=1; j <= r; j++){
4         if(j < i){
5             printf("-");
6         } else {
7             printf("%d", r-i > 3? j-i : '
8             *');
9         } printf("%d\n", i-r);}
10 }
```

01234-4
 -42424242-3
 --424242-2
 ---4242-1
 ----420

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

```
1 int a= 0xc;
2 while(a > 15 ? a++: (a--, --a)) {
3     printf("OK\n");
4     if (a + 2 >= 0x10) {
5         break;
6         printf("NO MORE OK\n");
7     }
8     a+=3; }
9 !a || a%2 || a++;
10 printf("a: %d\n", a);
```

OK
 OK
 OK
 OK
 OK
 OK
 a: 17

4. **3 punti** Dato il seguente *main*, definire la funzione *matrix_avg_row* che calcola in *avg_row* la media dei valori della matrice *per riga*.

```
1 int main() {
2     int m[3][2]= {1,4,5,2,1,1};
3     double avg_row[3];
4     matrix_avg_row(2, 3, m, avg_row);
5     printf("%f\n", avg_row[0]);
6     printf("%f\n", avg_row[1]);
7     printf("%f\n", avg_row[2]);
8 }
9
```

Vedere esercizio 4 del 18 Gennaio 2018 (calcolare la media sulle righe invece che sulle colonne)

```
for (int i=0; i < row; i++) {
    for (int j=0; j < column; j++)
        avg_row[i] += m[i][j];
    avg_row[i] /= 3;
}
```

5. **3 punti** Scrivere una singola espressione con 3 effetti collaterali su una variabile a e 2 effetti collaterali su b , che generi un warning *multiple unsequenced modifications* a causa di a ma non di b , evidenziando i *sequence points* con una freccia. Scrivere una seconda espressione con 2 effetti collaterali su a che invece non generi tale warning; questa espressione deve contenere a due volte come *lvalue* e due volte come *rvalue*.

1) $b = a++$, $a = a++$, $b++$
le frecce indicano le virgole

2) $a = a++$, $a+3$, a

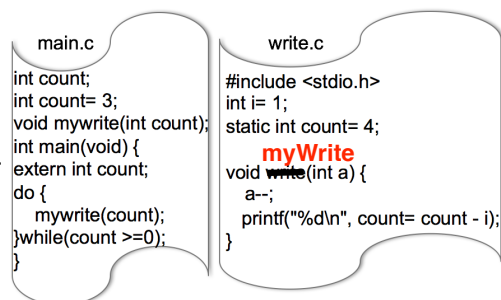
6. **3 punti** Su foglio protocollo, scrivere una funzione ricorsiva di nome *fib* che ha come parametro un intero n e restituisce il valore $fib(n) + 1$: invece che 0, 1, 1, 2, 3, 5, 8, 13, 21 (etc), 1, 2, 2, 3, 4, 6, 9, 14, 22 (etc). Qual è il numero massimo di copie (*frame*) della funzione aperte nello stack con $n == 3$?
7. **4 punti** Scrivere cosa stampa il seguente programma.

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

4 3 4 10
5 4 3 11
6 5 2 12
7 6 1 13
5
5 88 3

count ha linkage esterno in main.c e interno in write.c
i ha linkage esterno in write.c
myWrite ha linkage esterno in write.c e main.c

8. **4 punti** Dire quali compilazioni provocano errore a causa del linker (e perché): 1) `gcc -c write.c write.c`, 2) `gcc -c main main.c`, 3) `gcc -o main.c`, 4) `gcc write.c main.c -o main`. 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*? Cosa stampa il programma? Elencare tutte le definizioni e dichiarazioni in ogni file.



2) Errore, non esiste file main da compilare insieme a file main.c
3) Dopo -o deve essere specificato il file eseguibile da creare: in questo caso viene considerato main.c, ma conseguentemente mancano file di input per gcc

1) e 4) non ritornano nessun errore

In main, count viene prima dichiarato (tentativo definizione) e sotto definito, myWrite viene dichiarata, main viene definita, count viene dichiarato dentro main

In write.c i viene definita, count viene definita, myWrite viene definita

Viene stampato
3
2
1
0
-1
-2 e così via

9. **4 punti** Cerchiare le affermazioni vere dato $int\ a[5] = \{25, 81, [2] = INT_MAX, 131046, 131328\}$; $short\ int\ *p = (short*)\ a$; $char\ *q = (char*)\ a$; $q[16] = -1$; sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e $131072 = 2^{17}$ (valori rappresentati in *little endian*). A. $(*p - q[17] - *(q + 18)) - 0x16$ B. $*(p + 4) + *(q + 14) + q[11] == 128$ C. $((\&a[5] - a) + q[5]) \% 2$ D. $((int)(a + 5) - (int)(p + 5) + q[5]) \% 2$ E. $((q[17] | q[0]) \& q[4]) \% 5$ F. $(p[4] | *(p + 5)) \& p[8] \% 256$

```
#include<stdio.h>

unsigned int fib_rec();

int count_frame= 0;

unsigned int first_n= 0;

int main() {
    printf("Give me n\n");
    scanf("%u", &first_n);
    printf("Fibonacci of %u= %u\n", first_n,  fib_rec(first_n));
    printf("Opened frames= %d\n", count_frame);
}

unsigned int fib_rec(unsigned int n) {
    count_frame++;
    if (n == 0) {
        return 1;
    }
    if (n == 1) {
        return 2;
    }
    return fib_rec(n - 1) + fib_rec(n - 2) - 1;
}

//I frame aperti sono 5
```

```

#include<stdio.h>
#include<limits.h>

int main() {
    int a[5]= {25, 81, [2]= INT_MAX, 131046,131328};
    short int *p = (short*) a;
    char *q= (char*) a;
    q[16]= -1;

    printf("((*p - q[17] - *(q+18)) - 0x16 %d\n", (*p - q[17] - *(q+18)) -
        0x16);
    printf("(*p+4) + *(q+14) + q[10]) == 128 %d \n", (*p+4) + *(q+14) +
        q[11]) == 128);
    printf("((&a[5] - a) + q[5]) mod 2 %ld\n", ((&a[5] - a) + q[5]) % 2);
    printf("((int) (a + 5) - (int) (p+5) + q[5]) mod 2 %d \n", ((int) (a +
        5) - (int) (p+5) + q[5]) % 2);
    printf("(((q[17] | q[0]) & q[4]) % 5) mod 2 %d\n", (((q[17] | q[0]) &
        q[4]) % 5) % 2);
    printf("(p[4] | *(p+5)) & p[8] mod 256 %d\n", (p[4] | *(p+5)) & p[8] %
        256);

}

```

/* Mappa di memoria

```

10011000
00000000
00000000
00000000

```

```

10001010
00000000
00000000
00000000

```

```

11111111
11111111
11111111
11111110

```

```

01100111
11111111
10000000
00000000

```

```

11111111
10000000
01000000
00000000

```

*/